Департамент образования и науки Костромской области

Областное государственное бюджетное профессиональное образовательное

учреждение «Костромской энергетический техникум имени Ф.В. Чижова»

(ОГБПОУ «КЭТ имени Ф.В. Чижова»)

КУРСОВАЯ РАБОТА

Разработка информационной системы для

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ПМ.06 Сопровождение информационных систем

МДК.06.03 Устройство и функционирование информационной системы

специальность: 09.02.07 «Информационные системы и программирование»

Студент Барашков Александр Сергеевич \_\_\_\_\_\_\_\_\_\_\_ \_\_.\_\_.2023 г.

*Подпись*

Оценка выполнения и защиты курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_

Руководитель Бессараб С.К. **\_\_\_\_\_\_\_\_\_\_\_\_** \_\_.\_\_.2023 г.

*Подпись*

Кострома 2023

# **ГЛОССАРИЙ**

**Автоматизированная система управления (АСУ)** - комплекс аппаратных и программных средств, а также персонала, предназначенный для управления различными процессами в рамках технологического процесса, производства, предприятия [1].

**База данных (БД)** - совокупность взаимосвязанных данных, организованных в соответствии со схемой базы данных таким образом, чтобы с ними мог работать пользователь [2].

**Информационная система (ИС)** — система, которая организует хранение и манипулирование информацией о предметной области [2].

**Скорая медицинская помощь (СМП)** — вид медицинской помощи, оказываемой гражданам при заболеваниях, несчастных случаях, травмах, отравлениях и других состояниях, требующих срочного медицинского вмешательства [3].

**Станция скорой медицинской помощи** - государственное учреждение здравоохранения, оказывающее круглосуточную скорую медицинскую помощь, как заболевшим и пострадавшим, находящимся вне лечебно-профилактического учреждения, так и в пути следования в лечебно-профилактическое учреждение при состояниях, угрожающих здоровью или жизни граждан [4].

**ER-модель** — модель данных, позволяющая описывать концептуальные схемы предметной области [5].

# **ВВЕДЕНИЕ**

Скорая медицинская помощь – очень полезная служба, без которой многие больные и страдающие могли бы остаться без своевременной медпомощи. Но мало кто обращает внимание на то, что в медицине нужно вести множество учетов проделанной работы и скорая медицинская помощь не исключение. Ведение документации и отчетов в бумажном виде не эффективно. Бумага может потеряться, промокнуть, сгореть, испортиться. Скорость письма значительно уступает скорости печати. А скорость передачи документов в бумажном виде значительно медленнее скорости в электронном. К тому же документы занимают огромное количества пространства.

Именно поэтому информационная система учета вызовов скорой медицинской помощи актуальна для повышения эффективности работы сотрудников скорой помощи.

**Цель исследования:**

* автоматизация процесса учета вызовов скорой медицинской помощи;

**Объект исследования:** Работа фельдшера по приему вызовов скорой медицинской помощи и передаче их выездным бригадам

**Предмет исследования:** процесс учета вызовов скорой медицинской помощи.

**Задачи исследования:**

1. Обоснование необходимости внедрения и использования информационной системы;
2. Разработка общих требований к информационной системе;
3. Формирование кратких сведений об информационной системе;
4. Разработка технического задания на разработку информационной системы;
5. Анализ требований к базе данных;
6. Формирование логическая структура информационной системы;
7. Разработка запросов к информационной системе;
8. Разработка кода информационной системы;
9. Разработка плана тестирования информационной системы;
10. Написание тест-кейсов;
11. Разработка модульного тестирования.

**Заключение**: система, включающая в себя функции автоматизации ведения учетов вызовов скорой медицинской помощи в электронном виде, может быть полезна в области скорой медицинской помощи.

Структура работы соответствует логике исследования и включает в себя введение, практическую часть, заключение, список литературы, 5 приложений.

# **ОБОСНОВАНИЕ ОБЪЕКТА АВТОМАТИЗАЦИИ**

## **Обоснование необходимости внедрения и использования информационной системы**

Ведение документации и отчетов в бумажном виде не эффективно. Бумага может потеряться, промокнуть, сгореть, испортиться. Скорость письма значительно уступает скорости печати. А скорость передачи документов в бумажном виде значительно медленнее скорости в электронном. К тому же документы занимают огромное количества пространства.

Именно поэтому информационная система учета вызовов скорой медицинской помощи так необходима для повышения эффективности работы сотрудников скорой помощи.

информационная система будет реализована как web-сайт, так как web-сайт имеет следующие преимущества:

* создание сайта является более простой задачей, чем создания настольного или мобильного приложения;
* реализация системы в виде сайта позволит работать на ней с удаленными серверами;
* нет необходимости устанавливать систему на компьютер.

## **Общие требования к информационной системе**

**Бизнес-требования** — определяют набор бизнес задач, которые должна решать система, а также способы их решения.

**Пользовательские требования** — определяют набор пользовательских задач, которые должна решать программа, а также способы (сценарии) их решения в системе [6].

**Нефункциональные требования** - определяют свойства, которые система должна демонстрировать, или ограничения, которые она должна соблюдать, не относящиеся к поведению системы.

**Функциональные требования** - определяют свойства, которые система должна демонстрировать, или ограничения, которые она должна соблюдать, относящиеся к поведению системы [7].

В моей курсовой работе были определены следующие требования:

1. **Бизнес-требования**

* система должна позволять заполнять в электронном виде карту вызова скорой медицинской помощи;
* система должна хранить информацию в отдельной базе данных.

1. **Пользовательские требования**

* система должна выводить различные сообщения подтверждения или ошибки операции;
* в форме заполнения карты скорой медицинской помощи должны быть понятные вопросы и соответствующие варианты ответов;
* привилегированный пользователь должен иметь возможность добавления, редактирования и удаления аккаунтов.

1. **Нефункциональные требования**

* система должна быть выполнена в серых оттенках;
* система должна иметь русский интерфейс;
* на каждой странице должна быть кнопка, позволяющая выйти из системы;
* на каждой странице должна быть кнопка, позволяющая вернуться на предыдущую страницу.

1. **Функциональные требования**

* система должна иметь систему аутентификации;
* система должна позволять добавлять, редактировать и удалять аккаунты.
* система должна позволять добавлять, редактировать и удалять данные о вызовах скорой медицинской помощи.

## **Краткие сведения об информационной системе**

Для более простой разработки технического задания, необходимо определить общую информацию о будущей системе. Краткие сведения об информационной системе приведены в таблице ниже.

