**Министерство образования Республики Беларусь**

**Учреждение образования**

**«Белорусский государственный университет информатики и радиоэлектроники»**

Факультет непрерывного и дистанционного обучения

Специальность «ИиТП»

**ИНДИВИДУАЛЬНЫЕ ПРАКТИЧЕСКИЕ РАБОТЫ №№ 1, 2**

по предмету «Технологии обработки финансовой информации»

по теме «Кредитный отдел банка»

Студент дистанционной формы

обучения 5 курса

Группы № 393551

Гер Сергей Эдуардович

г.Минск, ул. Водолажского, 6А/41

Тел. +375-29-1846573

Минск 2017

Содержание

[1 Спецификация требований 4](#_Toc497819046)

[1.1 Неавторизованный пользователь 4](#_Toc497819047)

[1.2 Клиент (пользователь) 4](#_Toc497819048)

[1.3 Специалист по работе с клиентами 5](#_Toc497819049)

[1.4 Сотрудник службы безопасности 5](#_Toc497819050)

[1.5 Член кредитного комитета 5](#_Toc497819051)

[1.6 Начальник кредитного отдела 5](#_Toc497819052)

[1.7 Операционист 5](#_Toc497819053)

[1.8 Администратор 6](#_Toc497819054)

[1.9 Веб-приложение (псевдороль) 6](#_Toc497819055)

[2 План разработки 7](#_Toc497819056)

[3 Реализация приложения 8](#_Toc497819057)

[3.1 Область доступная всем 8](#_Toc497819058)

[3.2 Личный кабинет. Заявки на получение кредита 9](#_Toc497819059)

[3.3 Личный кабинет. Кредиты 14](#_Toc497819060)

[3.4 Возможности администратора 16](#_Toc497819061)

[4 База данных 20](#_Toc497819062)

[5 Программный код (выборочно) 22](#_Toc497819063)

[5.1 Скрипт начисления пени клиентам 22](#_Toc497819064)

[5.2 Регистрация пользователей 23](#_Toc497819065)

[5.3 Шаблон страницы регистрации пользователей 24](#_Toc497819066)

[5.4 Управление кредита в админ-панели 25](#_Toc497819067)

[5.5 Модель заявок на кредиты 27](#_Toc497819068)

[6 Приложение 1. Архив с приложением 30](#_Toc497819069)

[6.1 Тестовые данные 30](#_Toc497819070)

[6.2 Работа скрипта начисления пени в тестовом режиме 30](#_Toc497819071)

# Спецификация требований

Требуется разработать веб-приложение для банка для работы с кредитами физических лиц с использованием архитектуры клиент-сервер.

В приложении можно выделить следующие роли:

## Неавторизованный пользователь

Неавторизованный пользователь – это потенциальный кредитуемый или сотрудник банка, который еще не прошел процесс авторизации. Ему должны быть доступные следующие возможности:

1. Получить информацию о кредитной ставке банка на данный момент;
2. Воспользоваться так называемым «кредитным калькулятором» - формой, состоящей из полей суммы кредита и срока кредита. После отправки формы на сервер, система должна вывести пользователю на экран данные о графике погашения задолженности и суммах, которые нужно будет выплатить.
3. Зарегистрироваться в системе;
4. Авторизоваться.

Возможности 1 и 2 должны быть доступны любому другому пользователю.

## Клиент (пользователь)

После процесса авторизации клиенту должны быть доступные следующие возможности:

1. Заявки на кредит:

* Создать заявку на кредит и редактировать ее в режиме «Черновик»:
  + Указать в ней необходимые поля, такие как ФИО, паспортные данные и др.
  + Загрузить отсканированные изображения необходимых для получения кредита документов: паспорта, военного билета, и др.
* Отправить заявку на кредит на рассмотрение в банк.
* Ознакомиться с результатом рассмотрения заявки банком, в том числе видеть причину отказа в поле «Комментарии».
* Обозревать список своих заявок.

1. Кредиты:
   * Обозревать список своих кредитов;
   * Видеть план оплат по кредиту: дату и сумму оплаты;
   * Видеть начисленную системой пеню;
   * Видеть список собственных оплат.
   * Видеть выплачен кредит или нет.

## Специалист по работе с клиентами

Проверяет корректность заполнения заявки на получение кредита. Может подтвердить корректность или отклонить заявку, указав причину в поле «Комментарий».

## Сотрудник службы безопасности

По своей базе данных (не связанной с разрабатываемым нами ПО) проверяет потенциального клиента и выносит решение о безопасности сделки с ним. Может одобрить заявку на получение кредита или отклонить ее.

## Член кредитного комитета

Может одобрить заявку на получение кредита или отклонить ее.

## Начальник кредитного отдела

Выносит финальное решение о выдаче кредита или отклонении заявки на него.

## Операционист

Выдает деньги клиенту, фиксирует оплату клиентом. Система должна предоставить операционисту слеюущие возможности:

1. Видеть список одобренных заявок на получение кредита, по которым клиентам не были выданы деньги;

2. Отмечать факт выдачи денег клиенту в системе;

3. Видеть планируемые выплаты клиентов и их фактические выплаты.

4. Отмечать в системе факт оплаты части кредита клиентом.

## Администратор

Администратор системы должен иметь возможность менять роль пользователям, а также создавать, редактировать и удалять их.

## Веб-приложение (псевдороль)

Веб-приложение должно:

1. Начислять клиентам пеню;

2. Разделять платежи клиента на оплату пени и оплату кредита;

3. Помечать кредит как полностью выплаченный после последней выплаты по кредиту;

4. Разграничивать доступ пользователям с различными ролями.

