

УДК 004.6

*А.А. Козинский, В.В. Охримчук*

## **РЕАЛИЗАЦИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ПРОГРАММНОЙ СИСТЕМЫ УПРАВЛЕНИЯ УЧЕТОМ ВОЕННООБЯЗАННЫХ**

Работа посвящена созданию модели автоматизированной программной системы управления учетом военнообязанных. Освещено решение следующих задач: создание инфологической модели базы данных, масштабирование текстовых полей печатных форм, построение элементов интерфейса, справочников, системы поиска и отображения информации, сетевого пользовательского интерфейса. Значительное внимание уделено элементам технического задания, выбору программных средств реализации моделей системы управления. Объектом автоматизации системы «Учет военнообязанных» является комплекс задач по постановке военнообязанных на учет, созданию извещений о постановке на учет для пересылки по месту прежнего учета, формирование команд и др. Описаны фрагменты программных кодов, реализующих решение поставленных задач, скриншоты диалоговых страниц.

В 2011 году перед авторами была поставлена задача создания автоматизированной системы управления учетом военнообязанных. Заказчиком выступил Брестский городской военный комиссариат. Специфика ожидаемого решения определялась тем, что программная система (ПС) на всех этапах проектирования оперирует данными, которые непосредственно для разработчиков не могут быть конкретизированы. Вся информацию базы данных, как правило, составляют сведения, не подлежащих разглашению. Другими словами, разработчики имеют дело только с абстрактными данными и моделями их обработки. Конкретное наполнение информационной системы и обработка данных в соответствии с поставленной задачей выполняется работниками военкомата. Такие условия разработки налагают дополнительные требования к надежности системы, ее структуре и интерфейсу, обеспечению защиты информации от несанкционированного доступа, к сетевому доступу пользователей.

В настоящей работе представлено краткое описание основных этапов моделирования программной системы управления учетом военнообязанных. Их изложение представляет интерес, во-первых, как поэтапное описание реализации математической модели системы управления, во-вторых, как задача, требующая применения нестандартных решений, моделей и средств современных технологий программирования.

Дадим общее описание предметной области, являющееся компонентом технического задания по созданию системы управления.

Объектом автоматизации системы «Учет военнообязанных» является комплекс задач по постановке военнообязанных на учет, созданию извещений о постановке на учет для пересылки по месту прежнего учета, формирование команд, формирование основных составов команд с разбиением по группам (литерам), формирование организационного ядра на базе основного состава команд, формирование резервного состава команд, формирование предварительных пунктов сбора военнообязанных (ППСВ), формирование повесток о сборах на ППСВ с учетом временных интервалов сбора, формирование участков оповещения, формирование маршрутов оповещения для участков, формирование мобилизационных предписаний, формирование приписных карт, формирование именных списков. Цель автоматизации – сокращение большого объема ручных трудозатрат для ведения информации и отчетных документов по решению комплекса задач учета военнообязанных, задач формирования команд,

участков, маршрутов, учетных карточек, приписных карт, мобилизационных предписаний, извещений, повесток, алфавитных карточек, именных списков. Все вышеперечисленные компоненты системы дают представление об объеме документов, подлежащих обработке и, соответственно, доле труда. Большая часть такого труда выполняется сотрудниками военкомата без применения средств автоматизации. Организационная структура объекта автоматизации представлена военным комиссаром, отделом учета военнообязанных, ППСВ, участками оповещения, маршрутами.

Фрагмент описания функционирования объекта приведен ниже. Военнообязанный (ВО) становится на учет. Для него заполняются Учетная карточка (УК), Алфавитная карточка (АК), извещение. По параметрам трех разрядов учета (Первый: солдаты и сержанты от постановки на учет до 35 лет, мужчины-прапорщики до 45 лет; Второй: от 35 лет + 1 день до 45 лет, мужчины-прапорщики до 50 лет; Третий: от 45 лет + 1 день до 50 лет, мужчины-прапорщики до 55 лет, женщины до 45 лет (солдаты и сержанты), женщины-прапорщики до 50 лет). Для уточнения разрядов автоматизированная система содержит компонент «Разбиение ВО по разрядам» с фиксацией трех разрядов и свойств-фильтров по параметрам (пол, точный возраст (интервально), звания (перечисление), должности (перечисление), особые отметки).

