

**System oceny formalno-merytorycznej wniosków aplikacyjnych (SOFM) składanych w ramach Regionalnego Programu Operacyjnego Województwa Podlaskiego na lata 2014-2020**

# **Instrukcja administratora aplikacji**

Aplikacja współfinansowana przez Unię Europejską w ramach Regionalnego Programu Operacyjnego Województwa Podlaskiego na lata 2014-2020

Białystok, luty 2017

# Spis treści

## Spis treści

1.	Informacje ogólne .....	3
2.	Instalacja SOFM .....	3
2.1	Instalacja bazy danych .....	4
2.2	Instalacja aplikacji .....	5
2.3	Konfiguracja serwera HTTP Apache .....	13
3.	Konfiguracja aplikacji SOFM.....	17
3.1	Konfiguracja adresu e-mail.....	19
3.2	Konfiguracja czasu trwania sesji .....	20
4.	Szablony wiadomości e-mail .....	20
5.	Logi aplikacji SOFM .....	26
6.	Udostępniane Instrukcje .....	27
7.	Polityka Bezpieczeństwa i licencja.....	28
8.	Grafiki UE wykorzystywane na wydrukach.....	28
9.	Kopie bezpieczeństwa .....	29
9.1	Kopia bezpieczeństwa plików aplikacji.....	30
9.2	Kopia bezpieczeństwa bazy danych .....	30
9.3	Przywrócenie systemu z kopii .....	30

## 1. Informacje ogólne

Aplikacja SOFM przeznaczona jest do uruchomienia na serwerach z systemem operacyjnym Linux i jest złożona z 3 podstawowych komponentów:

- bazy danych (MySQL)
- aplikacji internetowej w technologii (Python+Django)
- serwera HTTP (Apache)

Wszystkie te komponenty współpracując ze sobą w systemie operacyjnym Linux udostępniają funkcjonalność aplikacji SOFM.

W skład systemu SOFM wchodzi dodatkowo szereg dodatkowych narzędzi i bibliotek, wymaganych do poprawnej pracy lub dodatkowych, ułatwiających administrację oraz przetwarzanie danych przez aplikację. Wśród nich są między innymi:

### 1. Baza danych

- MySQL-python – konektor do bazy danych, niezbędny aby aplikacja w technologii Python mogła się komunikować z bazą danych MySQL
- phpMyAdmin – opcjonalne narzędzie służące do administrowania bazą danych MySQL z poziomu przeglądarki internetowej (instalacja narzędzia jest opcjonalna)

### 2. Aplikacja SOFM (Python + Django)

- Django MPTT – biblioteka ułatwiająca przetwarzanie oraz zarządzanie drzewami dokumentów w Django
- django-guardian – biblioteka wspomagająca zarządzanie uprawnieniami użytkowników do dokumentów
- wkhtmltopdf + django-wkhtmltopdf – aplikacja/biblioteka wykorzystywana do generowania plików \*.pdf (wydruk kart oceny i raportów)
- XlsxWriter – biblioteka wykorzystywana do generowania plików \*.xlsx (mechanizm eksportu do excel)
- StringGenerator – biblioteka wykorzystywana do generowania losowych napisów
- Libre Office – oprogramowanie wykorzystywane do konwersji plików \*.docx, \*.doc i \*.odt na pliki \*.pdf
- ImageMagic – biblioteka wykorzystywana do konwersji plików \*.jpg i \*.png na pliki \*.pdf

### 3. Serwer http (domyślnie Apache)

- moduł mod-wsgi
- moduł ssl

## 2. Instalacja SOFM

W celu skutecznego zainstalowania i wstępnego skonfigurowania systemu SOFM na serwerze z zainstalowanym systemem operacyjnym Linux Ubuntu 14.04 Server należy wykonać czynności opisane poniżej.

## 2.1 Instalacja bazy danych

Zalecaną bazą danych jest MySQL. Istnieje również możliwość instalacji SOFM z bazą PostgreSQL. Aby zainstalować bazę MySQL najlepiej w linii komend systemu Ubuntu wykonać polecenie:

```
sudo apt-get install mysql-server
```

Na zainstalowanym serwerze MySQL należy utworzyć bazę danych np. o nazwie *sofm* oraz użytkownika, który będzie zapewniał dostęp do bazy danych z poziomu aplikacji.

**Bardzo istotna jest właściwa konfiguracja kodowania w bazie danych. Musi to być kodowanie oparte na UTF8. Takie kodowanie zapewni prawidłowe wyświetlanie polskich znaków.**

Najlepiej wykonać tę czynności korzystając z graficznego oprogramowania do zarządzania bazą danych MySQL. Można do tego celu wykorzystać oprogramowanie *phpMyAdmin*, które należy osobno zainstalować i skonfigurować na serwerze, lub inne oprogramowanie przeznaczone do uruchomienia na komputerze lokalnym takie jak *MySQL Workbench*, które pozwala na połączenie się ze zdalną bazą danych *MySQL* i zarządzanie ją.

Inną możliwością jest utworzenie bazy oraz użytkownika z poziomu linii komend systemu operacyjnego. Można to zrobić następująco:

Za pomocą polecenia:

```
sudo mysql -p
```

należy przejść do linii komend serwera bazy danych *MySQL*. Następnie należy wykonać polecenie tworzące nową bazę danych i ustawiające kodowanie utf8:

```
CREATE DATABASE `sofm` CHARACTER SET utf8 COLLATE utf8_general_ci;
```

W kolejnym kroku należy utworzyć użytkownika. Można to zrobić wykonując polecenie:

```
CREATE USER 'sofmuser'@'localhost' IDENTIFIED BY 'password';
```

Utworzony użytkownik bazy danych powinien dostać uprawnienia do utworzonej wcześniej bazy *sofm*. Użytkownik będzie wykorzystywany do połączeń aplikacji z bazą danych, dlatego musi mieć uprawnienia pozwalające mu na wszelkie operacje na tabelach w ramach danej bazy. Aby nadać mu takie uprawnienia należy wykonać polecenie:

```
GRANT CREATE, DELETE, DROP, EXECUTE, INSERT, SELECT, ALTER, UPDATE, INDEX,  
CREATE TEMPORARY TABLES, CREATE VIEW, TRIGGER, REFERENCES, EVENT, SHOW VIEW ON  
sofm . * TO 'sofmuser'@'localhost';
```

## Komunikacja z bazą od strony środowiska Python

Aby zapewnić komunikację z bazą danych MySQL z poziomu aplikacji SOFM należy doinstalować w systemie dwa dodatkowe pakiety niezbędne do późniejszej instalacji konektora do bazy danych. Aby to zrobić należy wykonać następujące komendy:

```
sudo apt-get install libmysqlclient-dev  
sudo apt-get install python-dev
```

## 2.2 Instalacja aplikacji

### Python

W celu poprawnego zainstalowania aplikacji SOFM należy w pierwszej kolejności zainstalować kilka komponentów, z których aplikacja korzysta dostarczając swych funkcjonalności. Kod aplikacji jest napisany w języku Python, dlatego pierwszą i podstawową czynnością będzie zainstalowanie bibliotek Python na serwerze. Aplikacja jest wytworzona i przystosowana do wersji Python 2.7.x, a tworzona była dokładnie na wersji Python 2.7.6. Przed instalacją warto się upewnić czy Python nie jest już zainstalowany w systemie serwera. Można to zrobić wpisując w linii komend polecenie:

```
python
```

jeśli środowisko Python jest zainstalowane, wynik polecenia powinien wyglądać następująco:

```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)  
[GCC 4.8.2] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
>>>
```

W Ubuntu 14.04 Server domyślnie zainstalowany jest Python 2.7.6, czyli wersja zgodna z wymaganiami aplikacji sofm.