Таблица №1

Краткие сведения об информационной системе

| **Вопросы** | **Ответ** |
| --- | --- |
| Название проекта | Название – «Информационная система учета вызовов скорой медицинской помощи» («ИС учета вызовов СМП»). |
| Цель ИС | * автоматизация процесса учета вызовов скорой медицинской помощи; * формирование и хранение отчетов в электронном виде. |
| Задачи проекта | * решение проблем автоматизации статистических отчетов, учета и анализа работы станции скорой медицинской помощи; * оперативный учет принятых по вызовов; * ввод информации об обслуженных вызовах; |
| Платформа | Персональные компьютеры |
| Тип сайта | Закрытый web-сервис |

Краткие сведения об информационной системе приведены, что значит, что для разработки технического задания уже есть основа.

## **Разработка технического задания на разработку информационной системы**

**Техническое задание (ТЗ)** - исходный технический документ для проведения работы, устанавливающий требования к создаваемому изделию (его составная часть или комплектующие изделия межотраслевого применения) и технической документации на него, а также требования к объему, срокам проведения работы и форме представления результатов [8].

Составление технического задания необходимо для последующей разработки проекта, так как именно в нем детально прописываются функционал, дизайн и структура продукта, а также другая необходимая информация.

Перечень нормативных документов регламентирующих структуру и содержание технического задания:

* ГОСТ 34.602—2020

Настоящий стандарт распространяется на автоматизированные системы (АС), предназначенные для автоматизации различных видов деятельности (управление, проектирование, исследования и т. п.), включая их сочетания, и устанавливает требования к составу, содержанию, правилам оформления документа «Техническое задание на создание (развитие или модернизацию) автоматизированной системы» (далее — ТЗ на АС) [9].

* ГОСТ 19.201-78

Настоящий стандарт устанавливает порядок построения и оформления технического задания на разработку программы или программного изделия для вычислительных машин, комплексов и систем независимо от их назначения и области применения [10].

В качестве технического задания я буду использовать бриф.

**Бриф** – это документ, краткая письменная форма согласительного порядка между планирующими сотрудничать сторонами, в которой прописываются основные параметры будущего проекта.

Таблица №2

Бриф на разработку информационной системы

| **Вопрос** | **Ответ** |
| --- | --- |
| Название проекта | «Информационная система учета вызовов скорой медицинской помощи» («ИС учета вызовов СМП»). |
| Задачи проекта | * решение проблем автоматизации статистических отчетов, учета и анализа работы станции скорой медицинской помощи; * оперативный учет принятых по вызовов; * ввод информации об обслуженных вызовах; |
| примеры удачных работ | «МЕД-Комплит» («MED-Complete»), АСУ «Скорая помощь», «ГЛОНАСС скорая помощь» |
| Целевая аудитория (возраст, пол, род занятий) | Возраст – 30 – 48 лет.  Пол – любой.  Род занятий – работники СМП. |
| Платформа | Персональные компьютеры |
| Тип сайта | Закрытый web-сервис |
| Структура и функционал проекта | Страница аутентификации с полями ввода логина и пароля пользователя. При вводе корректных пароля и логина пользователь вход на главный экран.  Если пользователь входит как непривилегированный:  Главный экран со списком вызовов и с их основной информацией. Над каждой записью есть кнопки «редактировать» и «удалить». При нажатии на кнопки открываются соответствующие окна взаимодействия с записями. Также под списком вызовов есть кнопка «создать запись». При нажатии на кнопку открывается окно с формой вызова СМП и кнопкой «отправить», добавляющая запись в БД и возвращающая на главный экран.  Если пользователь входит как привилегированный:  Структура та же, что и у непривилегированного пользователя, за исключением того, что на главном экране есть также кнопка «менеджер аккаунтов», при нажатии на которую пользователь входит в раздел менеджера аккаунтов.  В этом разделе выведен список всех аккаунтов системы, за исключением аккаунта пользователя. Напротив информации о пользователе кнопки «редактировать» и «удалить». При нажатии на кнопки открываются соответствующие окна взаимодействия с аккаунтами. Также под списком аккаунтов есть кнопка «создать аккаунт».  На каждой из страниц, кроме страницы аутентификации, присутствует кнопка выхода из системы. Также на каждой страницы, кроме главной страницы и страницы менеджера аккаунтов есть кнопка «назад», возвращающая на предыдущую страницу. |
| Желаемая стилистика сайта | Строгая |
| Наличие графических элементов | Минимальное |
| Логотип | Требуется |
| Фирменные цвета | Не требуются |
| Фирменный стиль | Не требуются |
| Иллюстрации | Не требуются |
| Пиктограммы | Минимальное количество |
| Модули сайта | * главный экран; * менеджер аккаунтов (только для администратора). |
| Дополнительные пожелания. | * Функционалом системы может воспользоваться только авторизованный пользователь, после ввода логина и пароля. * Удалить пользователя или авторизовать нового может только привилегированный пользователь. |

Таким образом, в ходе работы была обоснована необходимость автоматизации данной отрасли, были разработаны различные требования к системе, а именно: функциональные и нефункциональные требования, бизнес требования и пользовательские требования. Разработано техническое задание к информационной системе.

# **РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

## **Анализ требований к базе данных**

В техническом задании были определены требования к системе в целом, однако для реализации базы данных необходимо определить требования к ней. Требования к базе данных приведены в таблице ниже.

Таблица №3

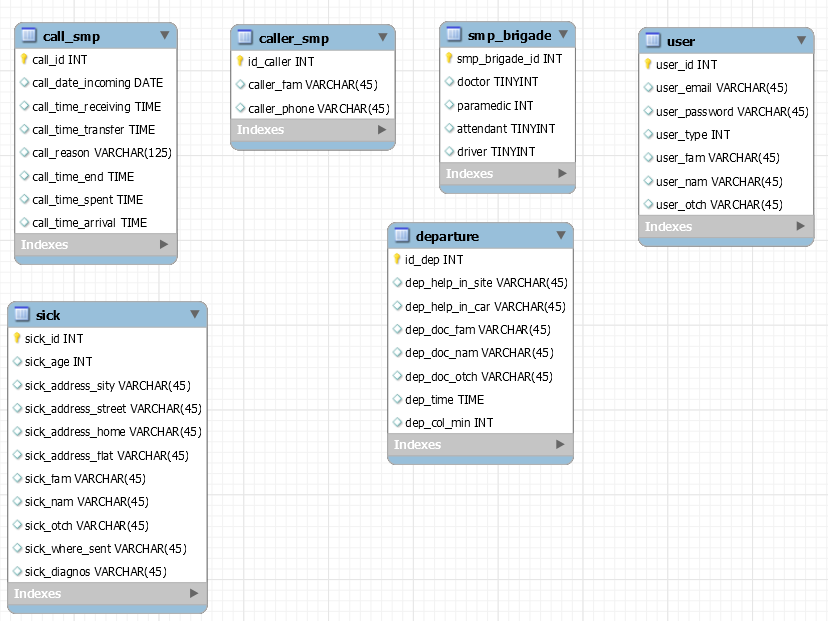
Требования к базе данных

| **№** | **Вопрос** | **Ответ** |
| --- | --- | --- |
| 1 | Для каких типов задач проектируется база данных? | * хранение общих сведений о вызове СМП; * хранение информации о больных; * хранение информации о вызывающих СМП; * хранение информации о составе бригады СМП * хранение информации о выезде бригады СМП; * хранение информации о пользователях системы. |
| 2 | Какими информационными объектами характеризуются эти задачи? | * журнал скорой медицинской помощи; * список сотрудников. |
| 3 | Каким текущим запросам должны удовлетворять данные информационные объекты? | * получение информации о вызове скорой медицинской помощи; * получение информации о пользователях системы. |
| 4 | Каким перспективным запросам должны удовлетворять информационные объекты в базе данных? | * редактирование, удаление и создание данных о вызове СМП; * вывод списка вызовов СМП в виде таблицы; * редактирование, удаление и создание пользователей; * вывод списка пользователей в виде таблицы. |

Разработав требования к базе данных, можно приступать к формированию логической структуры ИС.