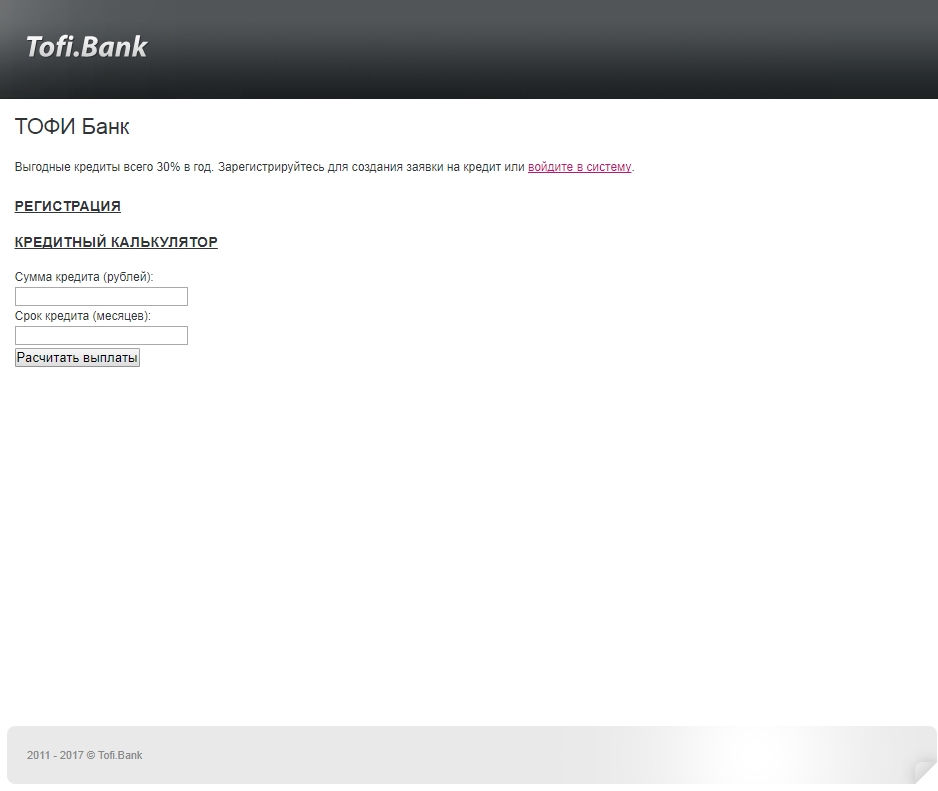
# План разработки

|  |  |  |
| --- | --- | --- |
| **№** | **Задача с кратким описанием** | **Срок разработ-ки (часы)** |
|  | Установка и настройка окружения: веб-сервер, рабочая среда | 8 |
|  | Разработка админ-панели и возможности авторизации | 16 |
|  | Создание механизма ролей | 8 |
|  | Создание тестовых пользователей в системе под каждую роль | 3 |
|  | Создание механизма CRUD пользователей для администратора системы | 7 |
|  | Создание раздела «Заявки на кредит» в панели администрирования с разделением управления заявками в зависимости от роли пользователя в приложении | 20 |
|  | Создание раздела «Кредиты» в панели администрирования с разделением управления кредитами в зависимости от роли пользователя в приложении | 20 |
|  | Создание кредитного калькулятора | 16 |
|  | Создание механизма начисления пени и вынос начисления в скрипт для crontab-задач | 8 |
|  | Тестирование и отшлифовка мелочей | 24 |
|  | Выгрузка в продакшн | 8 |
|  | **Итого** | 138 |

# Реализация приложения

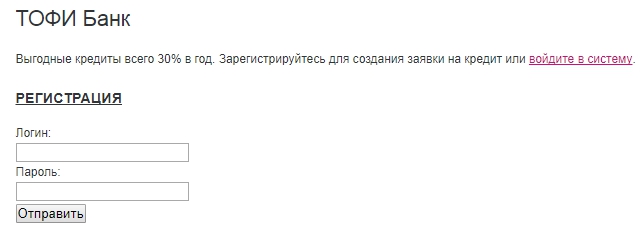
## Область доступная всем

Область доступная всем в приложении выглядит так:

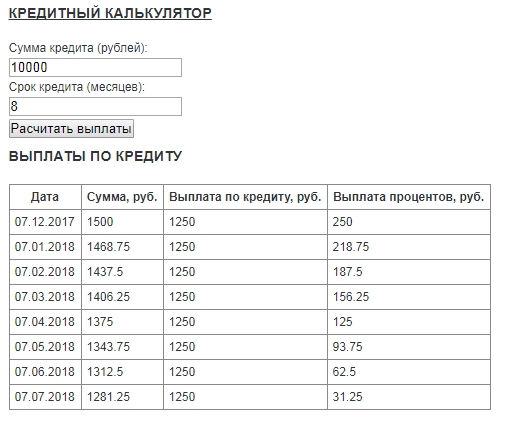


Пользователь может ознакомиться с кредитной ставкой, а также перейти к авторизации в личный кабинет.

