

*Anastas N. Madzharov,*

## **TECHNICAL IMPLEMENTATION OF A REPORTING SYSTEM AND ITS WORKFLOWS**

**Anastas N. Madzharov**

*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia,  
a.madzharov@acad.bg*

**Abstract:** *This paper discusses the architectural decisions, flows and the implementation details of a reporting system and its workflows, as it is used within NMU. The system has been delivered as part of project BG051PO001-3.1.08-0024 "Improvements on the administrative systems of National Military University "V. Levski" (NMU). The paper is limited to the initial implementation of the reporting system and its workflows, while recent improvements on its code base and architecture are out of scope. Details are shared on its integration with reference database, especially the integration with the hierarchical employee structure of NMU "Vasil Levski". It is not a user guide. The implementation details are low level and aim at facilitating the long-term maintenance and improvements of the system. The main features of the system are (i) reports templating, input and archiving; (ii) daily summary reports and (iii) tracking the yearly holiday allowances for all employees.*

**Keywords:** *Document Management Systems (DMS), Content Management Systems (CMS)*

## **ТЕХНИЧЕСКА РЕАЛИЗАЦИЯ НА СИСТЕМА ЗА ДОКЛАДИ И НЕЙНИТЕ РАБОТНИ ПОТОЦИ**

**Анастас Н. Маджаров**

*Секция „Високопроизводителни системи, мрежи и алгоритми“, Институт по информационни и комуникационни технологии, Българска академия на науките, ул. Акад. Г. Бончев, блок 25А, София, a.madzharov@acad.bg*

### **Въведение**

В тази статия се описва модула за молби и доклади от „Системата за управление на качеството“ в НВУ, към момента на неговото създаване. Разработван е по проект BG051PO001-3.1.08-0024 „Усъвършенстване на системите за управление в НВУ „В. Левски“. В настоящето описание не са взети предвид последващи изменения в структурата и кода на модула. Дават се подробности за интегрирането му с референтна бази данни, по-специално интеграцията с йерархичната структура на служителите в НВУ „Васил Левски“ [1]. То не е ръководство за потребители.

Целта на описанието е да се подпомогне поддръжката и експлоатацията на модула и евентуално неговото допълване и развитие.

Назначението на модула за доклади е да реализира въвеждане, резолиране, архивиране и отчетност на ежедневните доклади, да изчислява и отчита годишните отпуските на личния състав.

Системата поддържа над 60 бланки за доклади, които са разпределени според административните изисквания за водене на докладна книга. Всяка от бланките е реализирана чрез PHP мо-

дула Smarty, който поставя основата на динамично изграждане на документи чрез анотация на по-малки фрагменти. Всека бланка е в отделен php файл, като е разработена система за кодиране имената на файловете. Нова бланка се добавя като нов файл, с ново име. Това кодиране позволява при нормативни промени, старите доклади да се съхраняват в оригинален вид, а новите да посътпват с промените. Всеки потребител вижда само тези формуляри, които са му необходими. Този принцип също е заложен в системата за кодиране на имената на файловете.

## 1. Изграждане и съхраняване на доклад (молба)

Smarty е модул на PHP, който динамично попълва бланки на документи. Той предлага кеширане на документите във файловата система на сървъра, като това предотвратява ненужно натоварване на базата данни, както и предлага по-бързо показване на документите в браузър.

Моля на основание чл. 104 Правилника за вътрешния ред и дисциплината на НВУ „Васил Левски“ да се разреши на **{EmployeeName}** ваканция в размер на 10/20 (**{DaysString}**) дни за учебната **{AcademicYear}** г., за времето **{HolidayDates}**. Ваканцията ще прекара в **{Address}**.

Моля същият да се отчисли от котлова храна от **{Meal}** на **{Date}** г. по приложение №2 от МЗ №ОХ-482/30.07.2010 г.

**Фигура 1:** Вид на бланка, съхранявана в базата данни като файл

Променливите от бланката се попълват по времена на заявката от браузъра както следва:

```
$this->smarty = new Smarty;  
$smarty->assign(EmployeeName, "Иван Иванов Димитров");  
$this->smarty->display("string:" . $template);
```

### 1.1. Списък на разработените фрагменти за изграждане на бланка

Базовата функционалност на Smarty е недостатъчна за постигане целите на системата за доклади и са разработени множество разширения към модула, за да се поддържат различните типове доклади. Всичките тези разширения се наричат фрагменти за изграждане на бланка и основно имат две цели: 1) да визуализират фрагмента според изискванията, както и да 2) предложат специфична функционалност; например – търсене на курсант по име, падащи менюта за избиране на заместник, година за отпуск и други.