Jeśli Python nie jest zainstalowany, należy go doinstalować np. korzystając w Ubuntu z polecenia:

```
sudo apt-get install python2.7
```

### Django

Kiedy Python jest już zainstalowany, można przejść do instalacji frameworka Django, na którym bazuje aplikacja SOFM. Aby poprawnie i szybko zainstalować Django, należy przed

tą czynnością zainstalować oprogramowanie do instalacji modułów Python o nazwie **Pip**. W tym celu należy zastosować komendę:

```
sudo apt-get install python-pip
```

Aby maksymalnie uniezależnić instancję aplikacji od platformy systemowej zaleca się instalację Django i aplikacji w środowisku wirtualnym z wykorzystaniem narzędzia **virtualenv**. Narzędzie to pozwala na tworzenie niezależnych środowisk bazujących na technologii Python, dzięki czemu istnieje możliwość uruchomienia kilku niezależnych aplikacji Python na jednym systemie operacyjnym. Dodatkowo instalowanie kolejnych modułów nadpisuje jedynie lokalne środowisko wirtualne Python, dzięki czemu łatwiej jest zarządzać zainstalowanymi bibliotekami i modułami. Aby zainstalować narzędzie virtualenv należy skorzystać z komendy:

```
sudo pip install virtualenv
```

**Przed instalacją aplikacji na serwerze należy zainstalować w systemie serwer HTTP.** Zalecany serwerem jest **Apache HTTP Server**, natomiast możliwa jest instalacja i konfiguracja innego serwera HTTP, wspierającego technologię **WSGI**. Aby zainstalować w systemie serwer HTTP Apache należy wykonać następujące polecenie:

```
sudo apt-get install apache2
```

**konfiguracja serwera HTTP jest opisana w dalszej części dokumentu.**

Po instalacji serwera w strukturze katalogów systemu, w katalogu */var* powinien zostać wytworzony standardowy katalog serwera Apache o nazwie *www*. Aby można było bez problemów pracować na wskazanej lokalizacji, najlepiej zmodyfikować domyślnie nadane przez serwer uprawnienia. Apache domyślnie, w dostępie do plików, korzysta z użytkownika o nazwie **www-data** należącego do grupy użytkowników o tej samej nazwie - *www-data*. Należy zmienić właściciela oraz grupę właściciela katalogu */var/www* na **www-data**. Można to zrobić poleceniem:

```
sudo chown -R www-data:www-data /var/www/
```

następnie należy nadać uprawnienia zapisu do katalogu */var/www*, grupie właściciela katalogu czyli grupie *www-data*. Można to zrobić wykonując polecenie:

```
sudo chmod 775 -R /var/www/
```

Po wykonaniu tego polecenia można dodać własnego użytkownika do grupy `www-data`, co zapewni możliwość tworzenia katalogów i plików w katalogu `/var/www/`. Można to zrobić korzystając z polecenia:

```
sudo usermod -a -G www-data sofuser
```

gdzie **sofuser** to nazwa użytkownika, którym logujemy się na serwerze i za pomocą którego będą na serwerze umieszczane pliki.

Po poprawnej instalacji serwera HTTP i określeniu niezbędnych uprawnień, można przejść do instalacji Django w środowisku wirtualnym **virtualenv**.

Aby to zrobić, należy w pierwszej kolejności stworzyć środowisko wirtualne Python, z którego będzie korzystała aplikacja i na którym zostanie zainstalowane Django i pozostałe wymagane moduły.

Należy przejść do lokalizacji `/var/www/` i utworzyć w niej katalog np. o nazwie `sofm`. Wewnątrz tego katalogu należy utworzyć podkatalog na pliki środowiska wirtualnego np. o nazwie `env`. Po utworzeniu katalogu można przejść do wytworzenia nowego środowiska wirtualnego za pomocą komendy, podając lokalizację wytworzonego wcześniej katalogu `env`:

```
virtualenv --no-site-packages env
```

Po wytworzeniu środowiska wirtualnego można przejść do instalacji Django oraz aplikacji SOFM. Django należy zainstalować w nowo utworzonym środowisku wirtualnym. Aby to zrobić należy znajdując się w katalogu `/var/www/sofm/` wykonać polecenie:

```
env/bin/pip install django==1.8.9
```

wskazana wersja Django powinna zostać pobrana i zainstalowana w środowisku wirtualnym. Aby potwierdzić powodzenie instalacji oraz sprawdzić wersję Django należy wykonać polecenie:

```
env/bin/django-admin.py version
```

Aplikacja została napisana z wykorzystaniem Django w wersji 1.8.6. Powinna poprawnie funkcjonować na wersjach 1.8. Obecnie najnowszą wersją 1.8.x jest wersja 1.8.9. Wersja 1.8 jest oznaczona przez producenta Django jako LTS, dlatego obecnie można traktować ją jako najbardziej stabilną.

**Po instalacji Django, należy doinstalować w wytworzonym środowisku wirtualnym dodatkowe biblioteki oraz aplikacje, niezbędne do prawidłowego funkcjonowania aplikacji SOFM.**

## Django MPTT

Jest to biblioteka służąca do budowania i zarządzania hierarchicznymi strukturami danych. W aplikacji SOFM biblioteka wspomaga tworzenie i zarządzanie strukturą katalogów i dokumentów. Aby ją zainstalować należy znajdując się w katalogu `/var/www/sofm/` wywołać następujące polecenie:

```
env/bin/pip install django-mptt
```

## django-guardian

Jest to biblioteka wspomagająca zarządzanie uprawnieniami użytkowników do obiektów. Aby ją zainstalować należy znajdując się w katalogu `/var/www/sofm/` wywołać następujące polecenie:

```
env/bin/pip install django-guardian
```

## XlsxWriter

Jest to biblioteka wspomagająca eksport danych do plików typu Excel

```
env/bin/pip install XlsxWriter
```

## wkhtmltopdf

Jest to aplikacja służąca do generowania plików \*.pdf. Aplikacja nie jest częścią projektu aplikacji SOFM, ale niezależnym narzędziem w systemie operacyjnym. Aplikacja SOFM jedynie wykorzystuje jej możliwości do generowania wydruków kart oceny oraz raportów w formie dokumentów \*.pdf. Aby zainstalować w systemie wkhtmltopdf należy wykonać następujący zestaw komend:

```
sudo apt-get install libgtk2.0-0 libgdk-pixbuf2.0-0 libfontconfig1 libxrender1  
libx11-6 libglib2.0-0 libxft2 libfreetype6 libc6 zlib1g libpng12-0  
libstdc++6-4.8-dbg-arm64-cross libgcc1
```

```
sudo sudo wget http://download.gna.org/wkhtmltopdf/obsolete/linux/wkhtmltopdf-0.10.0\_rc2-static-amd64.tar.bz2
```

```
sudo tar xvjf wkhtmltopdf-0.10.0_rc2-static-amd64.tar.bz2
```

```
sudo mv wkhtmltopdf-amd64 /usr/local/bin/wkhtmltopdf
```

```
sudo chmod +x /usr/local/bin/wkhtmltopdf
```

Aplikacja domyślnie wykorzystuje wkhtmltopdf w wersji 10.0\_rc2



## **django-wkhtmltopdf**

Jest to biblioteka usprawniająca wykorzystanie aplikacji wkhtmltopdf w aplikacjach napisanych w oparciu o Python i Django

```
env/bin/pip install django-wkhtmltopdf
```

## **StringGenerator**

Jest to biblioteka wspomagająca generowanie losowej treści napisów.

```
env/bin/pip install StringGenerator
```

## **MySQL-python**

Jest to konektor do bazy danych, zapewniający możliwość komunikacji aplikacji Django z bazą danych MySQL. Należy go zainstalować w wytworzonym wcześniej środowisku wirtualnym. Aby to zrobić należy znajdując się w katalogu `/var/www/sofm/` wywołać następujące polecenie:

```
env/bin/pip install MySQL-python
```

## **Libre Office**

Aplikacja SOFM konwertuje importowane do systemu dokumenty edytowalne typu doc, docx, odt na pliki \*.pdf w celu ich prezentacji w przeglądarce. Wykorzystuje do tego pakiet Libre Office. Aby aplikacja mogła konwertować importowane pliki, w systemie musi być zainstalowany pakiet Libre Office. Aby go zainstalować można skorzystać z następujących komend:

```
sudo apt-get install default-jre  
sudo apt-get install libreoffice-common  
sudo apt-get install --no-install-recommends libreoffice-writer
```

## **Convert**

Aplikacja, w celu ujednolicenia prezentacji konwertuje na format \*.pdf również zdjęcia umieszczane w systemie. Wykorzystuje do tego narzędzie bazujące na pakiecie ImageMagick. Aby zainstalować niezbędny pakiet należy wykonać następujące polecenie:

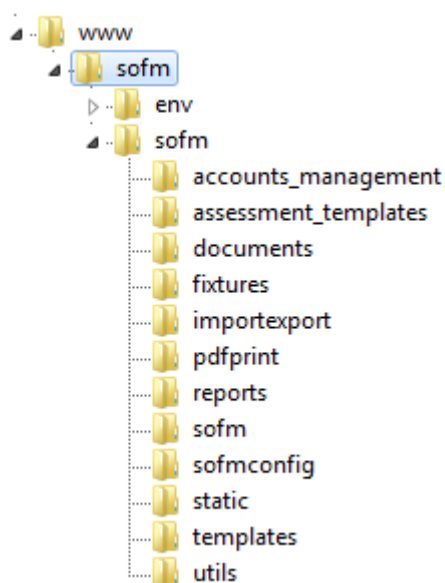
```
sudo apt-get install imagemagick
```

