

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий  
институт

Вычислительная техника  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ А. И. Легалов  
подпись          инициалы, фамилия  
« \_\_\_\_ » \_\_\_\_\_ 2016 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника  
код и наименование специализации

Веб-система восстановления хронологии событий. Разработка базы данных  
тема

Пояснительная записка

Руководитель	_____	<u>ст. преп.</u>	<u>Л. В. Макуха</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		<u>В.В. Тахтобин</u>
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	<u>доцент, к.т.н</u>	<u>В. И. Иванов</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2016



Студенту Тахтобин Вадим Викторович

фамилия, имя, отчество

Группа КИ12-09Б Направление (специальность) 09.03.01

номер

код

Информатика и вычислительная техника

наименование

Тема выпускной квалификационной работы

Веб-система восстановления хронологии событий. Разработка базы данных

Утверждена приказом по университету № \_\_\_\_\_ от \_\_\_\_\_

Руководитель ВКР Макуха Л.В., старший преподаватель кафедры ВТ

фамилия, инициалы, должность, учёное звание и место работы

Исходные данные для ВКР задание на проектирование

Перечень разделов ВКР анализ задания на проектирование, разработка архитектуры и основных технических решений

Перечень графического материала демонстрационное видео, презентация в формате Power Point

Руководитель ВКР

подпись

Макуха Л.В.

фамилия, инициалы

Задание принял к исполнению

подпись

Тахтобин В.В.

фамилия, инициалы

« \_\_\_ » \_\_\_\_\_ 2016 г.

## РЕФЕРАТ

Пояснительная записка содержит 33 страницы, 15 рисунков, 7 источников.

Объектом исследования является разработка и создание веб-приложение.

При создании сайта были рассмотрены языки веб программирования (PHP, HTML, JavaScript, CSS).

Результатом данной работы является разработка веб-приложения, осуществляющего функцию визуализации информации о преступлении.

Итогом будет являться веб-приложение, которое создавалось с целью упростить работу следователям, благодаря визуальному отображению общей картины с местом преступления.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Анализ задания на проектирование.....	5
1.1 Выбор картографического сервиса .....	5
1.1.1 Картографический сервис Google Maps .....	5
1.1.2 Картографический сервис Яндекс.Карты.....	6
1.1.3 Картографический сервис 2ГИС .....	6
1.1.4 Обоснования выбора картографического сервиса .....	7
1.2 Выбор средств разработки .....	7
1.2.1 Язык программирования PHP.....	7
1.2.2 Система управления базами данных MySQL .....	8
1.2.3 Язык программирования JavaScript .....	9
1.2.4 Обоснование выбора средств разработки .....	11
1.3 Техническое задание.....	11
1.3.1 Назначение и цели создания веб-приложения.....	11
1.3.2 Требование к веб-приложению .....	11
1.3.3 Требования к видам обеспечения.....	13
1.4 Вывод по разделу .....	13
2 Разработка архитектуры и основных технических решений.....	14
2.1 Функциональная модель приложения .....	14
2.2 Информационная структура веб-приложения .....	14
2.2.1 Диаграмма прецедентов .....	14
2.3 ER-модель созданной базы данных .....	15
2.4 Разработка функции записи дел и пользователей. ....	16
2.4.1 Структура записи дел и пользователей .....	16
2.4.2 Программная реализация функции записи дел и пользователей.....	16
2.4.3 Результат работы функции записи дел и пользователей .....	19
2.5 Разработка функции удаления дел и пользователей. ....	21
2.5.1 Структура удаления дел и пользователей .....	21
2.5.2 Программная реализация функции удаления дел и пользователей .	22

2.5.3 Результат работы функции удаления дел и пользователей.....	25
2.6 Разработка функции записи маркеров.....	27
2.6.1 Структура записи маркеров.....	27
2.6.2 Программная реализация функции записи маркеров.....	27
2.6.3 Результат работы функции записи маркеров.....	30
ЗАКЛЮЧЕНИЕ.....	32
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	33

## ВВЕДЕНИЕ

В наше время информационные технологии развиваются все с большей и большей скоростью. Они имеют свое применение во многих сферах, будь то медицина или образование, но хотелось бы сфокусировать внимание именно на раскрытии преступлений. Уровень их раскрытия, конечно, вырос за предыдущие годы, но все равно большое количество преступлений остается нераскрытыми.

Поэтому система, которая рассматривается в данной работе, будет более чем актуальна. Любому следователю был бы рад системе, которая отображает информацию о местоположении людей в момент совершения преступления, что позволит сэкономить время следователям, так как им будет удобнее расследовать дело с полностью отстроенной картиной произошедшего.

Предмет ВКР заключается в разработке веб-приложения для отображения информации об обстановке на месте преступления на картах Google Maps. Конечная цель данной выпускной работы – разработать эффективное приложение, с помощью которого можно будет считать данные с трекеров машин и отобразить их на карте, а также визуализировать местоположение людей в момент совершения преступления.

В связи с указанной целью возникают следующие задачи:

- изучить материалы необходимые для разработки веб-приложения;
- научиться работать с картографическим сервисом;
- реализовать функцию добавления маркеров на карту.