Според примера от фиг. 1, името “Иван Иванов Димитров” няма да е попълнено като текст в кода, а се използва подходящ фрагмент от допълнително разработените, за извличането му от модула „Личен състав“:

```
$this->smarty = new Smarty;  
EmployeeName = new \view\EmployeeNameView();  
$smarty->assign(EmployeeName, EmployeeNameView->display($nameFromDatabase));  
$this->smarty->display("string:" . $template);
```

### Най-важните фрагменти за системата са следните:

- EmployeeName - Показва името на потребител в поле, което не може да се редактира.
- HolidayEmployeeName - Същото като EmployeeName, но се използва за бланки които са предназначени за отпуск и в които се следи календарното време.
  - EmployeeNameSearch - Поле за търсене на име на потребител от базата данни.
  - HolidayEmployeeNameSearch - Същото като EmployeeNameSearch, но в бланки, предназначени за отпуск и в които се следи календарното време.
  - DatePicker - Поле за избиране на дати, с мини-календар като визуализация. Позволява да се избират дати от днес нататък.
  - DatePickerUnbound - Същото като DatePicker, но позволява да се избират и отминали дати, отпреди днешния ден.
  - HolidayDatesPicker - Група от два DatePicker-а, които са предназначени за избиране на начална и крайна дата на отпуск.
  - HolidayDatesPickerUnbound - Същото като HolidayDatesPicker, но позволява да се избират и отминали дати в първото поле за начална дата.
  - OptionalParagraph - Използва се за обозначаване на специален параграф, чийто текст да може да се скрива.
  - HolidayDeputyDropDown - Падащо меню за избор на заместник.
  - HolidayYearDropDown - Падащо меню за избиране на годината, за която да се изчерпва заявения отпуск.
  - TerminateDatesPicker - Същото като HolidayDatesPicker, но обозначава дати за прекъсване на отпуск. Тези дати са ограничени между датите на първоначално одобрения отпуск.
  - InputField - Стандартно поле за редактиране.
  - UnmodifiableInputField - Стандартно поле за показване на текст, който не може да бъде редактиран.

### Реализация на фрагментите в код

Типовете фрагменти са дефинирани в таблицата dokladi\_templateitemtype от базата данни.

Самите бланки се съхраняват в таблицата dokladi\_template, в която полето contents съответства на бланката, така както Smarty я изисква.

Всяка бланка се асоциира с фрагменти от подходящ тип и това е реализирано чрез таблицата dokladi\_templateitem.

Веднъж дефинирани, фрагментите са имплементирани с код, който е комбинация от PHP и JavaScript за визуализацията им.

Основният PHP пакет за работа с бланка е във директория /view/. Примерна реализация на фрагмент е:

```
abstract class AbstractFieldView {  
  
    public abstract function displayAsHtml(\Entities\TemplateItem $templateItem, $defaultValue = "");  
  
    public function displayInline(\Entities\TemplateItem $templateItem) {  
        return $defaultValue;  
    }  
}  
  
class InputFieldView extends AbstractFieldView {  
  
    public function displayAsHtml(\Entities\TemplateItem $templateItem, $defaultValue = "") {  
        return "<input type='text' name='".$templateItem->name.'" "  
            ."value='".$defaultValue.'" "  
            ."class='tooltip borderless' title='".$templateItem->tooltipText.'" "  
            ."data-autosize-input='{ \"space\": 20}' "  
            ."style='transition: width 0.25s; min-width: ".$(templateItem->size)."'";  
    }  
}
```

```
        ."/>";  
    }  
}
```

Функцията `displayAsHtml` се използва да покаже фрагмента в бланката по време на нейното попълване или редактиране, докато `displayInline` показва същият фрагмент, но като текст във вече попълнен доклад.