## **Логическая структура информационной системы**

Для реализации информационной системы необходимо спроектировать базу данных, в соответствии с требованиями. В свою очередь, для того чтобы спроектировать базу данных нужно разработать модель, а именно ER-диаграмму.

*Рис. №1 “ER-диаграмма базы данных”*

Для полноты понимания ниже приведено описание основных сущностей и их атрибутов:

Таблица №4

Словарь базы данных

| **call\_smp** | | |
| --- | --- | --- |
| 1 | call\_id [INT] | id вызова |
| 2 | call\_date\_incoming [DATE] | дата поступления вызова |
| 3 | call\_time\_receiving [TIME] | дата приема вызова |
| 4 | call\_time\_transfer [TIME] | дата передачи вызова |
| 5 | call\_reason [VARCHAR(125)] | повод для вызова |
| 6 | call\_time\_end [TIME] | время окончания вызова |
| 7 | call\_time\_spent [TIME] | время, затраченное на вызов |
| 8 | call\_time\_arrival [TIME] | время доезда до места вызова |

| **caller\_smp** | | |
| --- | --- | --- |
| 1 | id\_caller [INT] | id вызывающего |
| 2 | caller\_fam [VARCHAR(45)] | фамилия вызывающего |
| 3 | caller\_phone [VARCHAR(45)] | номер телефона вызывающего |

| **smp\_brigade** | | |
| --- | --- | --- |
| 1 | smp\_brigade\_id [INT] | id бригады |
| 2 | doctor [TINYINT] | наличие доктора |
| 3 | paramedic [INT] | наличие фельдшеров |
| 4 | attendant [TINYINT] | наличие санитара |
| 5 | driver [TINYINT] | наличие водителя |

| **departure** | | |
| --- | --- | --- |
| 1 | id\_dep [INT] | id выезда |
| 2 | dep\_help\_in\_site [VARCHAR(45)] | оказанная помощь на месте |
| 3 | dep\_help\_in\_car [VARCHAR(45)] | оказанная помощь в машине |
| 4 | dep\_doc\_fam [VARCHAR(45)] | фамилия доктора (фельдшера), оказавшего скорую медицинскую помощь. |
| 5 | dep\_doc\_nam [VARCHAR(45)] | имя доктора (фельдшера), оказавшего скорую медицинскую помощь. |
| 6 | dep\_doc\_otch [VARCHAR(45)] | отчество доктора (фельдшера), оказавшего скорую медицинскую помощь. |
| 7 | dep\_time [TIME] | время выезда бригады СМП |
| 8 | dep\_col\_min [INT] | через сколько минут автомобиль СМП выехал на вызов |

| **sick** | | |
| --- | --- | --- |
| 1 | sick\_id [INT] | id больного |
| 2 | sick\_age [INT] | возраст больного |
| 3 | sick\_address\_sity [VARCHAR(45)] | адрес (город) больного |
| 4 | sick\_address\_streeet [VARCHAR(45)] | адрес (улица) больного |
| 5 | sick\_address\_home [VARCHAR(45)] | адрес (дом) больного |
| 6 | sick\_address\_flat [VARCHAR(45)] | адрес (квартира) больного |
| 7 | sick\_fam [VARCHAR(45)] | фамилия больного |
| 8 | sick\_nam [VARCHAR(45)] | имя больного |
| 9 | sick\_otch [VARCHAR(45)] | отчество больного |
| 10 | sick\_where\_sent [VARCHAR(45)] | куда направлен больной |
| 11 | sick\_diagnos [VARCHAR(45)] | диагноз больного |

| **user** | | |
| --- | --- | --- |
| 1 | user\_id [INT] | id пользователя |
| 2 | user\_email [VARCHAR(45)] | email пользователя |
| 3 | user\_password [VARCHAR(45)] | пароль пользователя |
| 4 | user\_type [INT] | тип пользователя |
| 5 | user\_fam [VARCHAR(45)] | фамилия пользователя |
| 6 | user\_nam [VARCHAR(45)] | имя пользователя |
| 7 | user\_otch [VARCHAR(45)] | отчество пользователя |

Словарь базы данных отражает сущности и атрибуты базы данных в удобной форме и позволяет облегчить ориентацию в ней, что в свою очередь позволяет облегчить формирование запросов к базе данных.

## **Запросы к информационной системе**

**CRUD** — акроним, обозначающий четыре базовые функции, используемые при работе с базами данных: создание (create), чтение (read), модификация (update), удаление (delete). Введен как стандартная классификация функций по манипуляции данными [11].

В данной курсовой работе реализованы запросы создания, чтения, редактирования и удаления данных базы данных. В качестве примера ниже приведены следующие запросы к базе данных:

ввод данных в таблицу “call\_smp”;

"INSERT INTO call\_smp (call\_date\_incoming, call\_time\_receiving, call\_time\_transfer, call\_reason,

call\_time\_end, call\_time\_spent, call\_time\_arrival)

VALUES ('$call\_date\_incoming', '$call\_time\_receiving', '$call\_time\_transfer', '$call\_reason', '$call\_time\_end',

'$call\_time\_spent', '$call\_time\_arrival');"

вывод всех данных из таблицы “call\_smp”;

"SELECT \* FROM call\_smp;"

Редактирование (модификация) строки данных таблицы “call\_smp”;

"UPDATE call\_smp SET

call\_date\_incoming = '$call\_date\_incoming', call\_time\_receiving = '$call\_time\_receiving', call\_time\_transfer = '$call\_time\_transfer',

call\_reason = '$call\_reason', call\_time\_end = '$call\_time\_end', call\_time\_spent = '$call\_time\_spent',

call\_time\_arrival = '$call\_time\_arrival'

WHERE (call\_id = '$id');"

удаление строки данных из таблицы “call\_smp”.

"DELETE FROM call\_smp WHERE (call\_id = '$id');

Таким образом, в ходе работы над проектом были определены требования к базе данных системы учета вызовов скорой медицинской помощи, была реализована база данных, соответствующая требованиям, а также определены необходимые для работоспособности системы запросы к базе данных и их назначения.