Создание такого приложения довольно сильно экономит время следователям, так как им будет удобнее расследовать дело с полностью отстроенной картиной произошедшего.

## **1 Анализ задания на проектирование**

Необходимо разработать веб-приложение для визуализации информации об обстановке в момент совершения преступления:

- отображение карты в веб-приложении;
- авторизация;
- возможность добавления треков машин на карту;
- возможность добавления маркеров свидетелей, преступлений и подозреваемых на карту;
- возможность добавления путей маркеров на карту;
- выбор преступления;
- запись координат и информации маркеров в базу данных;
- чтение информации о маркерах с базы данных;
- отображение маркеров и треков на карте.

### **1.1 Выбор картографического сервиса**

Существует большое количество картографических сервисов, но только некоторые из них подходят для нашей работы:

- Google Maps;
- Яндекс.Карты;
- 2ГИС.

#### **1.1.1 Картографический сервис Google Maps**

Карты Google — набор приложений, построенных на основе бесплатного картографического сервиса и технологии, предоставляемых компанией Google. Созданы в 2005 году.



Достоинства:

- лучшее покрытие всего мира;
- обзорные фотографии улиц;
- интеллектуальный поиск;
- голосовой ввод.

Недостатки:

- большой размер данных при использовании офлайн;
- нет возможности общения между пользователями [1].

### **1.1.2 Картографический сервис Яндекс.Карты**

Яндекс.Карты — поисково-информационная картографическая служба Яндекса. Открыт в 2004 году.

Достоинства:

- лучшее покрытие России;
- хорошая детализация России;
- отображение пробок в крупных городах.

Недостатки:

- нет ночного режима;
- нет детализации на уровне зданий;
- большой размер данных при использовании офлайн [1].

### **1.1.3 Картографический сервис 2ГИС**

2ГИС — международная картографическая компания, выпускающая одноимённые электронные справочники с картами городов с 1999 года.

Достоинства:

- одна из лучших детализаций в городах присутствия;
- возможность загрузки и использования офлайн;

- подробная информация об организациях.

Недостатки:

- уступает конкурентам в покрытии как в России, так и в других странах;
- нет возможности общения между пользователями;
- нет обзорных фотографий улиц;
- нет ночного режима [1].

#### **1.1.4 Обоснования выбора картографического сервиса**

Для выполнения данной работы был выбран картографический сервис Google Maps в связи с удобным инструментарием для разработки веб-приложения.

### **1.2 Выбор средств разработки**

На сегодняшний день существует множество языков программирования. Каждый из языков удобен для той или иной ситуации, однако для написания веб-приложения в основном используются следующие языки:

- PHP;
- MySQL;
- JavaScript.

#### **1.2.1 Язык программирования PHP**

PHP — является распространенным интерпретируемым языком общего назначения с открытым исходным кодом. PHP создавался специально для ведения web-разработок и код на нем может внедряться непосредственно в HTML-код. Синтаксис языка берет начало из C, Java и Perl, и является легким

для изучения. Основной целью PHP является предоставление web-разработчикам возможности быстрого создания динамически генерируемых web-страниц, однако область применения PHP не ограничивается только этим [2].

Синтаксис PHP подобен синтаксису языка Си.

Для работы программы не требуется описывать какие-либо переменные, используемые модули и т. п. Любая программа может начинаться непосредственно с оператора PHP.

Достоинства:

- подключается к любым БД;
- свободный язык программирования и не требует лицензий на разработки на нем;
- можно запускать из консоли как обычно так и через планировщик cron что позволяет управлять рассылками и другими событиями.

Недостатки:

- процедурный стиль работы со строками/массивами;
- порядок параметров;
- скорость выполнения [3].

### **1.2.2 Система управления базами данных MySQL**

MySQL — свободная реляционная система управления базами данных.

Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц [4].

Достоинства:

- обеспечивается поддержкой большого количества типов таблиц;
- СУБД MySQL поставляется со специальным типом таблиц EXAMPLE демонстрирующим принципы создания новых типов таблиц;
- благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Недостатки:

- наиболее быстрый тип таблиц MyISAM имеет невысокую надежность;
- таблицы типа MyISAM не поддерживают транзакции [4].

### **1.2.3 Язык программирования JavaScript**

JavaScript — прототипно-ориентированный сценарный язык программирования.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке [5].

Достоинства:

- объекты, с возможностью интроспекции;
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическая сборка мусора;
- анонимные функции.

Недостатки:

- модульная система: JavaScript не предоставляет возможности управлять зависимостями и изоляцией областей видимости;
- стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных;
- стандартные интерфейсы к веб-серверам и базам данных;
- система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их [6].

## **1.2.4 Обоснование выбора средств разработки**

Для выполнения данной работы был выбран PHP вместе с базой данных MySQL, так как они хорошо взаимодействуют друг с другом и довольно просты в написании веб-приложения, а JavaScript был выбран для удобной работы с Google Maps. Их минусы не являются критичными для данной работы.

## **1.3 Техническое задание**

### **1.3.1 Назначение и цели создания веб-приложения**

#### **1.3.1.1 Назначение веб-приложения**

Веб-приложение — это вспомогательные программные средства, которые предназначены для автоматизированного выполнения каких-либо действий.