Данный фрагмент технического задания, в силу ограниченности объема статьи, дает лишь начальное представление о содержании системы управления и части решаемых задач. Подробное описание предметной области выполнено в тексте технического задания.

Программная система управления «Учет военнообязанных» предназначена для автоматизации решения следующих задач:

1. Ввод данных с использованием клавиатуры, отображение на дисплее сведений о военнообязанных в виде форм стандартного учета, сохранение данных на носителе, чтение данных с носителя, реализация стандартных запросов к базам данных с целью получения статистических сведений о военнообязанных, записанных средствами программной системы учета военнообязанных.

2. Обращение к данным средствами, предоставляемыми технологиями клиент-сервер.

3. Защита хранимых данных средствами криптографии, которая осуществляется на уровне авторизации пользователей, хранения данных, защиты управляющего кода системы управления базой данных.

4. Разграничение прав доступа на уровни доступа (описание уровней приведено в техническом задании).

Условиями функционирования программной системы учета военнообязанных является сетевая вычислительная система, состоящая из сервера, работающего под управлением операционной системы (Microsoft либо Linux) с установленным программным обеспечением (сервер Apache, MySQL, PHP), протоколами Ipv4, Ipv6, IPX, TCP и стандартных браузеров-клиентов [1–3].

Персонал, сопровождающий ПС ВО на сервере и клиенте должен владеть навыками установки и настройки сетевого программного обеспечения для сервера Apache, MySQL, PHP, протоколов Ipv4, Ipv6, IPX, TCP.

К пользователям программной системы учета военнообязанных предъявляются следующие требования: владение навыками ведения диалога средствами операционной системы; наличие навыков внесения и редактирования записей базы данных с использованием экранных форм; умение использовать средства СУБД для формулирования запросов; навыки работы с браузером.

Описание предметной области, фрагмент которого приведен выше, позволило представить для нее информационно-логическую модель базы данных.

Примерный перечень сущностей, определенный на начальном этапе разработки, следующий: Военнообязанный, Команда, ППСВ, военно-учетные специальности, Должности, Участок, Маршрут.

На основании проектирования сущностей и связей выполнено информационно-логическое моделирование предметной области [4].

На основании проектирования сущностей и связей выполнено информационно-логическое моделирование предметной области (рисунок 1) и моделирование системы управления данными.