## Aplikacja SOFM

Aby zainstalować aplikację należy utworzyć w lokalizacji `/var/www/sofm/` katalog o nazwie np. `sofm`, najlepiej obok utworzonego wcześniej katalogu `/var/www/sofm/env` przechowującego pliki środowiska wirtualnego Python. **Pakiet instalacyjny aplikacji SOFM zawiera spakowany kod aplikacji SOFM. Należy umieścić zawartość pakietu w nowo utworzonym katalogu.**

## Struktura katalogów aplikacji

Struktura katalogów zainstalowanej aplikacji w katalogu `/var/www/` powinna wyglądać następująco:



## Konfiguracja połączenia z bazą danych

Pierwszym krokiem w celu uruchomienia aplikacji jest konfiguracja połączenia z utworzoną wcześniej bazą danych. Głównym plikiem konfiguracyjnym aplikacji jest umieszczony w katalogu `sofm` plik `settings.py`. Zawiera on najważniejsze parametry konfiguracyjne aplikacji, w tym konfigurację połączenia z bazą danych. **Opis wybranych parametrów konfiguracyjnych jest umieszczony w dalszej części dokumentu.** Aby skonfigurować połączenie z bazą danych należy uruchomić w edytorze tekstu plik `settings.py` i odnaleźć sekcję **DATABASES**:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'nazwabazydanych',
        'USER': 'uzytkownik',
        'PASSWORD': 'haslo',
    }
}
```

Sekcja ta odpowiada właśnie z konfigurację bazy danych. W parametrze **NAME** należy wprowadzić nazwę utworzonej na potrzeby SOFM bazy danych. W parametrze **USER** należy

wprowadzić nazwę użytkownika, z wykorzystaniem którego aplikacja będzie łączyła się z bazą danych, a w parametrze **PASSWORD** należy wprowadzić hasło wprowadzonego użytkownika. Konfiguracja może wyglądać następująco:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'sofm',
        'USER': 'sofmuser',
        'PASSWORD': 'password321',
    }
}
```

zgodnie z konfiguracją bazy danych, którą wytworzono w punkcie 2.1 tego dokumentu.

### Wytworzenie struktury bazy danych

Struktura bazy danych jest zdefiniowana w aplikacji. Aby można było uruchomić aplikację SOFM należy na samym początku wytworzyć strukturę bazy danych w utworzonej i skonfigurowanej pustej bazie danych. Django posiada mechanizmy, które umożliwiają taką operację. W katalogu aplikacji `/var/www/sofm/sofm/` znajduje się plik **manage.py**. Jest to główny skrypt aplikacji Django umożliwiający dodatkowo sprawne zarządzanie i konfigurację środowiska. Za pomocą tego narzędzia można m.in. wytworzyć strukturę bazy danych aplikacji w skonfigurowanej w pliku **settings.py** bazie danych. Aby to zrobić, znajdując się w katalogu aplikacji, należy wykonać następujące polecenie:

```
../env/bin/python manage.py makemigrations
```

To polecenie wytworzy w tle niezbędne do wykonania migracji skrypty, a wykonanie polecenia:

```
../env/bin/python manage.py migrate
```

spowoduje wytworzenie w skonfigurowanej wcześniej bazie danych, odpowiedniej struktury. Wynik udanej operacji tworzenia struktury bazy danych powinien wyglądać następująco:

```

Operations to perform:
  Apply all migrations: sofmcconfig, assessment_templates, documents, sessions, admin, guardian, auth, con
nts_management
Running migrations:
  Rendering model states... DONE
  Applying accounts_management.0001_initial... OK
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying assessment_templates.0001_initial... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_auto_20160227_2236... OK
  Applying auth.0009_auto_20160227_2320... OK
  Applying documents.0001_initial... OK
  Applying guardian.0001_initial... OK
  Applying sessions.0001_initial... OK
  Applying sofmcconfig.0001_initial... OK
sofmuser@sofm:/var/www/sofm/sofm$

```

## Wypełnienie bazy danych wartościami początkowymi

Po wytworzeniu struktury bazy danych, należy wypełnić ją początkowymi danymi. **Należy zaimportować definicje uprawnień przypisywanych poszczególnym profilom oraz definicje ról użytkowników.** Django udostępnia mechanizm sprawnego importu zdefiniowanych wcześniej danych w formacie JSON. Pliki zawierające definicje danych początkowych aplikacji SOFM znajdują się w katalogu **fixtures**, w katalogu głównym aplikacji. Aby zaimportować do wytworzonej bazy danych definicję uprawnień należy wykonać polecenie:

```
../env/bin/python manage.py loaddata fixtures/permissions.json
```

co spowoduje utworzenie w bazie danych definicji uprawnień. Wykonanie polecenia:

```
../env/bin/python manage.py loaddata fixtures/roles.json
```

spowoduje utworzenie w bazie ról użytkowników.

## Wytworzenie pierwszego użytkownika (SUPERADMIN)

W celu umożliwienia pierwszego logowania do aplikacji oraz dalszej parametryzacji aplikacji należy utworzyć pierwszego użytkownika w bazie danych. Będzie on miał pełen dostęp do aplikacji. Aby utworzyć pierwszego użytkownika aplikacji należy skorzystać z polecenia:

```
../env/bin/python manage.py createsuperuser
```

efekt wykonania polecenia może wyglądać następująco:

```
Username (leave blank to use 'sofmuser'): sofmsadmin
Email address: sofm@sofm.pl
Password:
Password (again):
Superuser created successfully.
sofmuser@sofm:/var/www/sofm/sofm$
```

## Konfiguracja lokalizacji składowania importowanych do systemu plików

Aplikacja SOFM przechowuje importowane do systemu pliki dokumentów w strukturze katalogów systemu operacyjnego serwera. Standardową lokalizacją, w której aplikacja SOFM umieszcza importowane pliki jest katalog `/sofm/document_files`. Domyślnie w systemie nie ma takiego katalogu, dlatego trzeba go utworzyć i skonfigurować uprawnienia, tak aby pliki mogły być w nim zapisywane. Aby to zrobić najlepiej skorzystać z następujących komend:

```
sudo mkdir sofm
cd sofm
sudo mkdir document_files
```

w celu utworzenia katalogów oraz

```
sudo chown -R www-data:www-data /sofm/
```

w celu zmiany właściciela katalogu i tym samym umożliwienie zapisu plików do tej lokalizacji.

W celu przetestowania uruchomienia aplikacji już na tym etapie, należy skorzystać z komendy:

```
../env/bin/python manage.py runserver
```

Jeśli wszystko zostało skonfigurowane prawidłowo, w konsoli powinien zostać wyświetlony następujący ciąg komunikatów:

```
Performing system checks...
System check identified no issues (0 silenced).
March 01, 2016 - 11:57:28
Django version 1.8.4, using settings 'sofm.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

## 2.3 Konfiguracja serwera HTTP Apache

Do poprawnego funkcjonowania, aplikacja potrzebuje jeszcze właściwej konfiguracji serwera HTTP, w tym przypadku Apache. Serwer powinien być już zainstalowany w efekcie wcześniej wykonanych operacji, dlatego wystarczy go skonfigurować do współpracy z Django.

Podstawowym modulem niezbędnym do skonfigurowania udostępniania funkcjonalności aplikacji SOFM za pośrednictwem serwera Apache jest moduł **mod\_wsgi**. Zazwyczaj nie jest domyślnie zainstalowany, dlatego aby go pobrać i zainstalować, należy wykonać komendę:

```
sudo apt-get install libapache2-mod-wsgi
```

po udanej instalacji moduł zostanie automatycznie włączony.

W kolejnym kroku należy włączyć moduł SSL serwera Apache. Można to zrobić za pomocą komendy:

```
sudo a2enmod ssl
```

W celu konfiguracji dostępu do aplikacji SOFM należy otworzyć w edytorze tekstu plik konfiguracyjny */etc/apache2/sites-available/default-ssl.conf*. W pliku należy wprowadzić następującą konfigurację:

```
Alias /static /var/www/sofm/sofm/static
<Directory /var/www/sofm/sofm/static>
    Require all granted
</Directory>

<Directory /var/www/sofm/sofm/sofm>
<Files wsgi.py>
    Require all granted
</Files>
</Directory>

WSGIDaemonProcess sofm python-
path=/var/www/sofm/sofm:/var/www/sofm/env/lib/python2.7/site-packages
WSGIProcessGroup sofm
WSGIScriptAlias / /var/www/sofm/sofm/sofm/wsgi.py
```

Powyższa konfiguracja zawiera wskazanie, nadanie uprawnień do skryptu wsgi aplikacji SOFM oraz ścieżkę środowiska Python, za pomocą którego ma być uruchamiana aplikacja (w tym przypadku utworzone środowisko wirtualne env). Pierwsza część aplikacji definiuje dostęp do lokalizacji przechowującej pliki statyczne systemu SOFM (grafiki, css, js)

Warto też zwrócić uwagę czy we wspomnianym pliku konfiguracyjnym jest włączona obsługa SSL. Prawidłowa konfiguracja odpowiadająca za tą opcję powinna wyglądać następująco:

```
SSLEngine on
```

Aby zmiany konfiguracji zostały zastosowane, należy zresetować serwer np. poleceniem:

```
sudo /etc/init.d/apache2 restart
```

## Certyfikaty

Jeśli przy połączeniu SSL ma być wykorzystywany inny certyfikat niż domyślny należy podać lokalizację niezbędnych plików.