#### **1.3.1.2 Цель создания веб-приложения**

Основной целью создания веб-приложения является визуализация информации о преступлениях на картах Google Maps. Это позволит следователям проще и эффективнее изучить информацию по делу.

### **1.3.2 Требование к веб-приложению**

#### **1.3.2.1 Требования к структуре и функциональной части веб-приложения**

Функциональная структура веб-приложения должна включать в себя:

- запускаться в любом браузере;

- дружелюбный интерфейс;
- малую ресурсоемкость;
- быстроту работы;
- загрузка информации из БД;
- простое и понятное отображение информации на карте.

### **1.3.2.2 Перспективы развития веб-приложение**

Добавление новых функций в веб-систему должно проходить с минимальными добавлениями в основной код.

### **1.3.2.3 Требования к надежности**

#### **Показатель надежности**

Время восстановления работоспособности веб-приложения при любых сбоях и отказах не должно превышать одного рабочего дня, исключая случаи неисправности серверного оборудования.

### **1.3.2.4 Требования по эргономике и технической эстетике**

#### **Требования к внешнему оформлению**

При разработке визуальной части программы должны учитываться психофизиологические особенности человека:

- не слишком яркая цветовая палитра;
- визуальная часть веб-приложения должно сочетаться с визуальной частью сайта;
- информация должно выводиться простым и понятным способом;
- положение визуальной составляющей веб-приложения должно соответствовать структуре сайта.

### **1.3.3 Требования к видам обеспечения**

#### **1.3.3.1 Требования к лингвистическому обеспечению**

##### **Языки программирования**

Для удобной работы с Google Maps были использованы следующие языки программирования:

- PHP;
- MySQL;
- JavaScript.

##### **Языки взаимодействия пользователей и системы**

Основным языком взаимодействия пользователей и системы является русский язык.

#### **1.3.3.2 Требования к программному обеспечению**

Программное обеспечение должно быть представлено в виде веб-приложения.

### **1.4 Вывод по разделу**

В результате анализа задания на дипломное проектирование были сформулированы четкие требования к разрабатываемой системе, было принято важное техническое решение - разрабатываемая система будет получать информацию из БД для дальнейшего занесения на карту Google Maps.



## 2 Разработка архитектуры и основных технических решений

### 2.1 Функциональная модель приложения

Цель работы - является разработка веб-приложения для автоматизации процесса поиска свидетелей. С помощью данного веб-приложения следователи смогут визуализировать местоположение людей в момент совершения преступления, что позволит сэкономить время следователям, так как им будет удобнее расследовать дело с полностью отстроенной картиной произошедшего.

Для хранения информации о метках и пользователях мы используем базу данных MySQL. Благодаря базе данных мы можем запоминать их координаты и загружать на карту в следующих сессиях.

### 2.2 Информационная структура веб-приложения

Для начала необходимо составить информационную модель.

#### 2.2.1 Диаграмма прецедентов

В разрабатываемом веб-сервере выделены 2 роли пользователь и администратор. Диаграмма прецедентов представлена на рисунке 1.

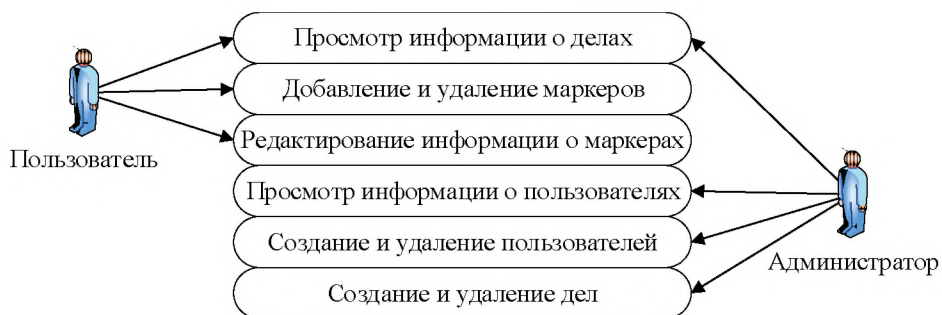


Рисунок 1 - Диаграмма прецедентов

Пользователь имеет функции:

- просмотр информации о делах;
- добавление и удаление маркеров;
- редактирование информации о маркерах.

Администратор имеет функции:

- просмотр информации о пользователях;
- создание и удаление пользователей;
- просмотр информации о делах;
- создание и удаление дел.