Пользователям приложения также доступна и регистрация:

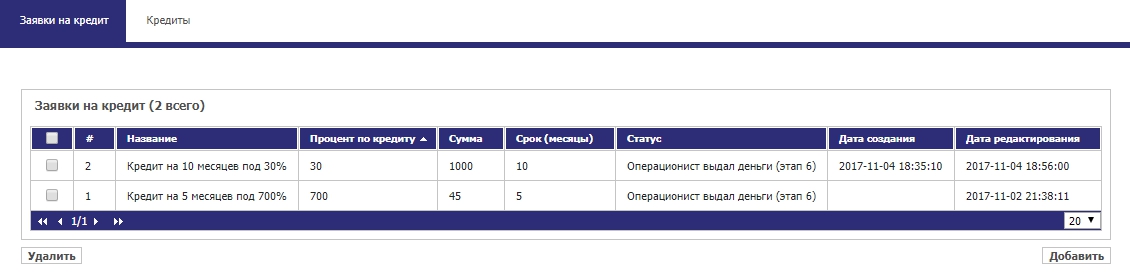


Ниже находится кредитный калькулятор:

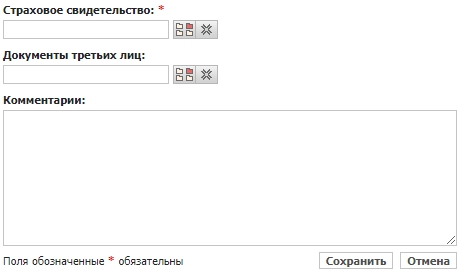
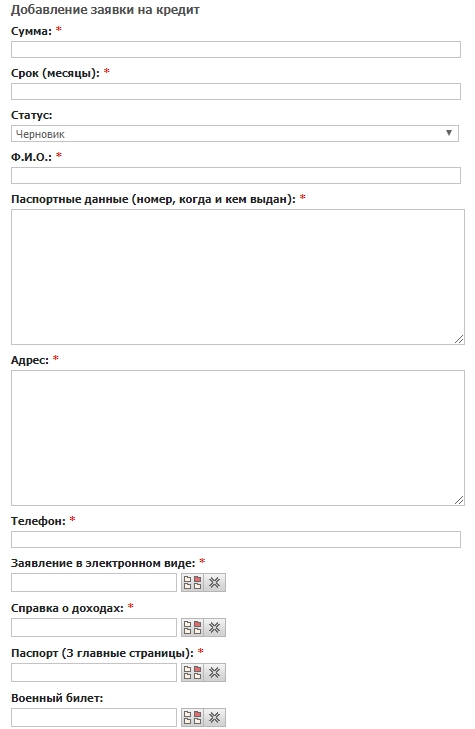


## Личный кабинет. Заявки на получение кредита

После авторизации пользователь попадает на список заявок на получение кредита:



Где может управлять своими заявками на получение кредита. В форме заявки – все необходимые поля:



После отправки сообщения (и смены статуса на «отправлена» заявка становится доступной:

1. Специалисту по работе с клиентами. В случае его одобрения она переходит

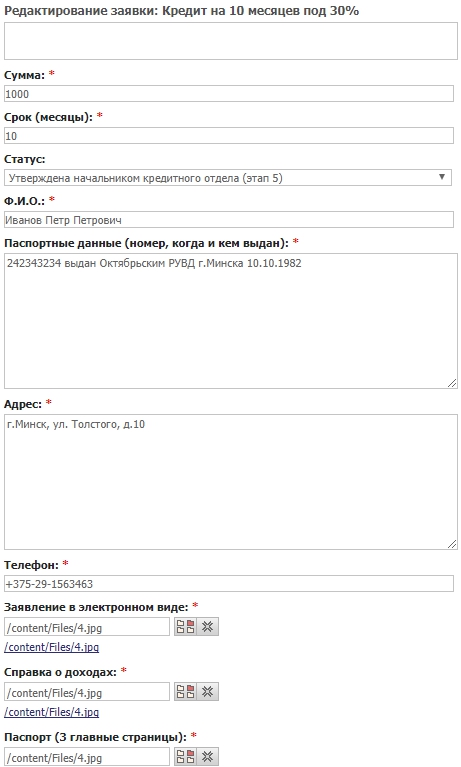
2. Сотруднику службы безопасности. В случае его одобрения она переходит

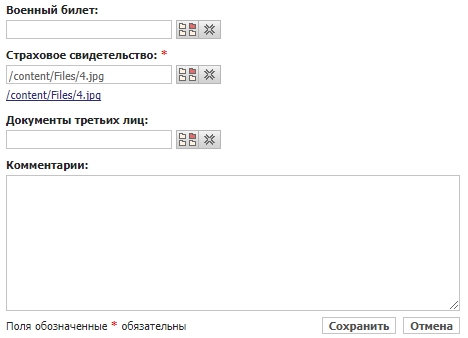
3. Члену кредитного комитета. В случае его одобрения она переходит

4. Начальнику кредитного отдела. В случае его одобрения она переходит

5. Операционисту. Он выдает кредит и отмечает в системе, что кредит выдан.

Интерфейс у всех проверяющих заявку схож:



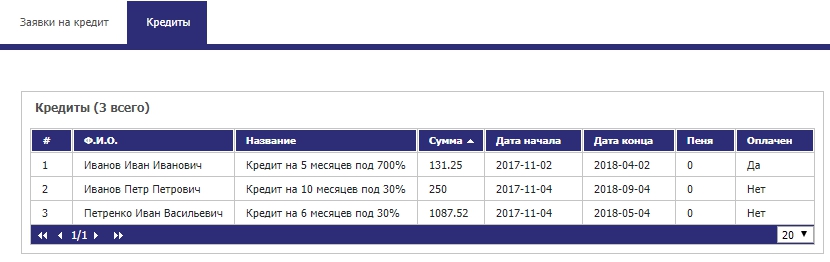


Каждый сотрудник банка одобряет или отклоняет заявку, о причине отклонения записывает в поле «Комментарии».