```
SSLCertificateFile /etc/apache2/ssl/www_server_com.crt  
SSLCertificateKeyFile /etc/apache2/ssl/server.key  
SSLCertificateChainFile /etc/apache2/ssl/www_server_com.ca-bundle
```

Aby zmiany konfiguracji zostały zastosowane, należy zresetować serwer np. poleceniem:

```
sudo /etc/init.d/apache2 restart
```

## Uruchomienie konfiguracji

Aby konfiguracja została zaczytana przez serwer, musi znaleźć się w katalogu */etc/apache2/sites-enabled/*. Jeśli nie ma jej we wskazanej lokalizacji, najlepiej jest wytworzyć dowiązanie, tak aby nie trzeba było modyfikować konfiguracji w dwóch miejscach. Aby wytworzyć dowiązanie należy skorzystać z następującego polecenia:

```
sudo ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/default-ssl.conf
```

Aby zmiany konfiguracji zostały zastosowane, należy zresetować serwer np. poleceniem:

```
sudo /etc/init.d/apache2 restart
```

## Przekierowanie HTTP -> HTTPS

Aby zapewnić pełne bezpieczeństwo należy wymusić komunikację z serwerem za pomocą protokołu HTTPS, z wykorzystaniem skonfigurowanego przed chwilą modułu SSL, można zastosować automatyczne przekierowania. Po ich zastosowaniu, każde zapytanie wysłane do serwera będzie obsługiwane z wykorzystaniem szyfrowanego protokołu HTTPS. Aby skonfigurować przekierowanie należy otworzyć w edytorze tekstu plik konfiguracyjny */etc/apache2/sites-available/000-default.conf*. W pliku należy wprowadzić następującą konfigurację:

```
RewriteEngine On
```

```
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
```

Aby zmiany przyniosły oczekiwany rezultat należy uruchomić moduł **rewrite**. Można to zrobić wywołując polecenie:

```
sudo a2enmod rewrite
```

Aby zmiany konfiguracji zostały zastosowane, należy zresetować serwer np. poleceniem:

```
sudo /etc/init.d/apache2 restart
```

## Dostęp do plików

Aplikacja SOFM wykorzystuje szereg plików statycznych (obrazki, pliki css, js), dlatego muszą być one widoczne z poziomu serwera HTTP. Aby zwiększyć poziom bezpieczeństwa należy zmodyfikować konfigurację serwera HTTP tak aby udostępniana była jedynie niezbędna lokalizacja, oraz aby serwer nie umożliwiał podglądu struktury katalogów, w których umieszczone są pliki statyczne. W tym celu należy zmodyfikować konfigurację w pliku */etc/apache2/apache2.conf*. W pliku należy zlokalizować wpisy/znaczniki typu **Directory** i zamienić domyślną konfigurację, która w Apache na Ubuntu 14.04 wygląda następująco:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www >
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
```

na:



```
#<Directory />
#     Options FollowSymLinks
#     AllowOverride None
#     Require all denied
#</Directory>

#<Directory /usr/share>
#     AllowOverride None
#     Require all granted
#</Directory>

<Directory /var/www/sofm/sofm/static>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
```

Po zapisaniu zmian należy zresetować serwer HTTP np. wykonując polecenie:

```
sudo /etc/init.d/apache2 restart
```

### Uruchamianie, restart, zatrzymywanie aplikacji SOFM.

**Po prawidłowym skonfigurowaniu serwera HTTP, aplikacja SOFM jest z nim niejako zintegrowana. Oznacza to, że uruchamia się, wyłącza i resetuje razem z serwerem Apache. Dzięki temu, aby uruchomić/zresetować/zatrzymać aplikację SOFM wystarczy zastosować odpowiednią komendę serwera HTTP Apache:**

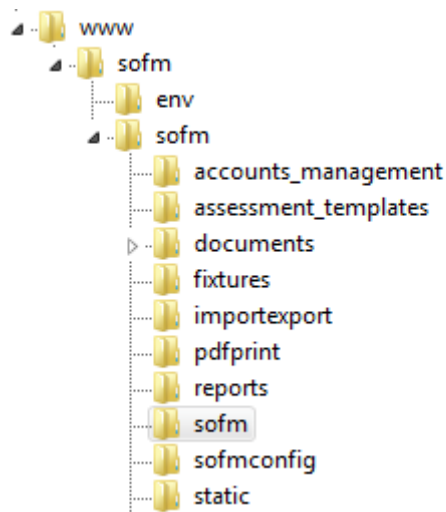
```
sudo /etc/init.d/apache2 start
sudo /etc/init.d/apache2 restart
sudo /etc/init.d/apache2 stop
```

Warto pamiętać, że serwer Apache jest domyślnie skonfigurowany w Ubuntu, tak aby uruchamiał się podczas startu systemu, dlatego w przypadku resetu systemu operacyjnego, aplikacja SOFM zostanie uruchomiona automatycznie razem z serwerem Apache.

## 3. Konfiguracja aplikacji SOFM

Głównym plikiem konfiguracyjnym aplikacji SOFM jest wspomniany wcześniej przy okazji instalacji i konfiguracji połączenia z bazą danych, plik **settings.py**. W ramach konfiguracji

aplikacji zostaną opisane jedynie wybrane parametry konfiguracyjne zawarte w tym pliku. Większość parametrów konfiguracyjnych, które nie są opisane w tym dokumencie, w trosce o poprawne funkcjonowanie systemu, nie powinna być modyfikowana. Plik **settings.py** znajduje się w katalogu aplikacji SOFM w podkatalogu *sofm*.



**Plik jest najważniejszym plikiem konfiguracyjnym aplikacji, dlatego bardzo istotne jest aby ewentualne zmiany były w nim przeprowadzane z możliwie największą ostrożnością. Zaleca się przestrzegania następujących zasad:**

- **Przed dokonaniem modyfikacji należy koniecznie wykonać kopię zapasową pliku.**
- Zaleca się aby nie wykonywać modyfikacji pliku w trakcie działania systemu. Może to spowodować problemy w dostępie do aplikacji w trakcie zmian, dlatego najlepiej zmiany wykonywać na kopii pliku, a gotową kopią ze zmianami nadpisywać konfigurację na serwerze.
- Plik jest skryptem napisanym w języku Python, dlatego podczas uruchamiania środowisko Python może utworzyć dla niego kopię tymczasową/skompilowaną, z której korzysta aplikacja SOFM. Takie pliki posiadają rozszerzenia \*.pyc i w celu osiągnięcia pewności, że zmiany zostaną od razu zastosowane, po zapisaniu modyfikacji należy je usunąć. Pozostawienie plików \*.pyc może spowodować brak efektów wprowadzonej modyfikacji.