### 2.3 ER-модель созданной базы данных

На рисунке 2 показана ER-модель уже созданной базы данных при помощи СУБД MySQL

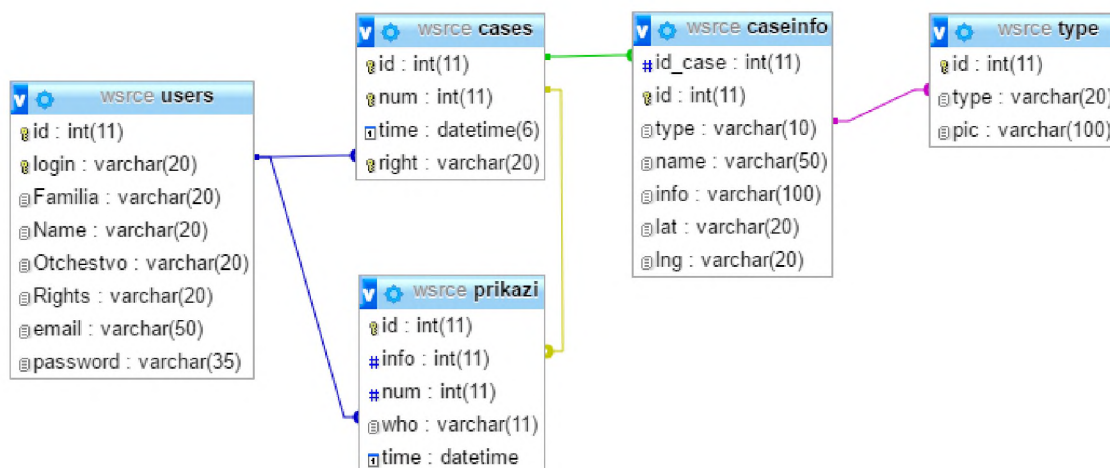


Рисунок 2 — ER-модель базы данных

Данная ER-модель представляет собой базу данных, хранящую информацию о пользователях, уголовных делах и маркерах.

Например, таблица «cases» хранит в себе информацию уголовных делах и ответственных за них.

## 2.4 Разработка функции записи дел и пользователей.

### 2.4.1 Структура записи дел и пользователей

При добавлении нового дела или пользователя мы подключаемся к нашей базе данных и записываем данные в соответствующие ячейки. Структура сохранения дел и пользователей показана на рисунке 3.



Рисунок 3 - Структура сохранения дел и пользователей

### 2.4.2 Программная реализация функции записи дел и пользователей

Для записи нового пользователя и дела были созданы два файла «adduser.php» и «addcase.php».

Рассмотрим файл «adduser.php»

Подключение к базе данных.

```
$connection = mysqli_connect('localhost', 'pma', '1', 'wsrce') or die(mysqli_error());  
if (isset($_POST['back']))  
{  
    header("Location: users.php");  
}  
if (isset($_POST['submit']))  
{  
    if (empty($_POST['login']))  
{
```

```

    $info_reg = 'Вы не ввели Логин';
}
elseif (empty($_POST['email']))
{
    $info_reg = 'Вы не ввели почту';
}
elseif (empty($_POST['password']))
{
    $info_reg = 'Вы не ввели пароль';
}
else
{

```

Присвоение написанного переменным

```

    $login = $_POST['login'];
    $email = $_POST['email'];
    $password = $_POST['password'];

```

Запрос в базу данных

```

    $query = "INSERT INTO `users` (login, email, password)
    VALUES ('$login', '$email', '$password')";

```

Инициализация запроса

```

    $result = mysqli_query($connection, $query) or die(mysqli_error());
    $info_reg = 'Вы успешно зарегистрировались!';
}
}

```

```

$info_reg = isset($info_reg) ? $info_reg : NULL;

```

```

echo $info_reg;

```

```

?>

```

```

</body>

```

```

</html>

```

Рассмотрим файл «addcase.php».

```

<html>

```

```

<head>

```

```

    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />

```

```

    <title>Создание уголовного дела</title>

```

```
<link rel='stylesheet prefetch' href='http://netdna.bootstrapcdn.com/font-awesome/3.2.0/css/font-awesome.min.css'>
```

```
<link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
```

```
</head>
```

```
<body>
```

```
<div class="box">
```

```
<div class="containerWrapper">
```

```
<div id="containerRegister" class="tabContainer">
```

```
<form action="addcase.php" method="POST">
```

```
<h2 class="loginTitle">Создание уголовного дела</h2>
```

```
<div class="registerContent">
```

```
<div class="inputWrapper">
```

```
<input type="text" name="case" placeholder="Номер дела" />
```

```
</div>
```

```
<button class="redBox" name="back"><span class="iconBack"></span>
```

```
Назад</button>
```

```
<button class="greenBox" name="submit"><span class="iconAdd"></span>
```

```
Создать уголовное дело</button>
```

```
<div class="clear"></div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<?php
```

```
Подключение к базе данных
```

```
$connection = mysqli_connect('localhost', 'pma', '1', 'wsrce') or die(mysqli_error());
```

```
if (isset($_POST['back']))
```

```
{
```

```
header("Location: cases.php");
```

```
}
```

```
if (isset($_POST['submit']))
```

```
{
```

```
if (empty($_POST['case']))
```

```
{
```

```

        $info_reg = 'Вы не ввели Номер дела';
    }
    else
    {
        $case = $_POST['case'];
        Запрос в базу данных
        $query = "CREATE TABLE `wsrce`.`$case` ( `id` INT(11) NOT NULL
        AUTO_INCREMENT, `type` VARCHAR(10) NOT NULL, `name` VARCHAR(50) NOT NULL
        ,
            `info` VARCHAR(100) NOT NULL, `lat` VARCHAR(20) NOT NULL,
            `lng` VARCHAR(20) NOT NULL, PRIMARY KEY (`id`)) ENGINE = InnoDB;";
        $result = mysqli_query($connection, $query) or die(mysqli_error());
        $info_reg = 'Дело успешно создано!';
    }
}
$info_reg = isset($info_reg) ? $info_reg : NULL;
echo $info_reg;
?>
</body>
</html>

```

### 2.4.3 Результат работы функции записи дел и пользователей

При вводе данных пользователя (рисунок 4) или дела с (рисунок 6) мы подключаемся к нашей базе данных и записываем информацию в соответствующие ячейки, что мы видим на рисунке 5 и рисунке 7.