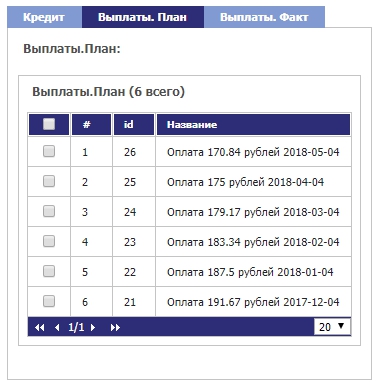
Операционист на последнем шаге отмечает, что кредит выдан и на базе данных заявки создается запись в разделе «Кредиты».

## Личный кабинет. Кредиты

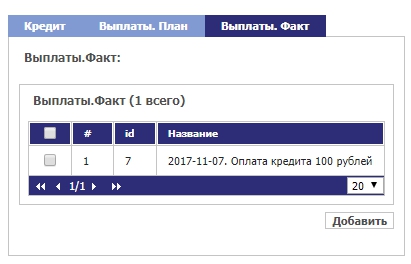
У операциониста в личном кабинете доступны кредиты всех клиентов. У клиента доступны только его собственные кредиты.



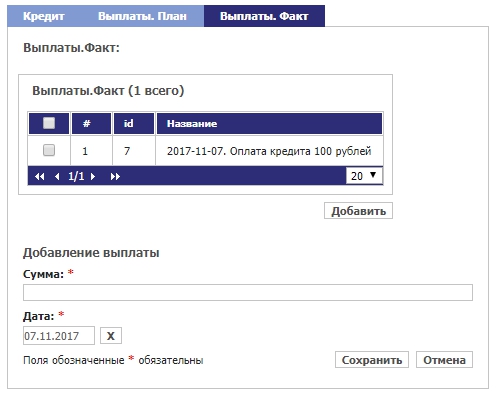
После создания кредита в базу данных записывается план выплат, который доступен операторам и клиентам:



Фактические выплаты клиента фиксируются оператором в подразделе карточки кредита Выплаты.Факт:

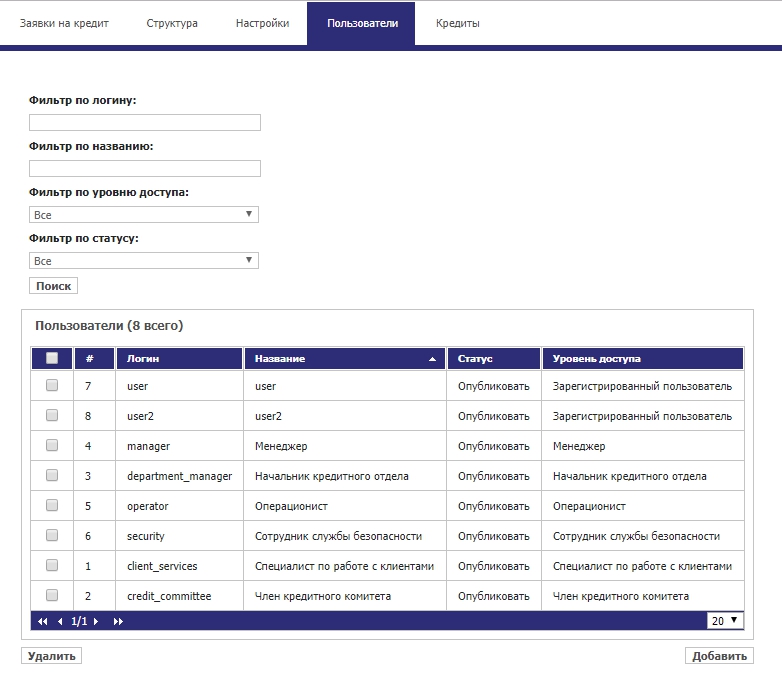


Клиент может видеть список своих выплат, а операционист и видеть и добавлять новые:

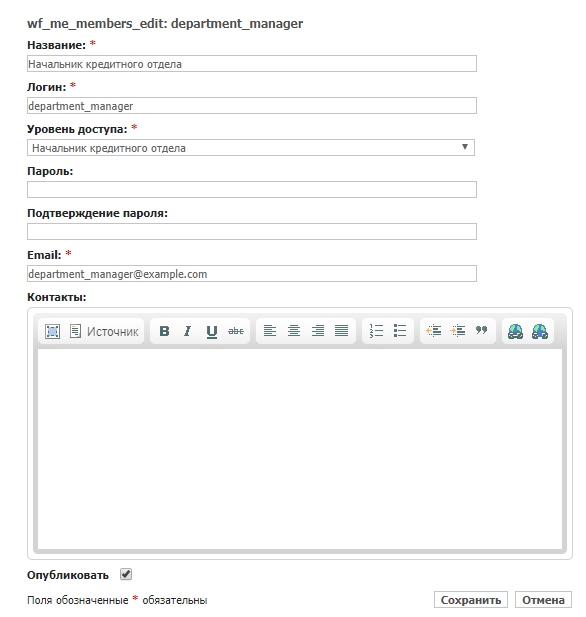


## Возможности администратора

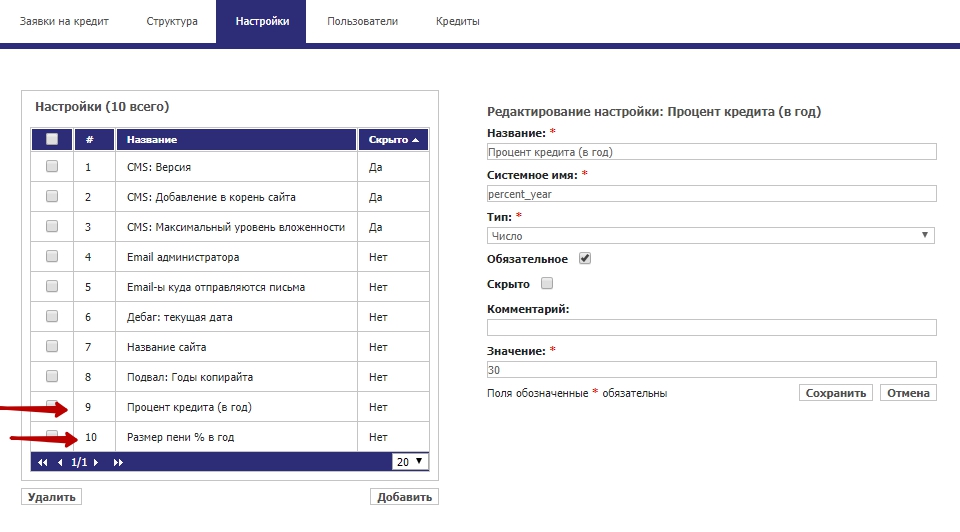
Администратору доступен раздел с пользователями:



С возможностью их создавать, редактировать, удалять, менять роли:

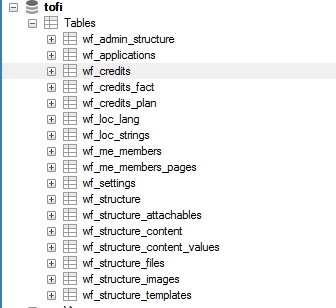


Администратору также доступен раздел с настройками, где можно указать размер процента по кредитам, а также размер пени.



# База данных

Все таблицы базы данных (в том числе и вспомогательные):



Структура таблиц, связанных с бизнес-логикой приложения:

CREATE TABLE `wf\_credits` (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`title` varchar(255) DEFAULT NULL,

`amount` float DEFAULT NULL,

`date\_start` date DEFAULT NULL,

`date\_end` date DEFAULT NULL,

`application\_id` int(11) DEFAULT NULL,

`fio` varchar(255) DEFAULT NULL,

`user\_id` int(11) DEFAULT NULL,

`fine` float NOT NULL DEFAULT '0',

`paid` int(1) NOT NULL DEFAULT '0',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8

CREATE TABLE `wf\_credits\_fact` (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`title` varchar(255) DEFAULT NULL,

`amount` float DEFAULT NULL,

`date` date DEFAULT NULL,

`credit\_id` int(11) DEFAULT NULL,

`is\_fine` int(1) NOT NULL DEFAULT '0',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8

CREATE TABLE `wf\_credits\_plan` (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`title` varchar(255) DEFAULT NULL,

`amount` float DEFAULT NULL,

`date` date DEFAULT NULL,

`credit\_id` int(11) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=27 DEFAULT CHARSET=utf8

CREATE TABLE `wf\_me\_members` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`id\_level` int(11) NOT NULL,

`password` varchar(64) NOT NULL,

`restored\_password` varchar(64) DEFAULT NULL,

`login` varchar(64) NOT NULL,

`title` varchar(256) NOT NULL,

`email` varchar(256) NOT NULL,

`uri` text,

`status` int(11) NOT NULL DEFAULT '0',

`contacts` text,

PRIMARY KEY (`id`,`id\_level`,`status`),

UNIQUE KEY `id` (`id`,`status`),

UNIQUE KEY `id\_2` (`id`),

KEY `status` (`status`)

) ENGINE=InnoDB AUTO\_INCREMENT=40 DEFAULT CHARSET=utf8

Все остальные таблицы, а также данные находятся в приложении (файл-дамп базы tofi.sql).

# Программный код (выборочно)

Ниже приведены фрагменты программного кода. Весь код можно найти в приложении.

## Скрипт начисления пени клиентам

**<?php  
  
require\_once** (**'../Classes/initializer.php'**);  
  
*define*(**'DEBUG\_MODE\_PROCESS\_DAY'**, 1);  
  
$currentDay = ***DEBUG\_MODE\_PROCESS\_DAY*** ? Core::*getInstance*()->**Settings**->getSetting(**'debug\_current\_date'**) : *date*(**'Y-m-d'**);  
  
$finePercentDay = Core::*getInstance*()->**Settings**->getSetting(**'fine'**) / 100 / 365;  
  
$db = Core::*getInstance*()->**DataBase**;  
  
$query = **"  
 SELECT c.\*,  
 (  
 ifnull((select sum(cp.amount) from wf\_credits\_plan cp where cp.credit\_id = c.id AND cp.date < '**$currentDay**'),0) + c.fine  
 -  
 ifnull((select sum(cf.amount) from wf\_credits\_fact cf where cf.credit\_id = c.id AND cf.is\_fine <> 1),0)  
 ) dept  
 FROM wf\_credits c  
 WHERE c.paid = 0 AND c.date\_start <= '**$currentDay**'  
 having dept > 0  
"**;  
  
$res = $db->selectCustomSql($query);  
  
**foreach** ($res **as** $credit) {  
 $credit[**'fine'**] += *round*($finePercentDay \* $credit[**'dept'**], 2);  
 Core::*getInstance*()->getModel(**'Credits'**)->updateItem($credit[**'id'**], **array**(**'fine'** => $credit[**'fine'**]));  
}  
  
  
**if** (***DEBUG\_MODE\_PROCESS\_DAY***) {  
 $nextDay = *date*(**'Y-m-d'**, *strtotime*(**'+1 day'**, *strtotime*($currentDay)));  
 $db->updateSql(**'wf\_settings'**, **array**(**'value'** => $nextDay), **array**(**'system'** => **'debug\_current\_date'**));  
}  
  