Много от фрагментите са допълнени с JavaScript, който допринася за функционалността на потребителския интерфейс. Например, падащото меню за заместници (`HolidayDeputyDropDown`) се презарежда всеки път, когато датите за отпуск се променят в полетата на бланката (`HolidayDatesPicker`); промяната на датите води до преизчисляване на работните дните за които се иска отпуск и резултатът се показва на потребителя; полетата за прекратяване на отпуск не са активни, докато не се намери необходимия доклад за поискан отпуск, който ще се прекратява чрез въвеждане на неговия регистрационен номер, присвоен от системата след утвърждаването му от последна инстанция; и други.

## 1.2. Списък на типовете бланки

Типът на всяка бланка е зададен според това за какъв вид отпуска ще бъде съответния доклад. Типовете са:

- `HOLIDAY` – Отпуск.
- `UNTRACKED_HOLIDAY` – Друг вид отпуск, в която не се следят и изчисляват работни дни и не се изваждат от полагаемите се за избраната годината.
- `ADDITIONAL_LEAVE` – Допълнителен отпуск за балнеолечение и/или отпуснат според Кодекса на труда. Дните се следят.
- `ADDITIONAL_LEAVE_STUDY` – Допълнителен отпуск за обучение. Дните се следят заедно с тези от типа `ADDITIONAL_LEAVE`.
- `SICK_LEAVE` – Болничен.
- `BUSINESS_TRIP` – Командировка.
- `NOT_HOLIDAY` – Друг тип доклади, различни от горните типове, обикновено такива, в които няма периоди от време.
- `LEFTOVER_OVERRIDE` – Останали дни отпуск за годината.
- `COMPESATION` – резервиран за бъдещо развитие на системата при отчитане на компенсации от наряд или занятия през почивни дни.

Типовете на бланки за доклади са описани в таблицата `dokladi_holiday_type`. Съответно, всяка бланка е анотирана със своя тип в таблицата `dokladi_template` (`holidayTypeId`). Допълнително, всяка бланка е обозначена с ролята на потребител, който може да я упражнява. Те са: 0 – общи; 1 – военнослужещ; 2 – курсант; 3 – кадет; 4 – цивилен служител; 5 – студент (резервиран за бъдещо развитие).

## 2. Функционални възможности за работа с доклади

### 2.1. Докладът като конкретна инсталация на бланка

Докладът представлява инсталация на бланка (фиг. 2) в браузер, която има конкретно зададени стойности за променливите и полетата. Докладите се записват основно в две таблици – 1) `dokladi_request`, която съдържа мета информация за доклада – коя бланка е използвана, кой и кога я е пуснал и т.н.; и 2) `dokladi_requestdetails`, която е аналогична на описаната по-горе `dokladi_templateitem` и още съдържа стойностите на фрагментите, които потребителят е задал.

Тук, е важно да се отбележи, че не всички стойности на фрагменти се записват в `dokladi_requestdetails`, а само тези, които са за фрагменти незапочващи с името `Holiday`. Тези, кои-

то започват с Holiday, специално се комбинират и обработват от системата, след което се записват в таблица dokladi\_holiday. Тази таблица е подробен запис на зададен отпуск с дати; заместници; година на отпуска; работните дни за които е поискан отпуск и които са калкулирани по време на записа на молбата; и др.

Фигура 2: Вид на доклад с полетата за попълване

## 2.2. Преглед на доклад

Детайлите на доклада, които са записани в dokladi\_request се използват за директното показване на списъка с доклади за всеки потребител. Това става от менютата “Молби” или “Мои молби”, като е реализирано в стил на пощенска кутия. Молбите в тъмно са тези, които логнатия потребител все още не е подписал (фиг. 3).

Молба	От	Дата на създаване	Статус
Отпуск по семейни причини	полк. доц. д-р инж. Анастас Николов Маджаров	04.05.2015 20:04	Одобрена
Платен годишен отпуск	полк. доц. д-р инж. Анастас Николов Маджаров	04.05.2015 19:53	Нерезолирана

Фигура 3: Вид на справка за собствените молби

Подробните детайли от таблиците dokladi\_holiday и dokladi\_requestdetails се извличат от базата данни и цялата молба се показва на потребителя при поискване (избиране от падащо меню).