### Добавление пользователя

ThiefPie

Тахтобин

Вадим

Викторович

Следователь

masterner@mail.ru

.....

+ Добавить пользователя
← Назад

Рисунок 4 - Функция записи данных пользователя

id	login	Familia	Name	Otchestvo	Rights	email	password
1	HunteRVAD	Потапов	Вадим	Павлович	Ст. Следователь	djsilenthunter@gmail.com	██████████
2	ThiefPie	Тахтобин	Вадим	Викторович	Следователь	masterner@mail.ru	██████████
3	sidorov	Сидоров	Анатолий	Юрьевич	Администратор	sidorov@mail.ru	██████████

Рисунок 5 - Записанная информация в базе данных

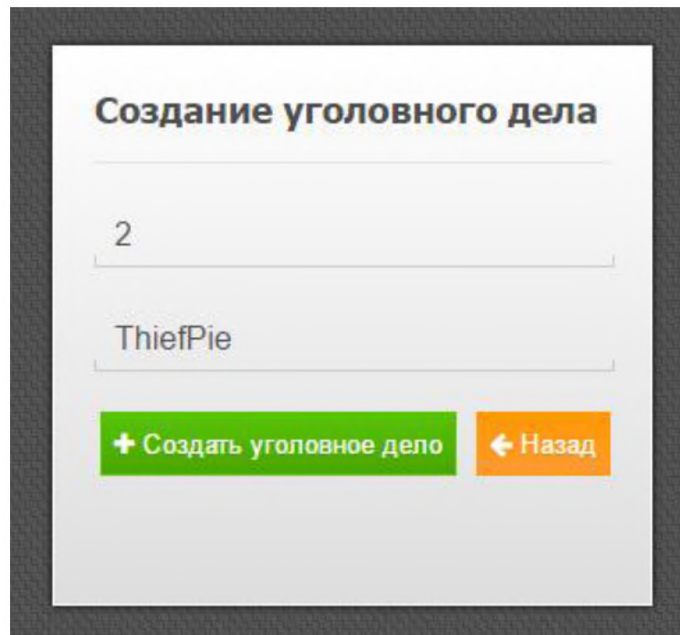


Рисунок 6 - Добавление нового уголовного дела

				id	num	time	right			
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	1	2016-06-14 20:11:17.000000	HunteRVAD
<input type="checkbox"/>		Изменить		Копировать		Удалить	2	2	2016-06-14 20:29:50.000000	ThiefPie

Рисунок 7 - Созданное дело в базе данных

## 2.5 Разработка функции удаления дел и пользователей.

### 2.5.1 Структура удаления дел и пользователей

Перед удалением пользователя или дела мы авторизуемся в качестве администратора, после мы вводим необходимые данные, подключаемся к нашей базе данных, находим соответствующие значения и удаляем запись из таблиц. Структуры последовательности наших шагов мы можем увидеть на рисунке 8.



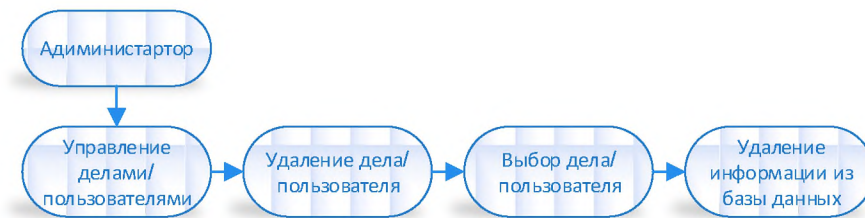


Рисунок 8 - Структура функции удаления дел и пользователей

## 2.5.2 Программная реализация функции удаления дел и пользователей

Для реализации данной функции были созданы два файла «deleteuser.php» и deletecase.php».

Рассмотрим код функции удаления пользователя «deleteuser.php».

```

<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <title>Удаление пользователя</title>
  <link rel='stylesheet prefetch' href='http://netdna.bootstrapcdn.com/font-
awesome/3.2.0/css/font-awesome.min.css'>
  <link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
</head>
<body>
<div class="box">
  <div class="containerWrapper">
    <div id="containerDelete" class="tabContainer">
      <form action="deleteuser.php" method="POST">
        <h2 class="loginTitle">Удаление пользователя</h2>
        <div class="registerContent">
          <div class="inputWrapper">
            <input type="text" name="login" placeholder="Логин" />
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <button          class="deleteBox"          name="delete"><span
class="iconDelete"></span> Удалить пользователя</button>
        <button          class="redBox2"           name="back"><span
class="iconBack"></span> Назад</button>