__init__.py	0 KB	2016-02-27 11:48:16	rw-r--r--	sofmuser
__init__.pyc	1 KB	2016-02-27 23:20:04	rw-r--r--	sofmuser
settings.py	5 KB	2016-02-27 22:35:39	rw-r--r--	sofmuser
settings.pyc	5 KB	2016-02-27 23:20:04	rw-r--r--	sofmuser
urls.py	14 KB	2016-02-27 11:48:15	rw-r--r--	sofmuser
urls.pyc	15 KB	2016-02-27 23:20:04	rw-r--r--	sofmuser
wsgi.py	1 KB	2016-02-27 11:48:15	rw-r--r--	sofmuser
wsgi.pyc	1 KB	2016-02-28 00:51:08	rw-r--r--	sofmuser

**Po każdej zapisanej modyfikacji pliku settings.py należy wykonać reset aplikacji, korzystając np. z polecenia:**

```
sudo /etc/init.d/apache2 restart
```

### 3.1 Konfiguracja adresu e-mail

W pliku **settings.py** znajduje się konfiguracja adresu e-mail, z którego system wysyła wiadomości powiadomienia i wiadomości e-mail. Aby skonfigurować prawidłowo adres e-mail, należy wprowadzić w pliku podstawowe dane adresu oraz serwera, na którym dany adres e-mail jest zarejestrowany. Do konfiguracji adresu email służy sekcja pliku oznaczona tytułem **#Adresy email**. W ramach systemu istnieje możliwość skonfigurowania dwóch niezależnych adresów e-mail. W obecnej wersji systemu, do wysyłania wszystkich wiadomości, wykorzystywany jest jedynie podstawowy adres email.

```
# Adresy email

# Podstawowy adres email:
EMAIL_HOST = 'box.wrotapodlasia.pl'
EMAIL_PORT = 465
EMAIL_HOST_USER = 'sofm.admin@wrotapodlasia.pl'
EMAIL_HOST_PASSWORD = 'haslo'
EMAIL_USE_SSL = True
DEFAULT_FROM_EMAIL = 'SOFM <sofm.admin@wrotapodlasia.pl>'

#Dodatkowy adres email:
#INFO_EMAIL_HOST =
#INFO_EMAIL_PORT =
#INFO_EMAIL_HOST_USER =
#INFO_EMAIL_HOST_PASSWORD =
#INFO_EMAIL_USE_SSL =
#INFO_DEFAULT_FROM_EMAIL =
```

Poszczególne parametry konfiguracji adresu e-mail:

- EMAIL\_HOST – adres serwera SMTP dla konfigurowanego adresu e-mail
- EMAIL\_PORT – port serwera SMTP dla konfigurowanego adresu e-mail
- EMAIL\_HOST\_USER – nazwa użytkownika dla konfigurowanego serwera SMTP – zazwyczaj adres e-mail
- EMAIL\_HOST\_PASSWORD – hasło użytkownika dla konfigurowanego serwera SMTP
- EMAIL\_USE\_SSL – jeśli połączenie z serwerem odbywa się z wykorzystaniem SSL – należy wpisać wartość True, w przeciwnym wypadku, należy parametr zakomentować znakiem # lub skonfigurować mu wartość False
- EMAIL\_USE\_TLS – jeśli połączenie z serwerem odbywa się z wykorzystaniem TLS – należy wpisać wartość True, w przeciwnym wypadku, należy parametr zakomentować znakiem #, usunąć lub skonfigurować mu wartość False.
- DEFAULT\_FROM\_EMAIL – pełna nazwa adresu e-mail, z którego będą wysyłane wiadomości. Ta nazwa będzie wyświetlana w adresie nadawcy u odbiorców wiadomości, dlatego można tutaj wprowadzić dodatkowo podpis wysyłającego np. SOFM <sofm.admin@wrotapodlasia.pl>

Konfiguracja adresu dodatkowego jest analogiczna i będzie polegała na wprowadzeniu wartości dla analogicznych parametrów, na tej samej zasadzie.

Po zakończonej konfiguracji adresu e-mail należy wykonać reset aplikacji. Można to wykonać komendą:

```
sudo /etc/init.d/apache2 restart
```

### 3.2 Konfiguracja czasu trwania sesji

W pliku **settings.py** znajduje się również konfiguracja czasu trwania sesji, czyli czasu po którym użytkownik zostanie wylogowany z systemu w przypadku bezczynności. Parametr, odpowiadający za tą konfigurację znajduje się w sekcji oznaczonej tytułem **#Czas trwania sesji** i nazywa się **SESSION\_INACTIVE\_TIME**

```
#Czas trwania sesji  
SESSION_INACTIVE_TIME = 20*60 #czas w sekundach
```

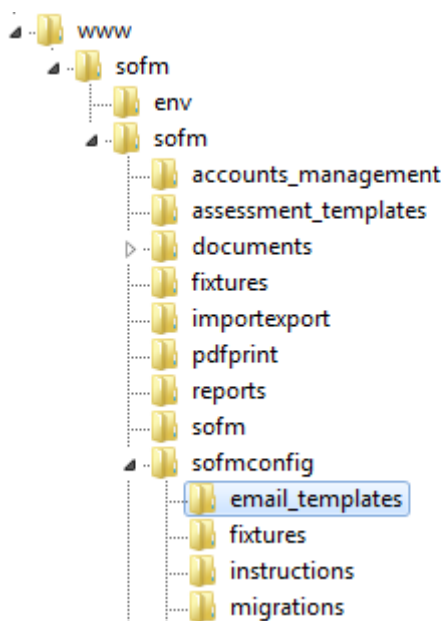
Czas należy wprowadzać w sekundach. Dozwolone jest konfigurowanie wartości za pomocą operacji arytmetycznych, tak jak na powyższym zdjęciu  $20\text{minut} = 20*60\text{sekund}$ , lub za pomocą wartości np. 1200. **Domyślna wartość czasu trwania sesji to 20 minut.**

Po zakończonej konfiguracji czasu trwania sesji, należy wykonać reset aplikacji. Można to wykonać komendą:

















```
sudo /etc/init.d/apache2 restart
```

## 4. Szablony wiadomości e-mail

Aplikacja przechowuje treści wysyłanych wiadomości email w tekstowych plikach szablonów. Szablony są napisane z wykorzystaniem kodu HTML i mogą zawierać znaczniki **%s**. W miejsce znaczników **%s** system przy wysyłce wstawia dane dotyczące konkretnej wiadomości (nazwy dokumentów, użytkowników itp.) zapewniając dzięki temu pewien poziom personalizacji korespondencji. Szablony wiadomości e-mail znajdują się w katalogu aplikacji w podkatalogu *sofmconfig/email\_templates*.



Domyślnie aplikacja SOFM wykorzystuje 14 szablonów wiadomości e-mail.

 add_application_document_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 add_recruitment_document_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 ass_card_accept_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 ass_card_back_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 ass_card_back_email_sec_copy.txt	2017-02-21 22:17	Plik TXT	1 KB
 ass_card_cancel_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 ass_card_complete_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 ass_card_start_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 assessor_in_app_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 lock_account_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 new_account_email.txt	2017-02-21 22:17	Plik TXT	2 KB
 password_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 send_question_email.txt	2017-02-21 22:17	Plik TXT	1 KB
 send_question_email_app.txt	2017-02-21 22:17	Plik TXT	1 KB
 send_question_email_app_copy.txt	2017-02-21 22:17	Plik TXT	1 KB
 send_question_email_copy.txt	2017-02-21 22:17	Plik TXT	1 KB

Każdy szablon może się różnić treścią. Edytując treść szablonu należy pamiętać o stosowanych w szablonach znacznikach. Znaczniki wypełniane są w określonej kolejności, a pod określonymi numerami znaczników znajdują się zawsze ściśle określone wartości. System domyślnie wysyła wiadomości w formacie HTML, natomiast dodatkowo wysyła też format tekstowy, na wypadek gdyby odbiorca nie był w stanie odebrać wiadomości z treścią HTML. Aby możliwie było formatowanie treści wiadomości, szablon musi być napisany z wykorzystaniem HTML.

1. **add\_application\_document\_email.txt** – szablon wiadomości informującej o umieszczeniu dokumentu we wniosku.

```
<p>Witaj,</p>
<p>We wniosku <strong> %s </strong> został umieszczony dokument <strong>
%s </strong></p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 2 znaczniki:

- pod pierwszym %s zostanie umieszczony numer wniosku
- pod drugim %s zostanie umieszczona nazwa dokumentu

2. **add\_recruitment\_document\_email.txt** – szablon wiadomości informującej o umieszczeniu dokumentu we wnioskach z poziomu naboru.

```
<p>Witaj,</p>
<p>W naborze <strong> %s </strong> został umieszczony dokument <strong>
%s </strong>. Dokument znajduje się w katalogu wniosku.</p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 2 znaczniki:

- pod pierwszym %s zostanie umieszczony numer naboru
- pod drugim %s zostanie umieszczona nazwa dokumentu

3. **ass\_card\_accept\_email.txt** – szablon wiadomości informującej o zmianie statusu karty oceny na „Ocena zaakceptowana”.

```
<p>Witaj,</p>
<p>Status Twojej karty oceny we wniosku <strong> %s </strong> został
zmieniony na <strong> %s </strong></p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 2 znaczniki:

- pod pierwszym %s zostanie umieszczony numer wniosku
- pod drugim %s zostanie umieszczona nazwa statusu karty oceny

4. **ass\_card\_back\_email.txt** – szablon wiadomości informującej o zmianie statusu karty oceny z „Ocena zakończona” na „Ocena w trakcie”. Szablon wysyłany oceniającemu.