# **РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ**

## **Разработка кода информационной системы**

Разработка информационной системы включает в себя все работы по созданию информационного программного обеспечения и его компонентов в соответствии с заданными требованиями [12].

В процессе разработки ИС можно выделить следующие этапы [13]:

* изучение объекта;
* выполнение необходимых научно-исследовательских работ (НИР);
* разработка вариантов концепции информационной системы;
* выбор вариантов концепции, который удовлетворяет требования;
* оформление отчета о проделанной работе.

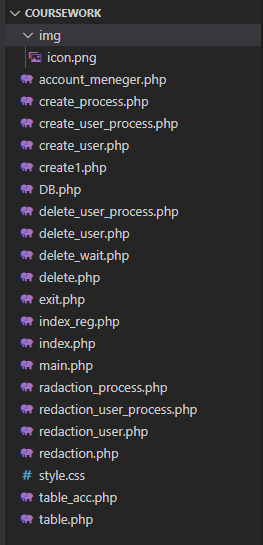
Информационная система курсовой работы была реализована на языке программирования php, с использованием языка стилизации документа css. Для проектирования и управление базой данных информационной системы было использовано приложение mySQL Workbench, а для непосредственной разработки кода информационной системы был использован Visual Code.

**MySQL Workbench** — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL [14].

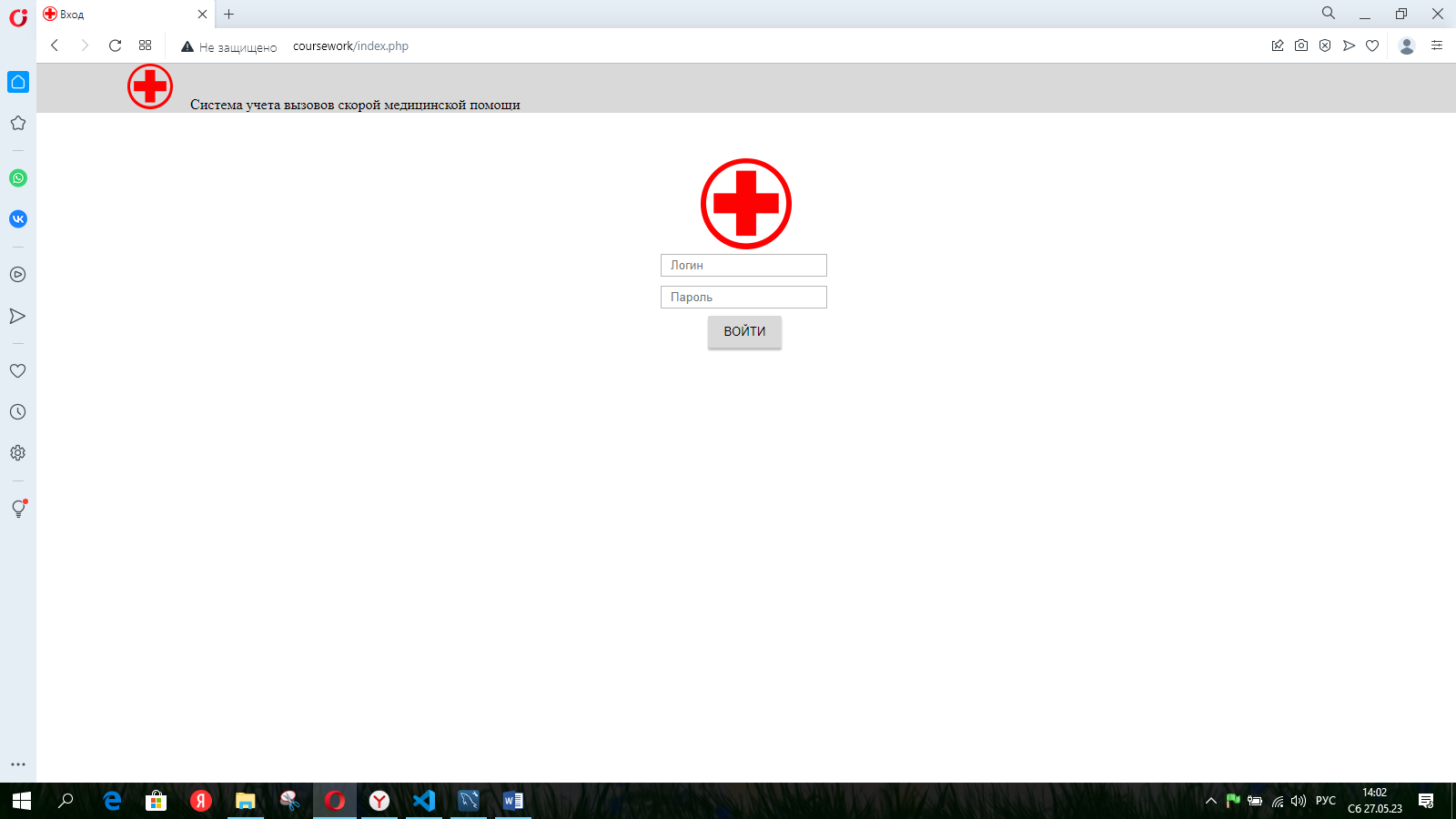
**PHP** (рекурсивный акроним словосочетания PHP: Hypertext Preprocessor) - это распространенный язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML [15].

**CSS** — [формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) декорирования и описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML). Также может применяться к любым XML-документам, например, к SVG или XUL [16].

**Visual Studio Code (VS Code)** — текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как редактор кода для кроссплатформенной разработки [веб](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)- и облачных приложений [17].



*Рис. №2 “Структура информационной системы”*

*Рис. №3 “Страница входа в систему”*

session\_start();

$try\_login = $\_SESSION["try\_login"];

session\_destroy();

require\_once ("DB.php");

echo "<section class='index\_main'>

<div class='index\_main\_img'>

<img src='img/icon.png'>";

if ($try\_login == true) {

echo "</div>

<div class='try\_is\_true\_p'><p>неправильный логин или пароль</p></div>

<div class='index\_main\_form'>

<form action='index\_reg.php' method='post' class='try\_is\_true'>";

}

else{

echo "</div>

<div class='index\_main\_form'>

<form action='index\_reg.php' method='post'>";

}

echo "

<input name='login' required type='text' placeholder='Логин' style='margin-left: 0px'><br>

<input name='password' required type='password' placeholder='Пароль' style='margin-left: 0px'><br>