```

```

    </div>

```

```

</div>

```

```

<?php

```

```

    $connection = mysqli_connect('localhost', 'pma', '1', 'wsrce') or die(mysqli_error());

```

```

    if (isset($_POST['back']))

```

```

    {

```

```

        header("Location: users.php");

```

```

    }

```

```

    if (isset($_POST['delete']))

```

```

    {

```

```

        if (empty($_POST['login']))

```

```

        {

```

```

            $info_reg = 'Вы не ввели Логин!';

```

```

        }

```

```

        else

```

```

        {

```

```

            $login = $_POST['login'];

```

Удаление данных о пользователе из базы данных

```

            $query = "DELETE FROM users WHERE login = '$login'";

```

```

            $result = mysqli_query($connection, $query) or die(mysqli_error());

```

```

            $info_reg = 'Пользователь успешно удален!';

```

```

        }

```

```

    }

```

Рассмотрим файл с функцией удаления дел deletecase.php»

```

<html>

```

```

<head>

```

```

    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />

```

```

    <title>Удаление уголовного дела</title>

```

```
<link rel='stylesheet prefetch' href='http://netdna.bootstrapcdn.com/font-  
awesome/3.2.0/css/font-awesome.min.css'>
```

```
<link rel="stylesheet" href="css/style.css" media="screen" type="text/css" />
```

```
</head>
```

```
<body>
```

```
<div class="box">
```

```
<div class="containerWrapper">
```

```
<div id="containerRegister" class="tabContainer">
```

```
<form action="deletecase.php" method="POST">
```

```
<h2 class="loginTitle">Удаление уголовного дела</h2>
```

```
<div class="registerContent">
```

```
<div class="inputWrapper">
```

```
<input type="text" name="case" placeholder="Номер дела" />
```

```
</div>
```

```
<button class="deleteBox" name="delete"><span  
class="iconDelete"></span> Удалить уголовное дело</button>
```

```
<button class="redBox2" name="back"><span  
class="iconBack"></span> Назад</button>
```

```
<div class="clear"></div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<?php
```

```
$connection = mysqli_connect('localhost', 'pma', '1', 'wsrce') or die(mysqli_error());
```

```
if (isset($_POST['back']))
```

```
{
```

```
header("Location: cases.php");
```

```
}
```

```
if (isset($_POST['delete']))
```

```
{
```

```
if (empty($_POST['case']))
```

```
{
```

```
$info_reg = 'Вы не ввели Номер дела';
```

```

}
else
{
    $case = $_POST['case'];
    Удаление данных о деле из базы данных
    $query = "DROP TABLE $case";
    $result = mysqli_query($connection, $query) or die(mysqli_error());
    $info_reg = 'Дело успешно удалено!';
}
$info_reg = isset($info_reg) ? $info_reg : NULL;
echo $info_reg;
?>
</body>
</html>

```

### 2.5.3 Результат работы функции удаления дел и пользователей

При вводе данных пользователя (рисунок 9) или дела (рисунок 11) мы подключаемся к нашей базе данных и удаляем информацию из соответствующей таблицы, что показано на рисунке 10 и рисунке 12.

Рисунок 9 - Ввод данных о пользователе

id	login	Familia	Name	Otchestvo	Rights	email	password
1	HunteRVAD	Потапов	Вадим	Павлович	Ст. Следователь	djsilenthunter@gmail.com	██████████
2	ThiefPie	Тахтобин	Вадим	Викторович	Следователь	mastermer@mail.ru	██████████
3	sidorov	Сидоров	Анатолий	Юрьевич	Администратор	sidorov@mail.ru	██████████

Рисунок 10 - Удаление уголовного дела

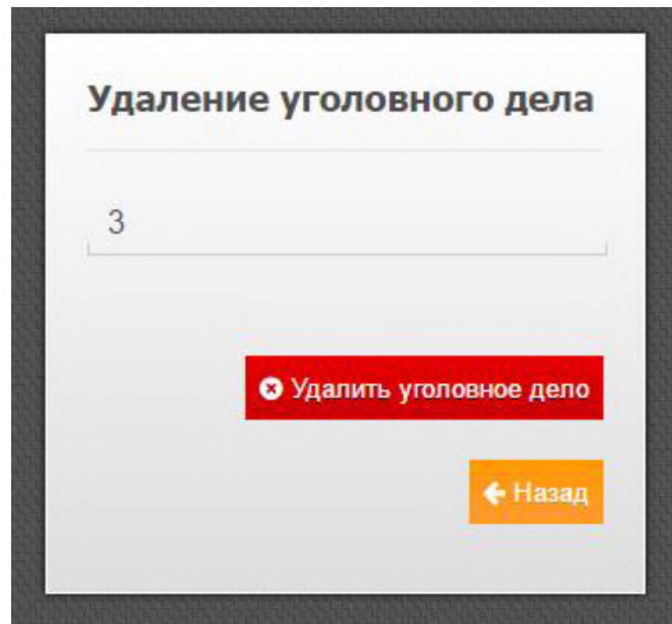


Рисунок 11 - Ввод данных о уголовном деле

				id	num	time	right
<input type="checkbox"/>		Изменить		Копировать		Удалить	1 1 2016-06-14 20:11:17.000000 HunteRVAD
<input type="checkbox"/>		Изменить		Копировать		Удалить	2 2 2016-06-14 20:29:50.000000 ThiefPie
<input type="checkbox"/>		Изменить		Копировать		Удалить	4 4 2016-06-16 13:19:36.000000 HunteRVAD