```
<p>Witaj,</p>
<p>Status Twojej karty oceny we wniosku <strong> %s </strong> został
zmieniony na <strong> %s </strong></p>
<p><strong>Uzasadnienie cofnięcia karty:</strong></p>
<p>%s</p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 3 znaczniki:

- pod pierwszym %s zostanie umieszczony numer wniosku
- pod drugim %s zostanie umieszczona nazwa statusu karty oceny
- pod trzecim %s zostanie umieszczone uzasadnienie cofnięcia karty

5. **ass\_card\_back\_email\_sec\_copy.txt** – szablon wiadomości informującej o zmianie statusu karty oceny z „Ocena zakończona” na „Ocena w trakcie”. Szablon wysyłany sekretarzowi.

```
<p>Witaj,</p>
<p>Zmieniono status karty oceny oceniającego <strong> %s </strong> we
wniosku <strong> %s </strong> na <strong> %s </strong></p>
<p><strong>Uzasadnienie cofnięcia karty:</strong></p>
<p>%s</p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 4 znaczniki:

- pod pierwszym %s zostanie umieszczone imię i nazwisko oceniającego
- pod drugim %s zostanie umieszczony numer wniosku
- pod trzecim %s zostanie umieszczona nazwa statusu karty oceny
- pod czwartym %s zostanie umieszczone uzasadnienie cofnięcia karty

6. **ass\_card\_cancel\_email.txt** – szablon wiadomości informującej o zmianie statusu karty na „Ocena anulowana”.

```
<p>Witaj,</p>
```

```
<p>Status Twojej karty oceny we wniosku <strong> %s </strong> został
zmieniony na <strong> %s </strong></p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 2 znaczniki:

- pod pierwszym %s zostanie umieszczony numer wniosku
- pod drugim %s zostanie umieszczona nazwa statusu karty oceny

7. **ass\_card\_complete\_email.txt** – szablon wiadomości informującej o zmianie statusu karty na „Ocena zakończona”.

```
<p>Witaj,</p>
<p>Użytkownik <strong> %s </strong> zakończył ocenę wniosku <strong> %s
</strong> w systemie SOFM</p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 2 znaczniki:

- pod pierwszym %s zostanie umieszczony numer wniosku
- pod drugim %s zostanie umieszczona nazwa statusu karty oceny

8. **ass\_card\_start\_email.txt** – szablon wiadomości informującej o zmianie statusu karty na „Ocena w trakcie” z „Przed oceną”.

```
<p>Witaj,</p>
<p>Oceniający <strong> %s </strong> rozpoczął ocenę wniosku <strong> %s
</strong></p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 2 znaczniki:

- pod pierwszym %s zostanie umieszczony numer wniosku
- pod drugim %s zostanie umieszczona nazwa statusu karty oceny

9. **assessor\_in\_app\_email.txt** – szablon wiadomości informującej użytkownika o wskazaniu go do oceny wniosku.

```
<p>Witaj,</p>
<p>Zostałeś wskazany jako osoba oceniająca we wniosku <strong> %s
</strong> w systemie SOFM</p>
<p>Wiadomość wygenerowana automatycznie przez System Oceny Formalno-
Merytorycznej (SOFM)</p>
```

Szablon posiada 1 znacznik:

- pod %s zostanie umieszczony numer wniosku

10. **lock\_account\_email.txt** – szablon wiadomości informującej o zablokowaniu konta użytkownika.

```
<p>Szanowni Państwo,</p>
<p>informujemy, że użytkownikowi<strong> %s %s (login: %s)</strong>
zostało zablokowane konto w Systemie Oceny Formalno-Merytorycznej
(SOFM).</p>
```

```
<p>Wiadomość wygenerowana automatycznie przez SOFM – proszę na nią nie odpowiadać.</p>
<p>Pozdrawiamy</p>
<p>Zespół Administratorów Merytorycznych SOFM</p>
<p>Departament Rozwoju Regionalnego<br>
Referat Monitorowania i Systemów Informatycznych<br>
Urząd Marszałkowski Województwa Podlaskiego<br>
ul. Poleska 89, 15-874 Białystok</p>
```

Szablon posiada 3 znaczniki:

- pod pierwszym %s zostanie umieszczone imię użytkownika
- pod drugim %s zostanie umieszczone nazwisko użytkownika
- pod trzecim %s zostanie umieszczona nazwa użytkownika

11. **new\_account\_email.txt** – szablon wiadomości informującej użytkownika o założeniu dla niego konta w systemie SOFM.

```
<p>Szanowni Państwo,</p>
<p>W Systemie Oceny Formalno-Merytorycznej (SOFM) dla użytkownika <strong>%s %s</strong> utworzono konto o loginie <strong>%s</strong>. System dostępny jest pod adresem <a href="%s">%s</a></p>
<p>W celu uzyskania hasła należy kliknąć na przycisk Przypomnij hasło znajdujący się na stronie logowania, a następnie postępować zgodnie z komunikatami systemu. Hasło zostanie wysłane na adres mailowy podany w zgłoszeniu nadania uprawnień.</p>
<p>Prosimy o niezwłoczną aktywację konta. Jednocześnie informujemy, iż w przypadku długotrwałego braku logowania do systemu uprawnienia użytkownika ze względów bezpieczeństwa zostaną wycofane.</p>
<p>Wiadomość wygenerowana automatycznie przez SOFM – proszę na nią nie odpowiadać.</p>
<p>Pozdrawiamy</p>
<p>Zespół Administratorów Merytorycznych SOFM</p>
<p>Departament Rozwoju Regionalnego<br>
Referat Monitorowania i Systemów Informatycznych<br>
Urząd Marszałkowski Województwa Podlaskiego<br>
ul. Poleska 89, 15-874 Białystok</p>
```

Szablon posiada 5 znaczników:

- pod pierwszym %s zostanie umieszczone imię użytkownika
- pod drugim %s zostanie umieszczone nazwisko użytkownika
- pod trzecim %s zostanie umieszczony login użytkownika
- pod czwartym %s zostanie umieszczony adres aplikacji SOFM
- pod piątym %s zostanie umieszczony adres aplikacji SOFM

Adres aplikacji jest powtórzony w celu umożliwienia zbudowania aktywnego odnośnika HTML (<a>)

12. **password\_email.txt** – szablon wiadomości informującej użytkownika o wskazaniu tymczasowego hasła do systemu oraz linku aktywacyjnego.

```
<p>Witaj,</p>
<p>Do Państwa konta w SOFM wygenerowano następujące hasło dostępu: %s</p>
<p>Link aktywujący hasło do systemu: <a href="%s">%s</a></p>
```



```
<p>Wiadomość wygenerowano automatycznie - proszę na nią nie odpowiadać.
Jeśli nie korzystałeś z funkcji przypominania hasła systemu SOFM
skontaktuj się z naszym administratorem: <a href="mailto:%s">%s</a></p>
<p>Pozdrawiamy</p>
<p>Zespół Administratorów Merytorycznych SOFM</p>
<p>Departament Rozwoju Regionalnego<br>
Referat Monitorowania i Systemów Informatycznych<br>
Urząd Marszałkowski Województwa Podlaskiego<br>
ul. Poleska 89, 15-874 Białystok</p>
```

Szablon posiada 5 znaczników:

- pod pierwszym %s zostanie umieszczone wygenerowane tymczasowe hasło dostępu
- pod drugim i trzecim %s zostanie umieszczony wygenerowany link aktywacyjny
- pod czwartym i piątym %s zostanie umieszczony adres e-mail administratora systemu

Adres aplikacji oraz email jest powtórzony w celu umożliwienia zbudowania aktywnych odnośników HTML

13. **send\_question\_email.txt** – szablon wiadomości przekazującej treść wiadomości od użytkownika systemu do administratora. Funkcjonalności Wyślij pytanie.

```
<p>Witaj,</p>
<p>Użytkownik wysłał wiadomość z systemu SOFM.</p>
<p><b>Treść wiadomości:</b></p>
<p>%s</p>
<p><b>%s / email: <a href="mailto:%s" target="_top">%s</a></b></p>
```