**echo 'Current Day: '** . $currentDay;

## Регистрация пользователей

**<?php  
  
class** RegistrationPage **extends** FrontPage  
{  
 **protected function** initialize()  
 {  
 **parent**::*initialize*();  
  
 $this->**tplVars**[**'percent'**] = $this->**Settings**->getSetting(**'percent\_year'**);  
  
 Form::*addForm*(**'registration\_form'**);  
 Form::*addField*(**'registration\_form'**, **new** FormFieldRequired(**'registration\_form\_name'**, **array**(**'minLength'** => 4, **'maxLength'** => 64)));  
 Form::*addField*(**'registration\_form'**, **new** FormFieldRequired(**'registration\_form\_password'**, **array**(**'minLength'** => 4, **'maxLength'** => 64), **'password'**));  
 Form::*addField*(**'registration\_form'**, **new** FormField(**'registration\_form\_store'**, **array**(**'default'** => 0), **'hidden'**));  
 **if** (Form::*isSubmited*(**'registration\_form'**))  
 {  
 **if** (Form::*validate*(**'registration\_form'**))  
 {  
 $this->**Core**->**currentLang** = InPost(**'admin\_area\_id\_lang'**, 1);  
 SetCacheVar(**'admin\_area\_id\_lang'**, $this->**Core**->**currentLang**);  
  
 $l = $this->**tplVars**[**'registration\_form\_name'**];  
 $p = $this->**tplVars**[**'registration\_form\_password'**];  
  
 Loader::*loadBlock*(**'ArrayOutput'**, **'Helper'**);  
 BlocksRegistry::*getInstance*()->registerBlock(**'ArrayOutput'**, **new** ArrayOutput());  
  
 **if** ($this->**User**->getCount(**array**(**'login'** => $l))) {  
 $this->**tplVars**[**'error'**][] = **'Пользователь с таким логином уже есть в системе. Выберите другой логин.'**;  
 } **else** {  
 **if** ($this->**User**->addItem(**array**(  
 **'login'** => $l,  
 **'title'** => $l,  
 **'email'** => $l . **'@example.com'**,  
 **'password'** => $p,  
 **'cpassword'** => $p,  
 **'status'** => 1,  
 ))) {  
*// $this->tplVars['info'][] = 'Вы успешно зарегистрировались в системе.';* **if** ($this->**User**->login($l, $p, 0)) {  
 Core::*redirect*($this->**tplVars**[**'HTTP'**].**'admin/'**);  
 **return**;  
 }  
 }  
  
 $this->**tplVars**[**'error'**][] = **'Произошла неизвестная ошибка при регистрации.'**;  
  
 }  
 }  
 **else** {  
 Loader::*loadBlock*(**'ArrayOutput'**, **'Helper'**);  
 BlocksRegistry::*gШаблоetInstance*()->registerBlock(**'ArrayOutput'**, **new** ArrayOutput());  
  
 $this->**tplVars**[**'registration\_form\_errors'**] = Form::*getFormErrors*(**'registration\_form'**);  
 }  
 }  
  
 $this->**tplVars**[**'calculator\_show'**] = **false**;  
 **if** (**isset**($\_GET[**'credit\_amount'**])) {  
 $this->**tplVars**[**'calculator\_show'**] = **true**;  
 $percent = Core::*getInstance*()->**Settings**->getSetting(**'percent\_year'**);  
 $credits = Core::*getInstance*()->getModel(**'Credits'**);  
 */\*\*****@var*** *Credits $credits \*/* $this->**tplVars**[**'creditInfo'**] = $credits->getCreditInfo(  
 $\_GET[**'credit\_term'**],  
 $\_GET[**'credit\_amount'**],  
 $percent  
 );  
 }  
 }  
}

## Шаблон страницы регистрации пользователей

<**div** class=**"wrap"**>  
 <**h2**>ТОФИ Банк</**h2**>  
 <**div**>Выгодные кредиты всего <?**=$percent;**?>% в год. Зарегистрируйтесь для создания заявки на кредит или <**a** href=**"/admin/"**>войдите в систему</**a**>.</**div**>  
 <**br** />  
 <**h3** style=**"**text-decoration: **underline**; cursor: **pointer**;**"** onclick=**"**$(**'.register-block'**).toggle();**"**>Регистрация</**h3**>  
  
 <**style**>  
 **p**.**err** {  
 color: **red**;  
 margin: 3**px** 0 3**px** 0;  
 padding-bottom: 0**px !important**;  
 }  
 .**register-block label**, .**calculator-block label** {  
 width: 200**px**;  
 display: **block**;  
 margin-bottom: 3**px**;  
 margin-top: 3**px**;  
 }  
  
 .**register-block label**.**error**, .**calculator-block label**.**error** {  
 width: **auto**;  
 color: **red**;  
 display: **inline**;  
 margin-left: 5**px**;  
 }  
  
 .**register-block button**, .**calculator-block button** {  
 margin-top: 3**px**;  
 }  
 </**style**>  
 <**div** class=**"register-block"** style=**"**display:**none**;**"**>  
 <?**=get\_block('FormGenerator', array(  
 'name' =**> 'registration\_form',  
 'action' => '/',  
 'submit\_title' => 'button\_register',  
 'form\_type' => 'RegisterForm',  
 ))?>  
 </**div**>  
  