Рисунок 12 - Удаление пользователя

## 2.6 Разработка функции записи маркеров.

### 2.6.1 Структура записи маркеров

Для работы с маркерами мы авторизуемся в качестве пользователя и выбираем необходимое нам дело после того, как наше дело загрузится мы можем нажать на нужное место карты для создания нового маркера. Его структура показана на рисунке 13.



Рисунок 13 - Структура записи маркеров

### 2.6.2 Программная реализация функции записи маркеров

Для реализации данной функции были созданы два файла «phpsql.php» и «phpxml.php».

Рассмотрим файл с функцией сохранения маркеров в базу данных «phpsql.php».

```
<?php
session_start();
?>
<?php
Получаем данные с URL
$name = $_GET['name'];
$info = $_GET['info'];
$lat = $_GET['lat'];
$lng = $_GET['lng'];
```

```
$type = $_GET['type'];
```

Подключаемся к нашей таблице

```
$connection=mysqli_connect ('localhost', 'root', '');
```

```
if (!$connection) {
```

```
    die('Not connected : '. mysql_error());
```

```
}
```

Выбираем нужную базу данных

```
$db_selected = mysqli_select_db($connection, 'wsrce');
```

```
if (!$db_selected) {
```

```
    die ('Can\'t use db : '. mysql_error());
```

```
}
```

```
$choice=$_SESSION['choice'];
```

Вставляем данные в новую строку базы данных

```
$query = sprintf("INSERT INTO $choice" .
```

```
    " (id, name, info, lat, lng, type) " .
```

```
    " VALUES (NULL, '%s', '%s', '%s', '%s', '%s');" ,
```

```
    mysqli_real_escape_string($connection, $name),
```

```
    mysqli_real_escape_string($connection, $info),
```

```
    mysqli_real_escape_string($connection, $lat),
```

```
    mysqli_real_escape_string($connection, $lng),
```

```
    mysqli_real_escape_string($connection, $type));
```

```
$result = mysqli_query($connection,$query);
```

```
if (!$result) {
```

```
    die('Invalid query: '. mysql_error());
```

```
}
```

```
?>
```

Рассмотрим файл с функцией загрузки маркера на карту «phrxml.php»

```
<?php
```

```
    session_start();
```

```
?>
```

```
<?php
```

```
function parseToXML($htmlStr)
```

```
{
```

```
    $xmlStr=str_replace('<','&lt;',$htmlStr);
```

```

$xmlStr=str_replace('>','&gt;',$xmlStr);
$xmlStr=str_replace('"','&quot;',$xmlStr);
$xmlStr=str_replace("'",'&#39;',$xmlStr);
$xmlStr=str_replace("&","&amp;",$xmlStr);
return $xmlStr;
}

```

Подключаемся к нашей базе данных

```

$connection=mysqli_connect ('localhost', 'root', '');
if (!$connection) {
    die('Not connected : '. mysql_error());
}
$db_selected = mysqli_select_db($connection, 'wsrce');
if (!$db_selected) {
    die ('Can\'t use db : '. mysqli_error());
}

```

Выбираем таблицу

```

$choice=$_SESSION['choice'];
$query = "SELECT * FROM $choice WHERE 1";
$result = mysqli_query($connection,$query);
if (!$result) {
    die('Invalid query: '. mysqli_error());
}
header("Content-type: text/xml; charset=utf-8");

```

Создаем динамический XML файл

```

echo '<case>';
while ($row = @mysqli_fetch_assoc($result)){
    echo '<entity ';
    echo 'name="'. parseToXML($row['name']). ' " ';
    echo 'info="'. parseToXML($row['info']). ' " ';
    echo 'lat="'. $row['lat']. ' " ';
    echo 'lng="'. $row['lng']. ' " ';
    echo 'type="'. $row['type']. ' " ';
    echo '/>';
}

```



### 2.6.3 Результат работы функции записи маркеров

При загрузке карты маркеры размещаются в отмеченных нами местах как показано на рисунке 14. В дальнейшем веб-приложение подключается к нужной базе данных и сохраняет данные о новой метке в таблице, что видно на рисунке 15.