Szablon posiada 4 znaczniki:

- pod pierwszym %s zostanie umieszczona treść wysyłanej wiadomości
- pod drugim %s zostanie umieszczona nazwa użytkownika
- pod trzecim i czwartym %s zostanie umieszczony adres email użytkownika

14. **send\_question\_email\_app.txt** – szablon wiadomości przekazującej treść wiadomości od oceniającego do sekretarza. Funkcjonalności Wyślij pytanie dostępna oceniającym na poziomie wniosku.

```
<p>Witaj,</p>
<p>Użytkownik wysłał wiadomość z systemu SOFM.</p>
<p><b>Treść wiadomości:</b></p>
<p>%s</p>
<p><b>%s / email: <a href="mailto:%s" target="_top">%s</a></b></p>
```

Szablon posiada 4 znaczniki:

- pod pierwszym %s zostanie umieszczona treść wysyłanej wiadomości
- pod drugim %s zostanie umieszczona nazwa użytkownika
- pod trzecim i czwartym %s zostanie umieszczony adres email użytkownika

15. **send\_question\_email\_app\_copy.txt** – szablon wiadomości wysyłanej użytkownikowi jako kopia w wyniku skorzystania z mechanizmu wysyłającego wiadomość od oceniającego do sekretarza. Funkcjonalności Wyślij pytanie dostępna oceniającym na poziomie wniosku.

```
<p>Witaj,</p>
<p>Wysłałeś wiadomość z systemu SOFM.</p>
<p><b>Treść wysłanej wiadomości:</b></p>
<p><i>%s</i></p>
```

Szablon posiada 1 znacznik:

- pod %s zostanie umieszczona treść wysyłanej do sekretarza wiadomości

16. **send\_question\_email\_copy.txt** – szablon wiadomości wysyłanej użytkownikowi jako kopia w wyniku skorzystania z mechanizmu przekazującego treść wiadomości od użytkownika systemu do administratora. Funkcjonalności Wyślij pytanie.

```
<p>Witaj,</p>
<p>Wysłałeś wiadomość z systemu SOFM.</p>
<p><b>Treść wysłanej wiadomości:</b></p>
<p><i>%s</i></p>
```

Szablon posiada 1 znacznik:

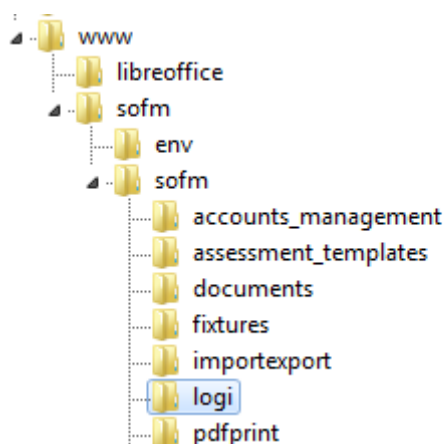
- pod %s zostanie umieszczona treść wysyłanej do administratora wiadomości

**Po każdej zapisanej modyfikacji pliku settings.py należy wykonać reset aplikacji, korzystając np. z polecenia:**

```
sudo /etc/init.d/apache2 restart
```

## 5. Logi aplikacji SOFM

Aplikacja SOFM generuje logi własnego stanu pracy. Logi stanu pracy aplikacji są umieszczane w pliku umieszczonym w katalogu głównym aplikacji w podkatalogu *logs*. Plik logów domyślnie nosi nazwę *sofm.log*.



Aplikacja tworzy i wypełnia 1 plik logów. W pliku pisane są wszystkie logi generowane przez aplikację. **Należy systematycznie kontrolować stan pliku, tak aby jego rozmiar nie był zbyt duży.**

W pliku logów rejestrowane są:

1. Informacje o błędach lub zagrożeniach wynikających z funkcjonowania aplikacji SOFM

## 2. Informacje o próbach nieuprawnionego dostępu do bazy danych, w tym:

- Informacje o próbach nieuprawnionego dostępu do edycji danych
- Informacje o próbach pobrania plików dokumentów, z ominięciem mechanizmu prezentowania dokumentów aplikacji SOFM
- Informacje o próbach wydruku dokumentu (informacja jest rejestrowana w momencie kliknięcia przycisku Drukuj w przeglądarce)
- Informacje o potencjalnym uruchomieniu przez użytkownika, narzędzi do analizy kodu strony

Standardowy, przykładowy wpis w logach ma następującą postać:

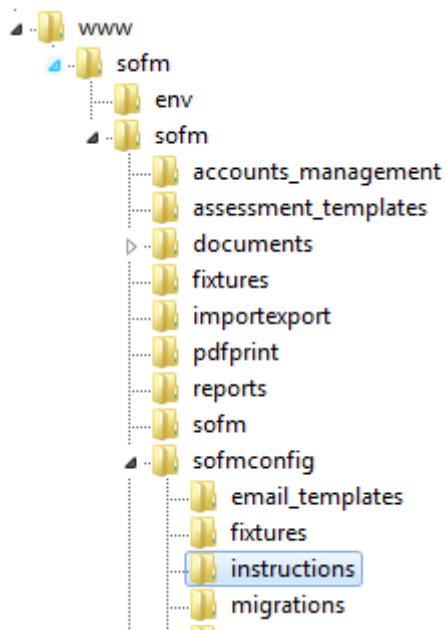
```
2016-02-26 22:05:25.359 WARNING views - page_not_permissions: Uwaga!!! Próba nieuprawnionego dostępu. Id użytkownika: 59. Zdarzenie: /edycjakatalogu/197/
```

Wpis składa się z dokładnej daty wystąpienia zdarzenia, rodzaju zdarzenia (ERROR, WARNING, INFO), opisu modułu w którym wystąpiło zdarzenie, opisu zdarzenia (w tym w zależności od typu zdarzenia np. id użytkownika oraz informacji co próbował zrobić)




Powyższy wpis w logu może sugerować próbę edycji katalogu o id 197 przez użytkownika o id 59, nie posiadającego odpowiednich uprawnień do tego celu.

## 6. Udostępniane Instrukcje

Aplikacja SOFM udostępnia użytkownikom instrukcje, w zależności od profilu, z jakiego korzysta dany użytkownik. Instrukcje udostępniane są do pobrania lub odczytu w formie plików \*.pdf. Fizycznie pliki znajdują się w strukturze plików w katalogu aplikacji SOFM w lokalizacji *sofmconfig/instructions*.



System obsługuje trzy instrukcje umieszczone w tej lokalizacji.

 Instrukcja użytkownika SOFM - Administrator.pdf	2 768 KB	2016-02-18 09:13:40	rw-r--r--	sofmuser
 Instrukcja użytkownika SOFM - oceniający.pdf	1 078 KB	2016-02-27 11:48:39	rw-r--r--	sofmuser
 Instrukcja użytkownika SOFM - Sekretarz.pdf	1 864 KB	2016-02-27 11:48:40	rw-r--r--	sofmuser

Bardzo istotne jest aby umieszczane tutaj instrukcje miały dokładnie następujące nazwy. **Istotna jest również dokładnie ta sama wielkość liter:**

- **Instrukcja użytkownika SOFM - Administrator.pdf** - jest to plik instrukcji administratora
- **Instrukcja użytkownika SOFM - oceniający.pdf** – jest to plik instrukcji oceniającego
- **Instrukcja użytkownika SOFM - Sekretarz.pdf** – jest to plik instrukcji sekretarza

**Jedynie poprawnie nadane nazwy umożliwią pobieranie tych instrukcji przez użytkowników**

## 7. Polityka Bezpieczeństwa i licencja

Aplikacja SOFM udostępnia politykę bezpieczeństwa w dwóch miejscach i formach. Po rejestracji użytkownika w systemie w formie przewijanej listy, oraz w menu **Pomoc** w formie pliku \*.pdf do pobrania. Dokumenty źródłowe, z których korzystają wymienione mechanizmy znajdują się w katalogu aplikacji SOFM w tej samej lokalizacji co pliki instrukcji, czyli w *sofmconfig/instructions*.