 <**h3** style=**"**text-decoration: **underline**; cursor: **pointer**;**"**>Кредитный калькулятор</**h3**>  
 <**div** class=**"calculator-block"**>  
 <**form** name=**"credit\_form"** id=**"credit\_form"** method=**"get"** action=**"/"**>  
 <**div**>  
 <**div**>  
 <**label** for=**"credit\_amount"**>Сумма кредита (рублей):</**label**>  
 <**input** autocomplete=**"off"** type=**"text"** value=**"<?=(isset($\_GET['credit\_amount']) ? $\_GET['credit\_amount'] : '')?>"** name=**"credit\_amount"** id=**"credit\_amount"** class=**"i-text"** />  
 </**div**>  
 <**div**>  
 <**label** for=**"credit\_term"**>Срок кредита (месяцев):</**label**>  
 <**input** autocomplete=**"off"** type=**"text"** value=**"<?=(isset($\_GET['credit\_term']) ? $\_GET['credit\_term'] : '')?>"** name=**"credit\_term"** id=**"credit\_term"** class=**"i-text"** />  
 </**div**>  
 <**button** type=**"submit"** onclick=**""**><**span**>Расчитать выплаты</**span**></**button**>  
 <**div** class=**"clr"**></**div**>  
 </**div**>  
 <?**php if ($calculator\_show) :** ?>  
 <**div**>  
 <**h3**>Выплаты по кредиту</**h3**>  
 <**table**>  
 <**tr**>  
 <**th**>Дата</**th**>  
 <**th**>Сумма, руб.</**th**>  
 <**th**>Выплата по кредиту, руб.</**th**>  
 <**th**>Выплата процентов, руб.</**th**>  
 </**tr**>  
 <?**php foreach ($creditInfo['plans'] as $payment) :**?>  
 <**tr**>  
 <**td**><?**=date('d.m.Y', strtotime($payment['date']))**?></**td**>  
 <**td**><?**=$payment['amount']**?></**td**>  
 <**td**><?**=$payment['credit\_month\_amount']**?></**td**>  
 <**td**><?**=$payment['percent\_month\_amount']**?></**td**>  
 </**tr**>  
 <?**php endforeach;** ?>  
 </**table**>  
 </**div**>  
 <?**php endif;** ?>  
 </**form**>  
 </**div**>  
</**div**>

## Управление кредита в админ-панели

**<?php  
class** CreditsOptions **extends** ModelOptions  
{  
 **function** Init ()  
 {  
 $this->**vAdd** = **false**;  
 $this->**vDelete** = **false**;  
 $this->**eSave** = **false**;  
  
 { *//filters* $this->**vFilters** = **array**(  
 );  
 }  
  
*// if ($this->Core->User->UserData['id\_level'] == USER\_LEVEL\_USER )* { *// edit* $this->**eFields** = **array**(  
 );  
 }  
  
 { *// view* $v\_select = **array**(  
 **'id'** => **''** ,  
 **'fio'** => **'50%'**,  
 **'title'** => **'50%'**,  
 **'amount'** => **'50%'**,  
 **'date\_start'** => **'50%'**,  
 **'date\_end'** => **'50%'**,  
 **'fine'** => **'50%'**,  
 **'paid'** => **'50%'**,  
 );  
  
 $where = **'true'**;  
 **if** ($this->**Core**->**User**->**UserData**[**'id\_level'**] == ***USER\_LEVEL\_USER***) {  
 **unset**($v\_select[**'fio'**]);  
 $where = **' user\_id = '** . $this->**Core**->**User**->**UserData**[**'id'**] . **' '**;  
 } **elseif** ($this->**Core**->**User**->**UserData**[**'id\_level'**] == ***USER\_LEVEL\_CREDIT\_OPERATOR***) {  
 $where = **' true '**;  
 } **else** {  
 $where = **' false '**;  
 }  
  
 $v\_length = **array**(**'title'** => 100);  
  
 $this->**vSQL** = **"  
 select  
 id,  
 fio,  
 title,  
 amount,  
 date\_start,  
 date\_end,  
 fine,  
 IF(paid, 'Да', 'Нет') paid  
 from  
 wf\_credits  
 where** $where  
 **"**;  
 $this->**vOrder** = **'id'**;  
 $this->**vOrderType** = **'DESC'**;  
 $this->MakeVSelect($v\_select, $v\_length);  
*// $this->vFieldsFunc['cdate'] = 'dateconvert';* }  
 { *//* $this->**eTitleField** = **'title'**;  
 $this->**ControlName** = **'credits'**;  
 }  
 }   
}

## Модель заявок на кредиты

**<?php  
  
class** Applications **extends** Model  
{  
 **public function** \_\_construct()  
 {  
 **parent**::*\_\_construct*(**'Applications'**, **'wf\_applications'**);  
   
 $this->**uri** = **false**;  
 $this->**savetime** = **false**;  
 }  
  
 */\*\*  
 \* Custom On Add Validation  
 \*  
 \** ***@param*** *array $arr  
 \** ***@return*** *bool  
 \*/* **public function** onAdd(&$arr) {  
 $arr = $this->updateData($arr);  
 $arr[**'user\_id'**] = $this->**Core**->**User**->**UserData**[**'id'**];  
 $arr[**'cdate'**] = *date*(**'Y-m-d H:i:s'**);  
 $arr[**'udate'**] = *date*(**'Y-m-d H:i:s'**);  
 **return true**;  
 }  
  
 **public function** onUpdate($item\_id, &$arr) {  
  
 $statuses = **array**(  
 0 => **'Черновик'**,  
 1 => **'Отправлена (этап 1)'**,  
 2 => **'Проверена специалистом по работе с клиентами (этап 2)'**,  
 3 => **'Одобрена сотрудником службы безопасности (этап 3)'**,  
 4 => **'Одобрена членом кредитного комитета (этап 4)'**,  
 5 => **'Утверждена начальником кредитного отдела (этап 5)'**,  
 6 => **'Операционист выдал деньги (этап 6)'**,  
 100 => **'Отказано'**,  
 );  
  