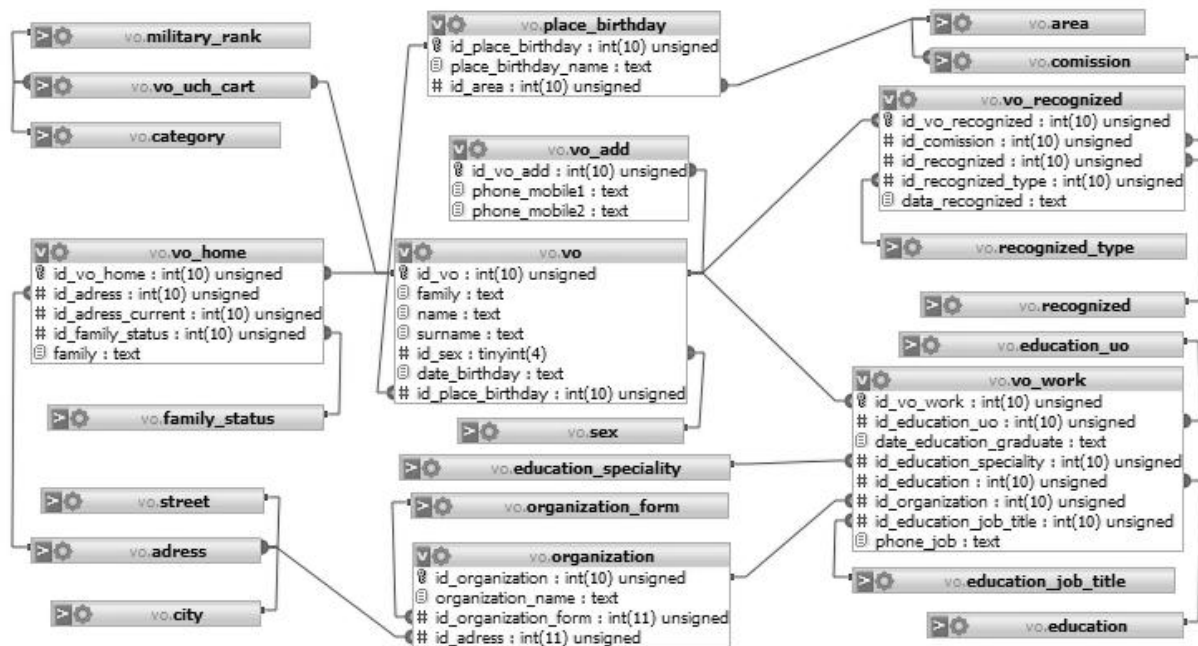


Рисунок 1 – Фрагмент инфологической модели предметной области «Учет военнообязанных»

Инфологическое моделирование выполнено средствами phpMyAdmin. Выбор программного обеспечения объясняется полной совместимостью с используемой базой данных MySQL и условиями распространения. Такая совместимость позволила легко перейти от инфологической модели базы данных к даталогической.

Модель системы управления, выполненная средствами phpMyAdmin, представлена на рисунке 2.

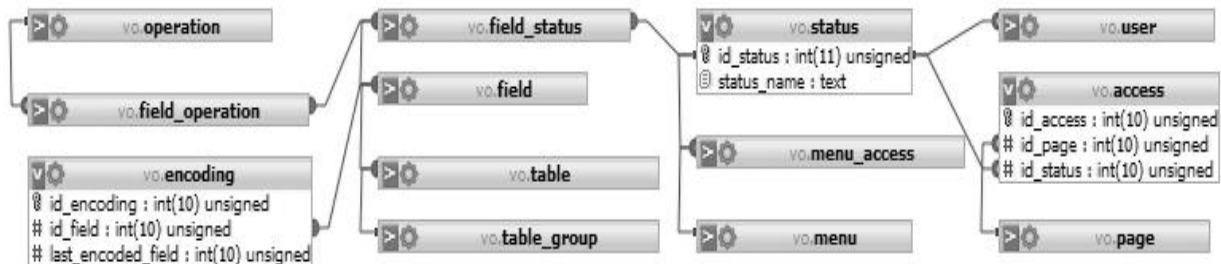


Рисунок 2 – Модель системы управления

Приведем сокращенный перечень основных и сопутствующих задач для автоматизации на основе базы данных, а также некоторые фрагменты их решения. Основными задачами являлись: формирование списочного состава команд с разбиением по составам и литерам, формирование ППСВ, формирование участков на основе адресных данных о ВО, формирование маршрутов на основе адресных данных об участках и адресов ВО, формирование именных списков. Сопутствующими задачами определены: поиск продублированных данных о ВО в форме визуального контроля по совпадающим фамилии, имени, отчеству с выводом для проверки дополнительных сведений (дата рождения, адрес места жительства, другие сведения по выбору), масштабирование данных для вывода в поле формы ограниченного размера. Число экземпляров форм, требующих обработки, пропорционально мощности базы данных. Число печатных форм на каждого военнообязанного более десяти. Предварительная экспертная оценка показала, что мощность базы данных составляет около 100 тысяч записей.

Входная информация для системы представлена стандартными печатными формами. Фрагмент одной из форм, реализованных в системе, приведен на рисунке 3.