При прегледа на молба, според йерархията на потребителя, молбата може да бъде редактирана от него, преди да я подпише. При редакцията на молбата, бланката се показва, както при началното подаване на молбата, но полетата са попълнени според зададените им стойности. Потребителите от първо ниво на йерархията вместо бутон „Редактирай“ имат бутон „Преглед“, който показва молбата във форматиран окончателен вид.

По същество, кодът който визуализира бланката и молбата е един и същ, независимо от режима – пускане на нова молба, преглед при пускане, редактиране на молба, гледане на молба. По

този начин се избягват бъгове при форматиране и изгледът на дадена бланка се запазва еднакъв през целия и живот като молба.

Моля, съгласно утвърден рапорт вх. № / и на основание чл. 197, ал. 1, т. 1 и т. 2 от ЗОВС и съгласно план-графика за отпуските да се разреши на полк. доц. д-р инж. Анастас Николов Маджаров () дни платен годишен отпуск за 2015год. за времето от 05.05.2015 до 07.05.2015. Отпуската ще прекара на адрес Варна, комплекс-Приморски.

Пътуването ще извърши с кола от Плевен до Варна и обратно.

Моля да бъде отдаден/а в заповед да изпълнява задълженията му/и по време на неговото/нейното отсъствие подп. д-р инж. Стефан Савов Билидеров.

Забележка: Доклад се извършва само при наличие на положително резолиран от Началника на Училищната рота.

✓ Одобри    ✗ Откажи

✎ Редактирай

Въведете причина за отказ/одобрение...

Декан на факултет  
● полк. доц. д-р инж. Марин Симеонов Маринов  
● Началник на катедра  
● полк. доц. д-р инж. Анастас Николов Маджаров

🖨️ Принтирай

Затвори

**Фигура 4:** Преглед на молба с възможност за отпечатване, одобряване, редактиране, въвеждане на резолюция или отхвърляне

От гледна точка на реализация на функционалността, се използват два главни пакета от PHP код. Те са обособени в следните директории:

- /entities/ - са тези, които правят връзка с базата данни и конструират обекти директно от нея. Така всеки ред от таблиците в базата от данни се реализира като PHP обект.
- /domain/ - са тези, които преобразява обекти от /entities/ към обекти, с които системата ефективно може да следи отпуски, потребители, йерархия, коментари, да извършва визуализации и т.н.

### 2.3. Фази на живот за доклад (молба)

След като попълнена бланка се подпише, тя става молба и в зависимост от това как се придвижва по йерархията, може да премине през няколко фази:

- **Нерезолирана и чакаща подпис** – Логнатият потребител е получил молба от подчинен, която не е подписал все още (фиг. 4).
- **Нерезолирана** – Логнатият потребител е подписал молбата, но тя не се е придвижила до края на йерархията и все още е нерезолирана (фиг. 3 – текстът в тъмно).
- **Одобрена** – молба одобрена от всички в йерархията. Веднъж одобрена молбата се замразява – не може да се променя повече, независимо от бъдещите промените в йерархията. Ако молбата е за отпуск, работните дни, които съдържа отпуска се отчитат в момента в който молбата преминава в одобрена. В този момент и се присвоява и номер в системата. Независимо от възможни бъдещи промени в националните празници, дните които отпускът от тази молба покрива няма да се променят.
- **Отказана** – молба, може да бъде отказана на всяко ниво над първото в йерархията. Отпускът който е поискан чрез молбата ще остане вписана в dokladi\_holiday, но няма да бъде отчетен. Веднъж отказана, молбата не може да се върне към по-ранен стадий. Опции за редактиране и манипулиране не се предлагат.

### Заклучение

Изградената система за доклади е реализирана за конкретните изисквания на потребителя. Предварително проведеното проучване на съществуващи Content Management Systems и документооборот с възможност за изчисляване и отчитане на годишните отпуските на личния състав не предоставяха необходимата функционалност и затова се премина към разработване на собствена, уникална структура на модула „Доклади“ от информационната система на Национален военен университет.

За да се поддържа и развива вече изградена система е необходимо да се познават детайлите от нейната структура и реализирани функционалности. Настоящата статия предоставя тази възможност.

## **Литература**

1. Свилен Стефанов, Т. Йотков, Ц. Харизанов. Единен функционален модел на Национален военен университет „Васил Левски“, ПРОЕКТ BG051PO001-3.1.08-0024, Велико Търново, 2013.