<div class='index\_main\_sub'>

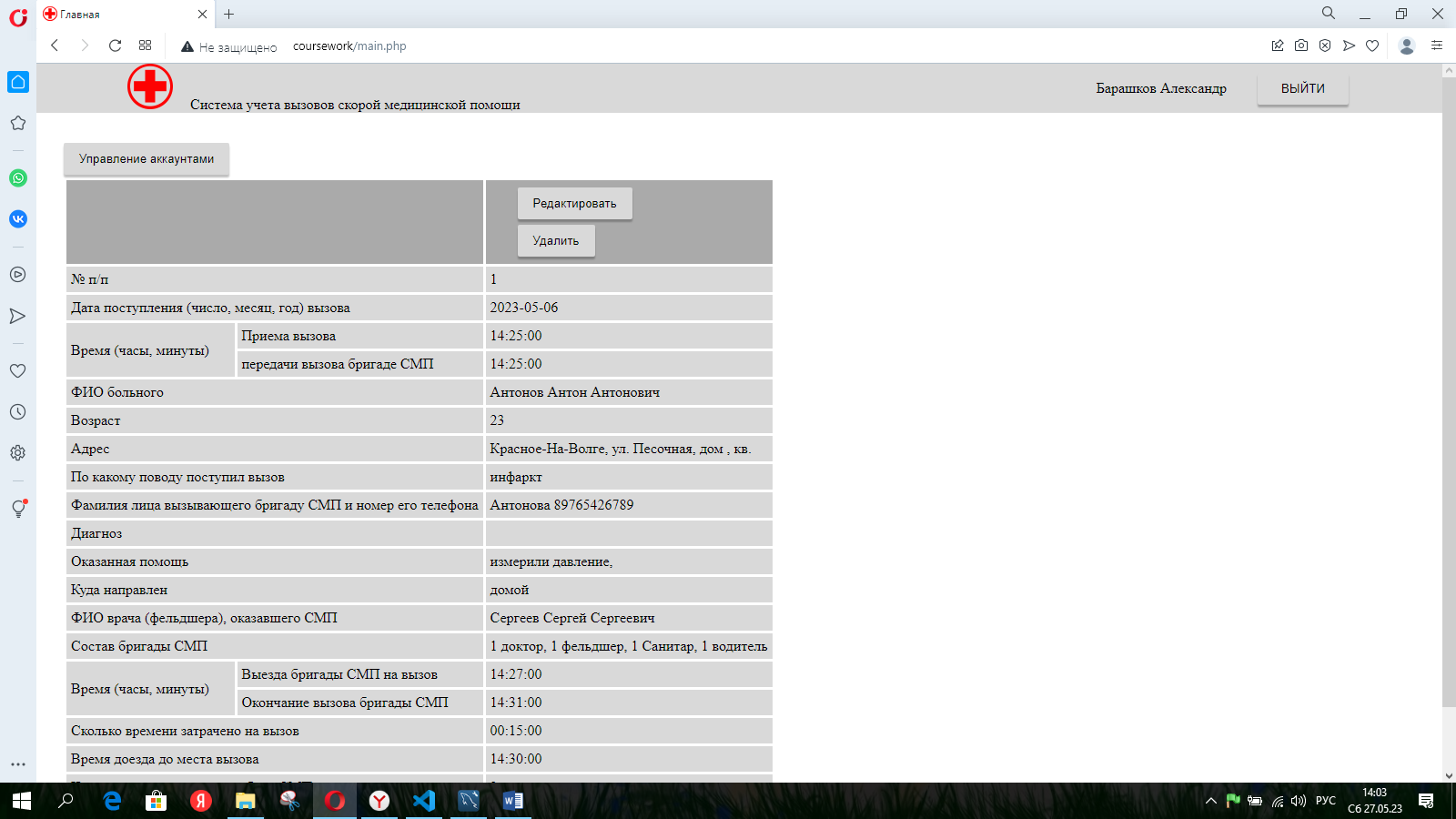
<input type='submit' value='ВОЙТИ' style='margin-left: 0px'>

</div>

</form>

</div>

</section>";

*Рис. №4 “Главная страница”*

<?php

session\_start();

require\_once ("DB.php");

$user\_id = $\_SESSION["id"];

if (!isset($\_SESSION["id"])) {

$new\_url = "http://coursework";

header('location: '.$new\_url);

}

$link = "SELECT \* FROM user where user\_id = '$user\_id';";

$result = mysqli\_query($connect, $link);

$row = $result -> fetch\_assoc();

echo "<section class='shapka'>

<div>

<img src='img/icon.png'> Система учета вызовов скорой медицинской помощи

</div>

<form action='exit.php'>

".$row['user\_fam']. " ".$row['user\_nam']."

<input type='submit' value='ВЫЙТИ'>

</form>

</section>";

if ($\_SESSION["type"] == 0) {

echo "<form action='account\_meneger.php'>

<input type='submit' value='Управление аккаунтами'>

</form>";

}

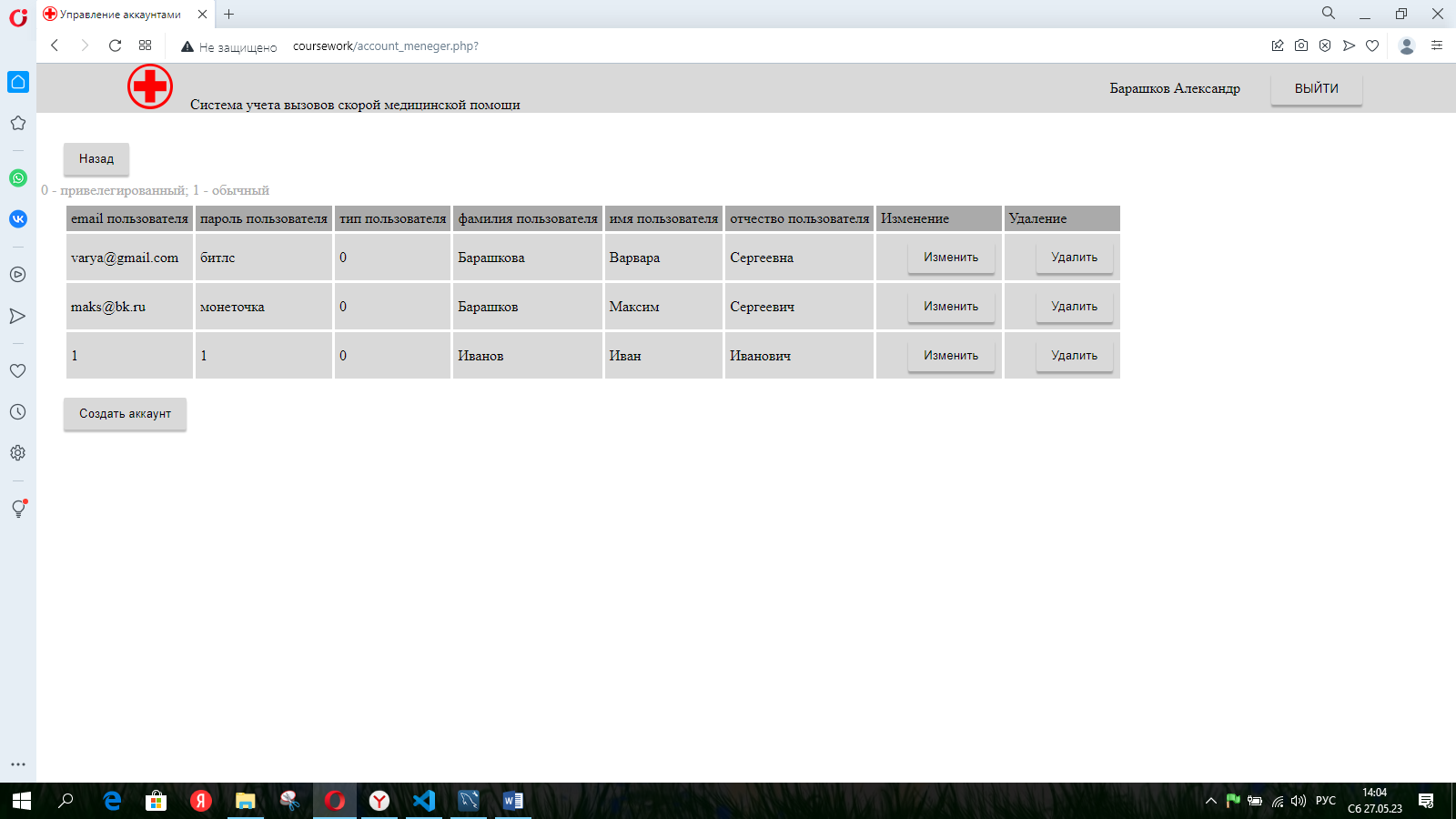
require\_once("table.php");