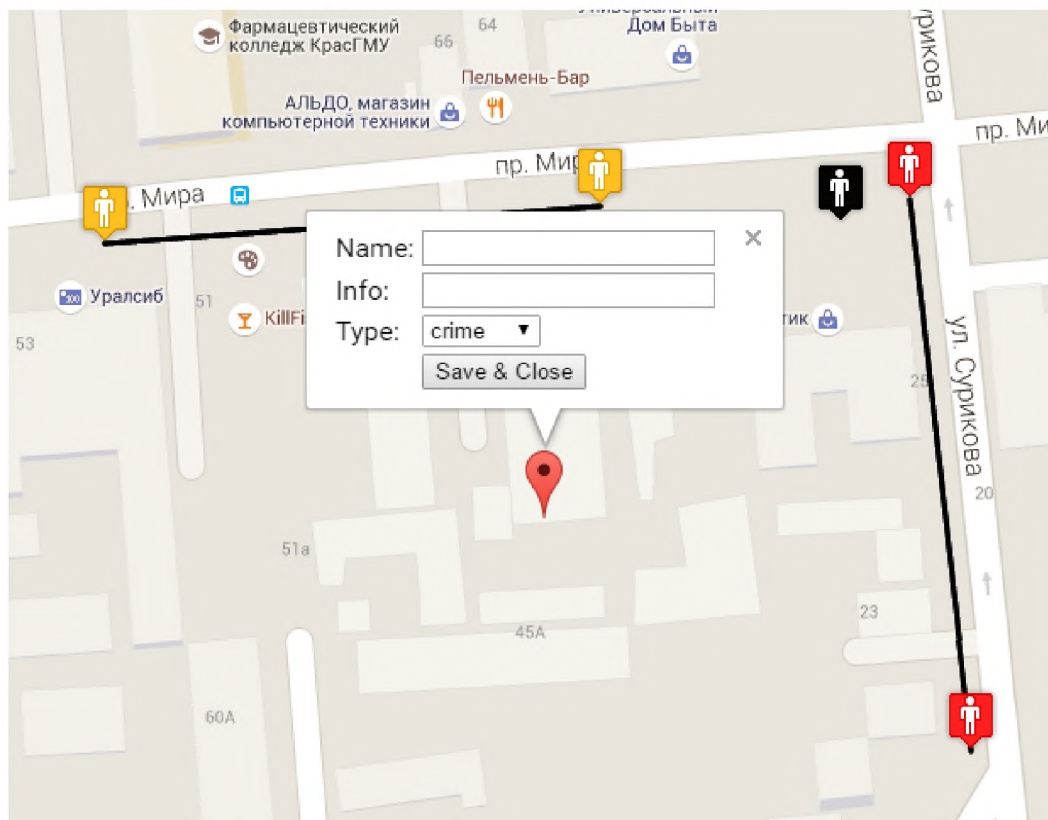


Рисунок 14 - Добавление маркера на карту.

id_case	id	type	name	info	lat	lng
1	1	crime	Colonel	Sanders	56.01217189716075	92.87597179412842
1	2	witness	McDonald	06.00	56.01213591154829	92.87364363670349
1	3	witness	McDonald	06.03	56.012201885145515	92.87521004676819
1	4	suspect	Jake King	06.00	56.011239259213696	92.87638485431671
1	5	suspect	Jake King	06.07	56.01221388033293	92.87619173526764
1	6	suspect	John Smith	8:10	56.00985075650451	92.85899877548218
1	7	suspect	John Smith	8:10	56.012867612412585	92.85852670669556
1	8	suspect	John Smith	8:20	56.013113508771774	92.86335468292236
2	9	crime	Lake Jackson	14.06.2016 20:36 Explosion	55.982742469217904	92.89602398872375
2	10	suspect	George Martin	20:38	55.982373334137826	92.89591133594513

Рисунок 15 - Данные добавленного маркера в базе данных.

## ЗАКЛЮЧЕНИЕ

В результате выполнения ВКР было разработано веб-приложение. Данная веб-система создана для следователей. С помощью разработанного приложения следователи смогут визуализировать местоположение людей в момент совершения преступления, что позволит сэкономить время работы над делом, так как им будет удобнее расследовать дело с полностью отстроенной картиной произошедшего правонарушения.

Также для полной работы веб-приложения потребуются администраторы, которые будут предоставлять доступ следователям и создавать новые дела.

При разработке веб-приложения было рассмотрено исчерпывающее количество современных технологий веб-разработки. Разработанное веб-приложение соответствует всем требованиям, приведенным в техническом задании.

В будущем возможно добавление большого количества функций, к примеру, таких как Google Maps API. Также возможна оптимизация кода и доработка интерфейса для уменьшения время загрузок и улучшения интуитивности.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Habrahabr.ru [Электронный ресурс]: Руководство по PHP Яндекс.Карты, 2ГИС или всё же Google Maps? - Режим доступа: <https://habrahabr.ru/post/242015>
2. PHP.net [Электронный ресурс]: Руководство по PHP - Режим доступа: <http://php.net/manual/ru/preface.php>
3. Клуб программистов [Электронный ресурс]: PHP, его минусы и плюсы - Режим доступа: <http://www.programmersclub.ru/php-его-минусы-и-плюсы>
4. Свободная энциклопедия [Электронный ресурс]: MySQL- Режим доступа: <https://ru.wikipedia.org/wiki/MySQL>
5. Свободная энциклопедия [Электронный ресурс]: JavaScript- Режим доступа: <https://ru.wikipedia.org/wiki/JavaScript>
6. J-S [Электронный ресурс]: Уроки по JavaScript - Режим доступа: <http://www.j-s.ru>
7. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 22.12.2014. – Красноярск : ИПК СФУ, 2014. – 41 с.