Год рождения	<u>1980</u>				Степень годности
(18.) № ВУС	Код должности	УЧЕТНАЯ КАРТОЧКА			Г
<u>100</u>	<u>915</u>	к военному билету серии № <u>НБ 1047628</u>			
(фамилия, имя, отчество)		(10.) Военное звание	<u>рядовой</u>	ПРЕДНАЗНАЧЕН:	
<u>Иванов</u>				№ команды и литер по ВУС	<u>Р00000А</u>
<u>Иван</u>		(16.) Категория учета	<u>1</u>		<u>100</u>
<u>Иванович</u>					
Число, месяц, год рождения		(17.) Группа учета	<u>ОПС</u>	на воинскую должность	<u>стрелок</u>
<u>14</u>	<u>07</u>	<u>1980</u>	(18.) Состав	<u>солдаты</u>	
(1.) Место рождения				В/звание по штату	<u>рядовой</u>
<u>г. Брест</u>					
<u>Брестская обл.</u>		(19.) Наимен. воинск. должности и ВУС (долж. квалифик.)		№ маршрута	<u>10/1284 10/2154</u>

Рисунок 3 – Фрагмент учетной карточки

Сложность автоматизированного заполнения такого рода форм (фиксированный размер и сложная разметка) состоит в невозможности априорно определить занимаемое место для вывода данных. Использование мелких шрифтов на электронной форме затрудняет чтение информации и эстетическое представление. Решение проблемы было построено с использованием Java Script Фреймворка jQuery. С его помощью реализована возможность динамического масштабирования записей на электронной форме на стороне клиента (для наглядности на рисунке 3 масштабируемые данные подчеркнуты). Фрагмент скрипт-кода для масштабирования приведен ниже.

```
Фрагмент 1 – jQuery код для масштабирования
// jQuery методы для вписывания данных в формы
$(document).ready(function () {
    var minFontSize = 8;
    var minLineHeight = 7;

    // Уменьшение размеров текста чтобы не выходило за внешний
// контейнер
    $('.data').each(
        function() {
            var cont = $(this).parent().parent();
            if (cont.hasClass('dcont'))
                cont = cont.parent();
            var width = cont.width();
            var height = cont.height();
            // Уменьшаем по ширине
            while (($(this).width() > width - 2) || ($(this).height() > height + 1)) {
                var newFontSize = $(this).css('font-size');
                newFontSize = newFontSize.replace("px", "") - 1;
                $(this).css('font-size', newFontSize + 'px');
                if (newFontSize <= minFontSize) {
                    $(this).css('width', width + 'px');
                    break;
                }
            }

            // Уменьшаем отступы между строками
            if (height > 0)
                while ($(this).height() > height) {
                    var newLineHeight = $(this).css('line-height');
                    newLineHeight = newLineHeight.replace("px", "") - 1;
                    $(this).css('line-height', newLineHeight + 'px');
                    if (newLineHeight <= minLineHeight) {
                        $(this).css('height', height + 'px');
                        break;
                    }
                }
            $(this).css('position', 'static');
        });
});
```

В ПС ВО присутствуют несколько десятков взаимосвязанных справочных таблиц. Проектирование интерфейса их взаимодействия представляло интерес при разработке системы управления учетом военнообязанных. Для обработки справочников реализована удобная и эффективная система, которая предоставляет навигацию, поиск и сортировку по записям и полям данных.

Внешний вид диалоговой среды системы предоставлен на рисунках 4, 5.

Справочники | Учёт военнообязанных

Данные ВО   Справочники   Выходные документы   Настройки   Права доступа   Выход

Таблицы street (Улицы) [699]

Страницы: ← 1 2 3 4 ... 9 10 11 12 13 14 15 ... 32 33 34 35 →

address (Адрес)	id_street	street_name		
area (Области)	221	ЖАСМИНОВАЯ		
category (Категории)	222	ЖЕЛЕЗНОДОРОЖНАЯ		
city (Города)	223	ЖЕЛЕЗНОДОРОЖНЫЙ, ПЕР		
comission (Комиссия)	224	ЖИТНИЙ, ПЕР		
education (Образование)	225	ЖИТНЯЯ		
	226	ЖИТОМИРСКАЯ		