?>

<form action="create1.php">

<input type="submit" value="Создать новую запись">

</form>

*Рис. №5 “Страница менеджера аккаунтов”*

<?php

session\_start();

require\_once ("DB.php");

$user\_id = $\_SESSION["id"];

if (!isset($\_SESSION["id"])) {

$new\_url = "http://coursework";

header('location: '.$new\_url);

}

$link = "SELECT \* FROM user where user\_id = '$user\_id';";

$result = mysqli\_query($connect, $link);

$row = $result -> fetch\_assoc();

echo "<section class='shapka'>

<div>

<img src='img/icon.png'> Система учета вызовов скорой медицинской помощи

</div>

<form action='exit.php'>

".$row['user\_fam']. " ".$row['user\_nam']."

<input type='submit' value='ВЫЙТИ'>

</form>

</section>";

echo "<form action='main.php'>

<input type='submit' value='Назад'>

</form>";

$user\_info = "SELECT \* FROM user WHERE user\_id <> '$user\_id'";

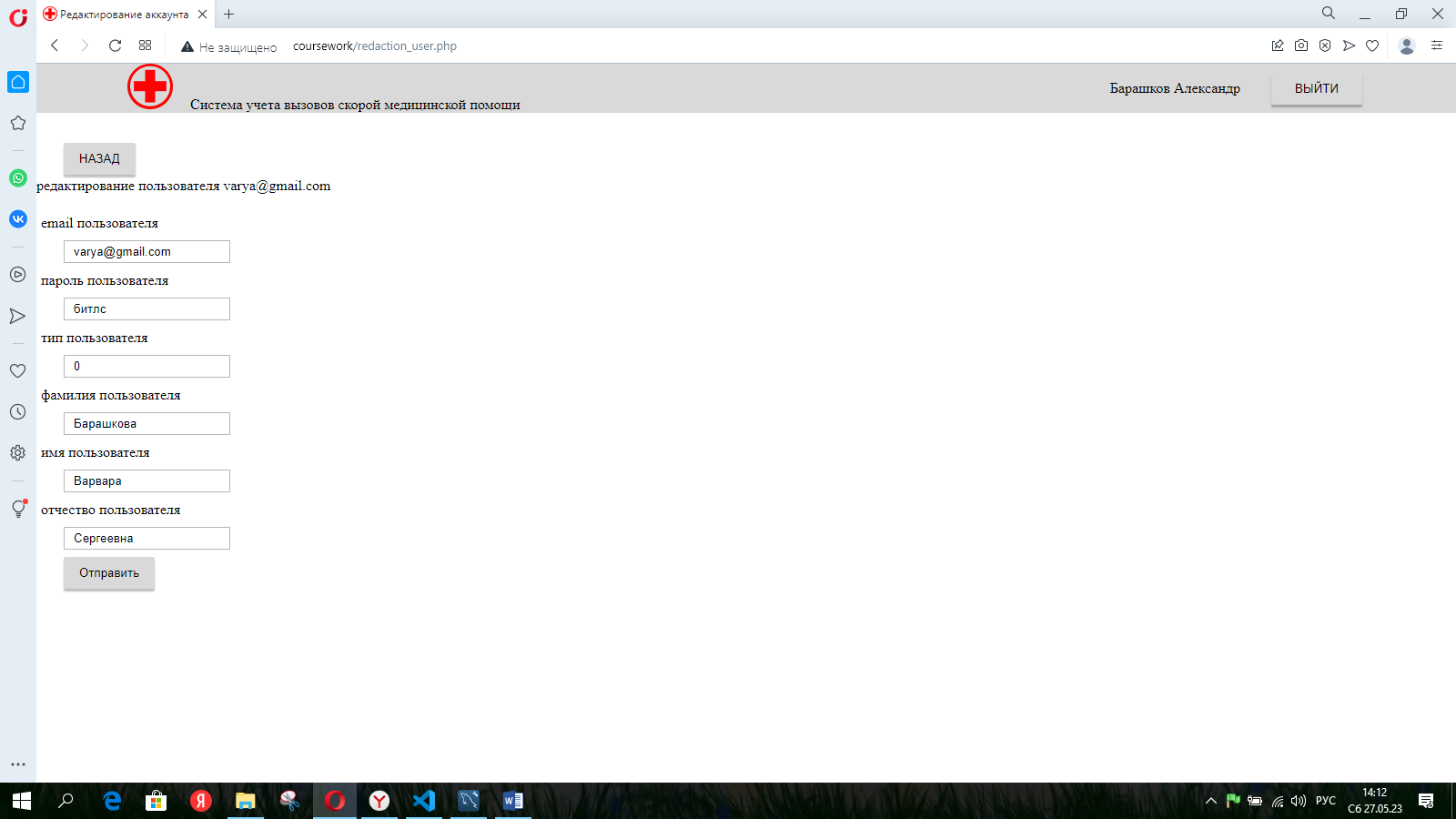
require\_once("table\_acc.php");

echo "<form action='create\_user.php'>

<input type='submit' value='Создать аккаунт'>

</form>";

?>

*Рис. №6 “Страница редактирования аккаунтов”*

session\_start();