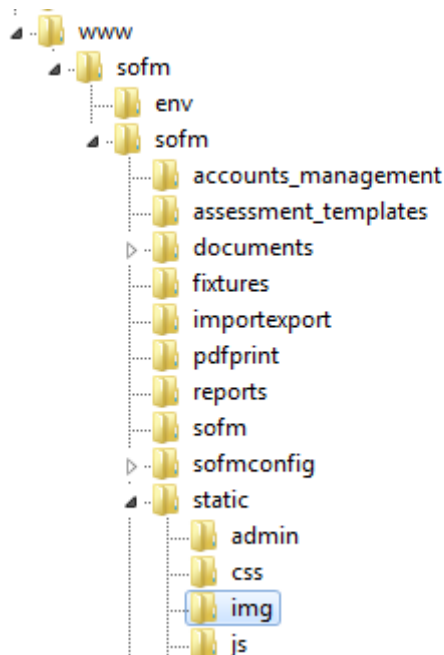
 Polityka Bezpieczeństwa Informacji - SOFM.pdf	238 KB	2016-02-27 11:48:39	rw-r--r--	sofmuser
 privacy.txt	43 KB	2016-02-27 11:48:39	rw-r--r--	sofmuser

Polityka bezpieczeństwa znajduje się w dwóch plikach.

- **Polityka Bezpieczeństwa Informacji - SOFM.pdf** – to plik dostępny do pobrania z menu Pomoc. Podobnie jak w przypadku instrukcji **jego nazwa nie może być zmieniona, aby system mógł go udostępnić.**
- **privacy.txt** – to tekstowa wersja polityki bezpieczeństwa. Z tego pliku korzysta mechanizm wymuszający akceptację Polityki Bezpieczeństwa po pierwszym logowaniu do systemu. Taka forma jest wymagana, aby poprawnie wyświetlić treść polityki w przewijanym oknie widocznym po pierwszym logowaniu do systemu

## 8. Grafiki UE wykorzystywane na wydrukach

Aplikacja wykorzystuje grafiki UE w stopce oraz na wydrukach kart oceny. Grafiki, podobnie jak wszystkie elementy widoczne w interfejsie przeglądarkowym aplikacji, znajdują się w katalogu aplikacji SOFM w lokalizacji *static/img/*



Aplikacja wykorzystuje następujące pliki UE:

- **logo\_ue\_main.jpg** – główny plik umieszczany w stopce interfejsu przeglądarkowego aplikacji
- **general\_standard\_logo.jpg** – plik w skali szarości umieszczany na wydrukach kart oceny z zaznaczoną opcją „Logotypy ogólne”
- **general\_bof\_logo.jpg** – plik w skali szarości umieszczany na wydrukach kart oceny z zaznaczoną opcją „Logotypy ogólne” oraz „ZIT BOF”
- **efrr\_standard\_logo.jpg** – plik w skali szarości umieszczany na wydrukach kart oceny w kartach EFRR (brak dodatkowych zaznaczeń)
- **efrr\_bof\_logo.jpg** – plik w skali szarości umieszczany na wydrukach kart oceny z zaznaczoną opcją „ZIT BOF” (bez zaznaczenia opcji „Logotypy ogólne”) w kartach EFRR
- **efs\_standard\_logo.jpg** – plik w skali szarości umieszczany na wydrukach kart oceny w kartach EFS (brak dodatkowych zaznaczeń)
- **efs\_bof\_logo.jpg** – plik w skali szarości umieszczany na wydrukach kart oceny z zaznaczoną opcją „ZIT BOF” (bez zaznaczenia opcji „Logotypy ogólne”) w kartach EFS

**Przy podmianie plików również bardzo istotne jest zachowanie niezmienionej nazwy pliku. Dotyczy to również wielkości liter. Każda zmiana nazwy uniemożliwi poprawne używanie tych plików przez aplikację SOFM**

## 9. Kopie bezpieczeństwa

Aplikacja SOFM przechowuje dane w strukturze katalogów systemu operacyjnego oraz w bazie danych. Zalecane jest tworzenie kopii bezpieczeństwa danych i kopiowania ich poza serwer aplikacji SOFM w celu ich zabezpieczenia na wypadek awarii.

## Wirtualizacja

Jeśli system SOFM jest zainstalowany na maszynie wirtualnej, zalecane jest tworzenie kopii bezpieczeństwa całej maszyny wirtualnej. Takie podejście zapewni możliwie najkrótszy czas przywrócenia funkcjonalności systemu w przypadku wystąpienia awarii. Do tego celu można wykorzystać wbudowane możliwości środowiska wirtualizującego lub narzędzia dodatkowe typu Veeam

## 9.1 Kopia bezpieczeństwa plików aplikacji

Pierwszym krokiem prowadzącym do zabezpieczenia danych aplikacji SOFM jest tworzenie kopii bezpieczeństwa plików aplikacji oraz plików dokumentów importowanych do systemu.

1. Pliki aplikacji znajdują się w katalogu aplikacji SOFM. Jeśli instalacja została przeprowadzona zgodnie z wytycznymi opisanymi w tym dokumencie, katalog aplikacji powinien znajdować się w `/var/www/sofm/sofm`. Katalog aplikacji powinien zostać zachowany w całości. **Kopia może być wykonywana jednorazowo po każdej aktualizacji serwera lub cyklicznie (jeśli istotne jest zachowywanie logów serwera)**
2. Pliki importowanych do systemu dokumentów są domyślnie przechowywane w lokalizacji `/sofm/document_files`. Należy cyklicznie wykonywać kopię zapasową całego katalogu. Jest ona niezbędna do odtworzenia dokumentów w systemie po awarii. Zaleca się wykonywanie kopii bezpieczeństwa tych danych co najmniej raz na dobę.

## 9.2 Kopia bezpieczeństwa bazy danych

Do poprawnego przywrócenia aplikacji po ewentualnej awarii niezbędne jest wykonywanie kopii bezpieczeństwa bazy danych aplikacji. Kopię można wykonywać ręcznie np. za pomocą graficznych narzędzi do zarządzania bazą danych typu **phpMyAdmin**. Innym narzędziem do wykonywania kopii bezpieczeństwa bazy danych MySQL może być dostępne w systemach typu Linux, konsolowe narzędzie **mysqldump**. Za jego pomocą można wykonywać kopię bazy danych ręcznie lub automatycznie np. poprzez umieszczenie odpowiedniego skryptu w dostępnym w systemach Linux oprogramowaniu **Cron**. Korzystając z narzędzi **Cron** oraz **mysqldump** można zaplanować cykliczne tworzenie kopii zapasowej bazy danych i kopiowanie jej do wyznaczonej lokalizacji. **Zaleca się uruchomienie procedury cyklicznego tworzenia kopii zapasowej bazy danych, która wykonywałaby się co najmniej raz na dobę, np. w nocy, kiedy obciążenie systemu zazwyczaj jest najniższe.**

**UWAGA! W celu zachowania spójności danych tworzonej kopii bezpieczeństwa aplikacji, kopia bezpieczeństwa plików z katalogu `/sofm/document_files` oraz kopia bezpieczeństwa bazy danych muszą być wykonywane w tym samym czasie. Np. tym samym skrypcem uruchamianym za pomocą Cron raz na dobę.**

## 9.3 Przywrócenie systemu z kopii

W przypadku konieczności przywracania systemu po awarii z kopii zapasowych plików, oraz bazy danych, przy założeniu, że do dyspozycji jest jedynie serwer z czystym systemem operacyjnym, należy:

1. Przeprowadzić procedurę instalacji systemu zgodnie z punktem 2 niniejszej instrukcji, uwzględniając następujące modyfikacje tej procedury:

- a) **Wszystkie pliki kopii zapasowych muszą być ze sobą zgodne tj.**
- Kopia katalogu aplikacji musi być aktualna – aplikacja gromadzi w swojej strukturze plików informacje o strukturze bazy danych, dlatego aby aplikacja po przywróceniu funkcjonowała prawidłowo, **musi być to kopia bezpieczeństwa wykonana np. po ostatniej aktualizacji, nie wcześniejsza**
  - kopia plików z katalogu */sofm/document\_files* musi być zgodna z kopią bezpieczeństwa bazy danych. W bazie danych trzymane są informacje umożliwiające interpretację struktury plików w katalogu */sofm/document\_files* dlatego kopie bezpieczeństwa tych elementów powinny być wykonywane w miarę możliwości w tym samym czasie
- b) Baza danych musi być wytworzona i zasilona danymi pochodzącymi z kopii bezpieczeństwa. Można to zrobić korzystając z narzędzi typu phpMyAdmin lub narzędzi dostępnych z poziomu linii komend
- c) **Należy pominąć kroki procedury instalacji aplikacji dotyczące wytworzenia struktury bazy danych, zasilenia bazy danych danymi początkowymi oraz tworzenia pierwszego użytkownika.**
- d) Instalując aplikację SOFM należy tworząc katalog główny aplikacji umieścić w nim kod aplikacji z przywracanej kopii bezpieczeństwa