 $errorText = **'Вы можете только отклонить или одобрить заявку'**;  
  
 **if** ($this->**Core**->**User**->**UserData**[**'id\_level'**] == ***USER\_LEVEL\_USER*** ) {  
 $res = $this->getByID($item\_id);  
 **if** ($res[**'status'**] != 0) {  
 $this->**lastError** = **'Нельзя менять отправленные заявки'**;  
 **return false**;  
 } **elseif** ($arr[**'status'**] != 1 && $arr[**'status'**] != 0) {  
 $this->**lastError** = **'Пользователи могут только отправлять заявки или редактировать черновики'**;  
 **return false**;  
 }  
 } **elseif** ($this->**Core**->**User**->**UserData**[**'id\_level'**] == ***USER\_LEVEL\_CLIENT\_SERVICES*** ) {  
 **if** (!*in\_array*($arr[**'status'**], **array**(1, 2, 100))) {  
 $this->**lastError** = $errorText;  
 **return false**;  
 }  
 } **elseif** ($this->**Core**->**User**->**UserData**[**'id\_level'**] == ***USER\_LEVEL\_SECURITY*** ) {  
 **if** (!*in\_array*($arr[**'status'**], **array**(2, 3, 100))) {  
 $this->**lastError** = $errorText;  
 **return false**;  
 }  
 } **elseif** ($this->**Core**->**User**->**UserData**[**'id\_level'**] == ***USER\_LEVEL\_CREDIT\_COMMITTEE*** ) {  
 **if** (!*in\_array*($arr[**'status'**], **array**(3, 4, 100))) {  
 $this->**lastError** = $errorText;  
 **return false**;  
 }  
 } **elseif** ($this->**Core**->**User**->**UserData**[**'id\_level'**] == ***USER\_LEVEL\_CREDIT\_DEPARTMENT\_MANAGER*** ) {  
 **if** (!*in\_array*($arr[**'status'**], **array**(4, 5, 100))) {  
 $this->**lastError** = $errorText;  
 **return false**;  
 }  
 } **elseif** ($this->**Core**->**User**->**UserData**[**'id\_level'**] == ***USER\_LEVEL\_CREDIT\_OPERATOR*** ) {  
 **if** (!*in\_array*($arr[**'status'**], **array**(5, 6, 100))) {  
 $this->**lastError** = $errorText;  
 **return false**;  
 }  
 }  
  
 $arr = $this->updateData($arr);  
 $arr[**'udate'**] = *date*(**'Y-m-d H:i:s'**);  
  
 **if** ($arr[**'status'**] == 6) {  
 $this->createCredit($item\_id, $arr);  
 }  
  
 **return true**;  
 }  
  
 **protected function** updateData($arr) {  
 $arr[**'percent'**] = $arr[**'percent'**] ?: $this->**Core**->**Settings**->getSetting(**'percent\_year'**);  
 $arr[**'title'**] = **'Кредит на '** . $arr[**'term\_months'**] . **' месяцев под '** . $arr[**'percent'**] . **'%'**;  
 **return** $arr;  
 }  
  
 **protected function** createCredit($item\_id, $application)  
 {  
 $credits = Core::*getInstance*()->getModel(**'Credits'**); */\*\*****@var*** *Credits $credits\*/* **if** (!$credits->getCount(**array**(**'application\_id'** => $application[**'id'**]))) {  
  
 $creditInfo = $credits->getCreditInfo(  
 $application[**'term\_months'**],  
 $application[**'amount'**],  
 $application[**'percent'**]  
 );  
  
 $arr = **array**();  
 $arr[**'date\_start'**] = $creditInfo[**'date\_start'**];  
 $arr[**'date\_end'**] = $creditInfo[**'date\_end'**];  
 $arr[**'application\_id'**] = $item\_id;  
 $arr[**'title'**] = $application[**'title'**];  
 $arr[**'fio'**] = $application[**'fio'**];  
 $arr[**'amount'**] = $creditInfo[**'amount'**];  
 $arr[**'user\_id'**] = $this->**Core**->**User**->**UserData**[**'id'**];  
 $creditId = $credits->addItem($arr);  
  
 $CreditsPlan = Core::*getInstance*()->getModel(**'CreditsPlan'**);  
 **foreach** ($creditInfo[**'plans'**] **as** $plan) {  
 $payArr = **array**();  
 $payArr[**'amount'**] = $plan[**'amount'**];  
 $payArr[**'date'**] = $plan[**'date'**];  
 $payArr[**'title'**] = $payArr[**'date'**] . **'. Оплата '** . $payArr[**'amount'**] . **' рублей'**;  
 $payArr[**'credit\_id'**] = $creditId;  
 $CreditsPlan->addItem($payArr);  
 }  
 }  
 }  
}

# Приложение 1. Архив с приложением

Полный архив с приложением tofi.zip, содержащий программный код и дамп базы данных.

## Тестовые данные

В разработанной базе данных доступны несколько тестовых пользователей для каждой роли:

1. user – user – роль «Пользователь»

2. client\_services - client\_services – роль « Специалист по работе с клиентами»

3. security – security – роль «Сотрудник службы безопасности»

4. credit\_committee – credit\_committee – роль «Член кредитного комитета»

5. department\_manager – department\_manager – роль «Начальник кредитного отдела»

6. operator – operator – роль «Операционист»

7. system – system – роль «Администратор»

## Работа скрипта начисления пени в тестовом режиме

Скрипт начисления пени /crontab/processDay.php работает в тестовом режиме – каждый запуск – новый день.