$user\_id = $\_SESSION["id"];

if (!isset($\_SESSION["id"])) {

$new\_url = "http://coursework";

header('location: '.$new\_url);

}

require\_once("DB.php");

$user\_email = $\_POST["user\_email"];

$user\_password = $\_POST["user\_password"];

$user\_type = $\_POST["user\_type"];

$user\_fam = $\_POST["user\_fam"];

$user\_nam = $\_POST["user\_nam"];

$user\_otch = $\_POST["user\_otch"];

$id = $\_SESSION["id\_user\_red"];

$link = "UPDATE user SET

user\_email = '$user\_email', user\_password = '$user\_password', user\_type = '$user\_type',

user\_fam = '$user\_fam', user\_nam = '$user\_nam', user\_otch = '$user\_otch'

WHERE (user\_id = '$id');";

$result = mysqli\_query($connect, $link);

if (!$result) {

echo "Ошибка записи данных (" .mysqli\_error($connect). ")<br>";

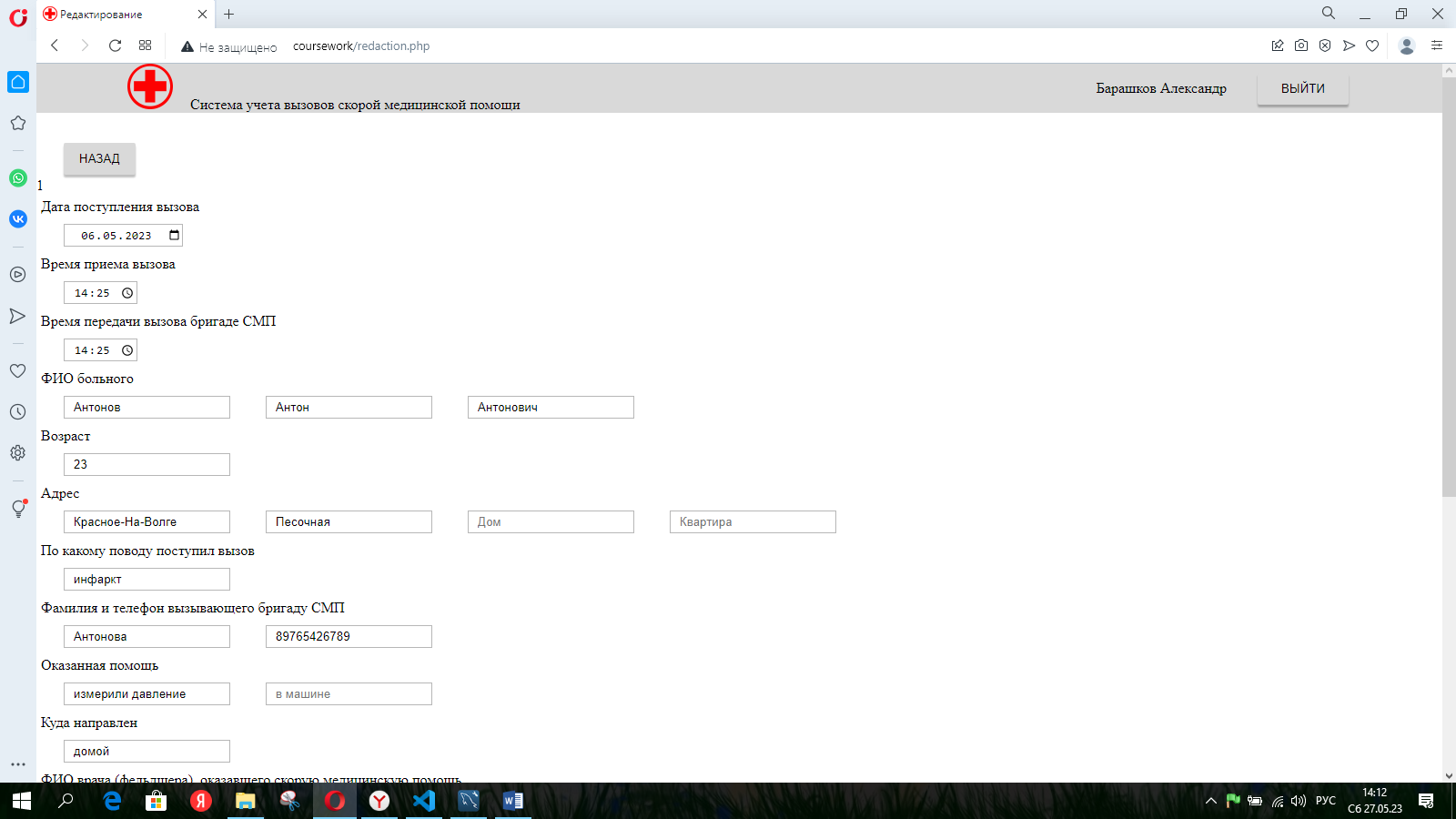
}

else {

$new\_url = "account\_meneger.php";

header('location: '.$new\_url);

}

*Рис. №7 “Страница редактирования записей”*

echo "

<form action='radaction\_process.php' method='post'>

<p>Дата поступления вызова</p>

<input name='call\_date\_incoming' type='date' value='".$row\_call\_smp["call\_date\_incoming"]."'>

<p>Время приема вызова</p>

<input name='call\_time\_receiving' type='time' value='".$row\_call\_smp["call\_time\_receiving"]."'>

<p>Время передачи вызова бригаде СМП</p>

<input name='call\_time\_transfer' type='time' value='".$row\_call\_smp["call\_time\_transfer"]."'>

<p>ФИО больного</p>

<input name='sick\_fam' type='text' placeholder='Фамилия' value='".$row\_sick["sick\_fam"]."'>

<input name='sick\_nam' type='text' placeholder='Имя' value='".$row\_sick["sick\_nam"]."'>

<input name='sick\_otch' type='text' placeholder='Отчество' value='".$row\_sick["sick\_otch"]."'>

<p>Возраст</p>

<input name='sick\_age' type='text' value='".$row\_sick["sick\_age"]."'>

<p>Адрес</p>

<input name='sick\_address\_sity' type='text' placeholder='Населенный пункт' value='".$row\_sick["sick\_address\_sity"]."'>

<input name='sick\_address\_street' type='text' placeholder='Улица' value='".$row\_sick["sick\_address\_street"]."'>

<input name='sick\_address\_house' type='text' placeholder='Дом' value='".$row\_sick["sick\_address\_home"]."'>

<input name='sick\_address\_flat' type='text' placeholder='Квартира' value='".$row\_sick["sick\_address\_flat"]."'>

<p>По какому поводу поступил вызов</p>

<input type='text' name='call\_reason' value='".$row\_call\_smp["call\_reason"]."'>

<p>Фамилия и телефон вызывающего бригаду СМП</p>

<input name='caller\_fam' type='text' placeholder='Фамилия' value='".$row\_caller\_smp["caller\_fam"]."'>