Рисунок 4 – Диалог. Взаимодействие со справочниками

Права доступа | Учёт военнообязанных

Данные ВО   Справочники   Настройки   Права доступа   Выход

Параметры	Администратор	Военный комиссар	Секретарь	Гость
Главная	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Пользователи	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Справочники	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Настройки	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Права доступа	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Данные о военнообязанных	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Выходные документы	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Таблицы	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Импорт данных	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Сохранить

Copyright © 2009-2011. Валерий Охримчук

Рисунок 5 – Диалог. Редактирование доступа на странице в зависимости от статуса пользователя

В ходе проектирования широко используется работа со списками. Многие из списков имеют большой размер, что затрудняет поиск и существенно замедляет взаимодействие. Пример такого списка содержит форма для редактирования информации о военнообязанном (рисунок 6). Для эффективной работы со списками форм используется расширенный поиск и оптимальная загрузка справочников (фрагмент 2). Для решения указанных задач применены технологии:

1) AJAX (для динамического обращения к серверу). Позволяет в описываемой системе управления выполнять загрузку только необходимого фрагмента страницы при обращении к серверу, тогда как «стандартные средства» HTML производят полную перезагрузку при любом ее изменении.

2) JSON. Представляет собой усовершенствованную технологию объектного подхода. Дает возможность эффективно организовывать передачу данных между сервером и клиентом.

3) jQuery, Java Script, Фреймворк, позволяющий облегчить работу с элементами HTML страниц [6].

Общая информация | Учётная карточка | Предназначение | Прохождение военной службы | Другие данные | Воинский учёт | Особые отметки

**Иванов Иван Иванович 01.02.1960**

Общая информация

Фамилия	Иванов
Имя	Иван
Отчество	Иванович
Пол	Мужской
Дата рождения	01.02.1960
(1.) Место рождения	Брест
(2.) Образование	Брестское ПТУ 65
В каком году	1999
(3.) Основная гражданская специальность	ист
Место работы	программист
Должность	СИСТЕМОТЕХНИК
	юрист

Сохранить

**Рисунок 6 – Форма для редактирования информации о военнообязанном**

Фрагмент 2 – Программный код динамической загрузки справочников

// jQuery методы для работы со списками

\$(document).ready(function () {

// Загрузка содержимого в div

load = function (element) {

if (element.attr('value-last') != element.attr('value')) {

element.attr('value-last', element.attr('value'));

var listContainer = element.parent().parent();

var listEditContainer = listContainer.find('.listEditContainer');

var listEdit = listContainer.find('.listEdit');

var listDiv = listContainer.find('.listDiv');

// Загрузка выпадающего списка

\$.getJSON("inc/ajax\_json.php",

{

table : element.attr('table'),

field : element.attr('field'),

value : element.attr('value')

},

function(data) {

listDiv.html("");

// Добавление данных на страницу

\$.each(data, function(key, val) {

listDiv.append('<div class="listResult" id="' + element.attr('field') +

val.id + "'> + val.content + '</div>');

});

// Подсвечивание результатов при наведении мышки

```

listContainer.find('.listResult').hover(
    function() {
        listDiv.find('.listResult.select').removeClass('select');
        $(this).addClass('select');
    });
// Обработка нажатий
listContainer.find('.listResult').bind('click',
    function() {
        listEdit.attr('value', $(this).text());
        load(element);
        element.focus();
    });
});
}
};
});

```

Специфика сведений, представленных в базе данных системы, требует защиты информации от несанкционированного доступа. Для защиты используется класс Encoding, программный код которого приведен во фрагменте 3.

Фрагмент 3 – PHP код класса кодирования

```

<?php
// Кодирование
class Encoding
{
    // Закодированные поля
    private $field_code = array();

    // Загружена таблица или нет
    private $field_code_load = array();

    private $gamma = "";

    public $key;
    public $temp_table;

    // Функция шифрования
    function encode($input, $code = 1) {
        if ($code)
            return base64_encode($this->xorShipher($input, $this->key));
        return $this->xorShipher(base64_decode($input), $this->key);
    }
    // Алгоритм шифрования
    function xorShipher($string, $password) {
        $str_len = strlen($string);
        while (strlen($this->gamma) < $str_len)
        {
            $seq = pack("H*", sha1($this->gamma . $password));
            $this->gamma .= substr($seq, 0, 8);
        }
    }
}

```



```
}
return $string ^ $this->gamma;
}

// Кодирование записи $record из таблицы $table, если $code = 0 -
раскодирование
// Метод кодирует необходимые значения ассоциируемого массива
function encode_record($table, $record, $code)
{
    global $mysql;
    // загружаем если не загружена
    if (!isset($this->field_code_load[$table])) {
        $this->field_code_load[$table] = 1;
        $id_table = $this->temp_table->load_id_table($table);
        $field_arr = $mysql->get_array_from_query("SELECT * FROM `field`
WHERE `id_table` = {$id_table} AND `field_code` = 1");
        if (null != $field_arr)
            foreach ($field_arr as $field)
                $this->field_code[$table][ ] = $field['field_name'];
    }
    // пробегаем по всем полям и перекодируем необходимые
    if (isset($this->field_code[$table]))
        foreach ($this->field_code[$table] as $field)
            $record[$field] = $this->encode($record[$field], $code);
    return $record;
}
}
```

В приведенном коде используется метод шифрования данных xorShipher, основанный на алгоритме гаммирования [5].

Программная система «Учет военнообязанных» предназначена для эксплуатации в сетевой системе, состоящей из сервера под управлением ОС Linux или Windows и клиентов с установленным Браузером IE, Opera, Firefox. Обслуживание ПС ВО обеспечивается персоналом, имеющим навыки работы на сетевом ПО под управлением ОС Linux (Windows), Квалификация обслуживающего персонала требует выполнения операций по установке и конфигурированию сетевого ПО. Выполнение основных операций редактирования базы данных обеспечивается персоналом, имеющим навыки работы с оконными приложениями на уровне оператора ЭВМ.

Приведенное описание этапов создания системы «Учет военнообязанных» дает представление об использованных математических моделях, технологиях и средствах, которые повышают эффективность работы программистов. В настоящее время готовится внедрение и испытание системы «Учет военнообязанных».

#### СПИСОК ЛИТЕРАТУРЫ

1. HTTP-сервер Apache [Электронный ресурс] – Режим доступа: <http://apache.com/>. – Дата доступа: 20.08.2011.
2. Язык серверных скриптов PHP [Электронный ресурс] – Режим доступа: <http://php.net/>. – Дата доступа: 21.08.2011.

3. Свободная система управления базами данных MySQL [Электронный ресурс] – Режим доступа: <http://mysql.com/>. – Дата доступа: 21.08.2011.
4. Дейт, К. Дж. Введение в системы базы данных. К. Дж. Дейт. – 7-е изд., пер. с англ. – М. : Издательский дом «Вильямс», 2001. – 1072 с.
5. Ховард, М. Защищённый код / М. Ховард, Д. Лебланк. – 2-е изд., испр., пер. с англ. – М. : Издательский дом «Русская Редакция», 2004. – 704 с.
6. Java Script jQuery [Электронный ресурс] – Режим доступа: <http://jquery.com/>. – Дата доступа: 22.08.2011.

***A.A. Kazinski, V.V. Akhrymchuk. Realization of Mathematical Model of a Software System of Management of the Account of Persons Liable for Military Service***

The topic of the present work is devoted to creation of models of automated software management system of the account of persons liable for military service. In the work of the sanctified by the decision of the following tasks: creation of инфологической database model, the scaling of the text fields in the printed form, construction of the interface elements, directories, search systems, and display information network user interface. Considerable attention is given to the elements of technical tasks, software choice of models of management systems. The object of automation system «the Account of persons liable for military service» is a complex task of setting the persons liable for military service on account of, the creation of the notice of registration for delivery at the place of the former accounting, formation of groups and others. The article describes the fragments of program code, implementing the decision of the set tasks, screenshots dialog pages.

Рукапіс паступіў у рэдкалегію 04.10.2011 г.