<input name='caller\_phone' type='text' placeholder='Номер телефона' value='".$row\_caller\_smp["caller\_phone"]."'>

<p>Оказанная помощь</p>

<input type='text' name='dep\_help\_in\_site' placeholder='на месте' value='".$row\_departure["dep\_help\_in\_site"]."'>

<input type='text' name='dep\_help\_in\_car' placeholder='в машине' value='".$row\_departure["dep\_help\_in\_car"]."'>

<p>Куда направлен</p>

<input type='text' name='sick\_where\_sent' value='".$row\_sick["sick\_where\_sent"]."'>

<p>ФИО врача (фельдшера), оказавшего скорую медицинскую помощь</p>

<input type='text' name='dep\_doc\_fam' value='".$row\_departure["dep\_doc\_fam"]."'>

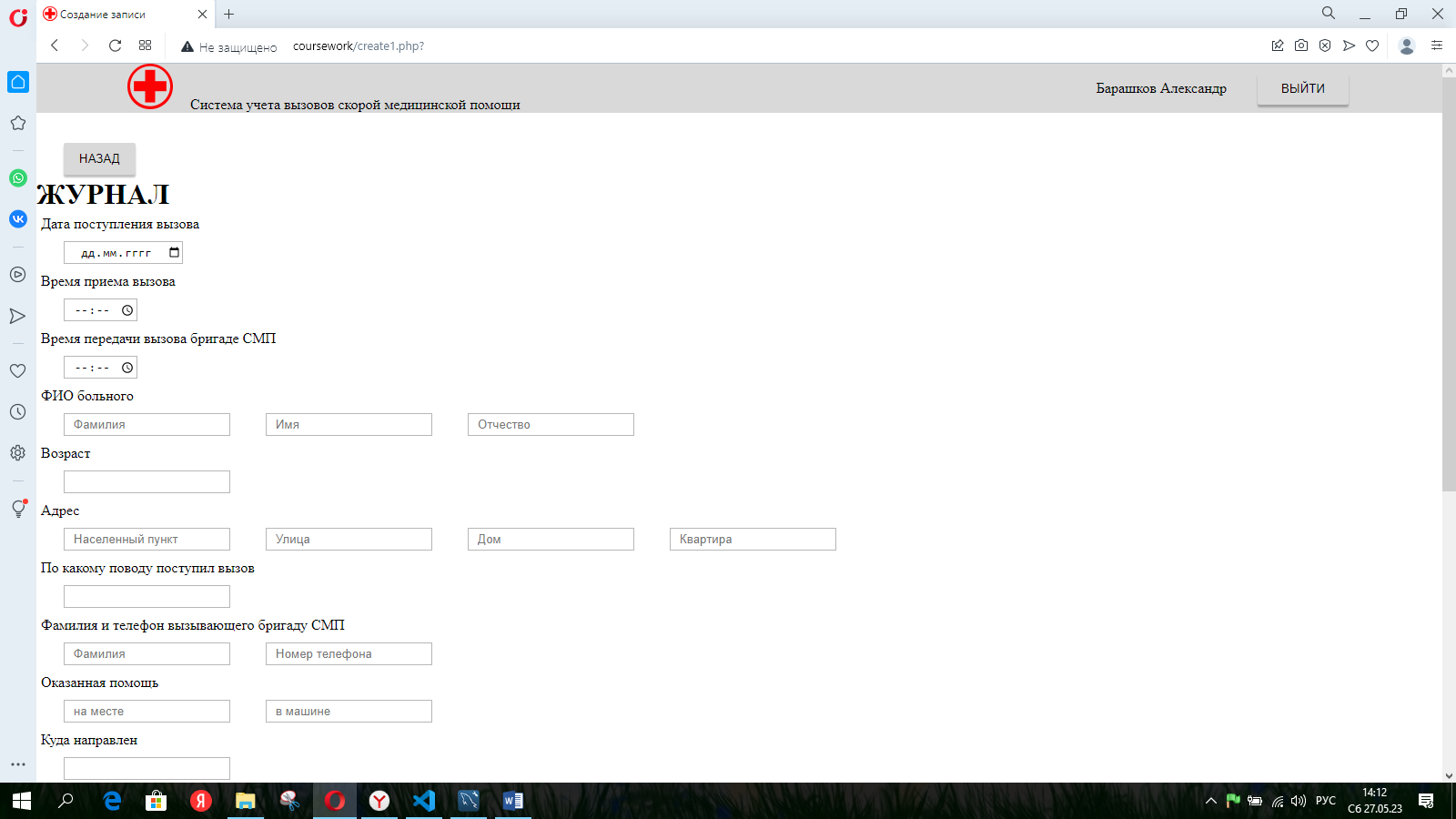
<input type='text' name='dep\_doc\_nam' value='".$row\_departure["dep\_doc\_nam"]."'>

<input type='text' name='dep\_doc\_otch' value='".$row\_departure["dep\_doc\_otch"]."'>

<p>Диагноз</p>

<input type='text' name='sick\_diagnos' value='".$row\_sick["sick\_diagnos"]."'>

<p>Состав бригады СМП</p>";

*Рис. №8 “Страница создание записей”*

<form action='create\_process.php' method='post'>

<p>Дата поступления вызова</p>

<input name='call\_date\_incoming' type='date'>

<p>Время приема вызова</p>

<input name='call\_time\_receiving' type='time'>

<p>Время передачи вызова бригаде СМП</p>

<input name='call\_time\_transfer' type='time'>

<p>ФИО больного</p>

<input name='sick\_fam' type='text' placeholder='Фамилия'>

<input name='sick\_nam' type='text' placeholder='Имя'>

<input name='sick\_otch' type='text' placeholder='Отчество'>

<p>Возраст</p>

<input name='sick\_age' type='text'>

<p>Адрес</p>

<input name='sick\_address\_sity' type='text' placeholder='Населенный пункт'>

<input name='sick\_address\_street' type='text' placeholder='Улица'>

<input name='sick\_address\_house' type='text' placeholder='Дом'>

<input name='sick\_address\_flat' type='text' placeholder='Квартира'>

<p>По какому поводу поступил вызов</p>

<input type='text' name='call\_reason' >

<p>Фамилия и телефон вызывающего бригаду СМП</p>

<input name='caller\_fam' type='text' placeholder='Фамилия'>

<input name='caller\_phone' type='text' placeholder='Номер телефона'>

<p>Оказанная помощь</p>

<input type='text' name='dep\_help\_in\_site' placeholder='на месте'></input>

<input type='text' name='dep\_help\_in\_car' placeholder='в машине'></input>

<p>Куда направлен</p>

<input type='text' name='sick\_where\_sent'>

<p>ФИО врача (фельдшера), оказавшего скорую медицинскую помощь</p>

<input type='text' name='dep\_doc\_fam' >

<input type='text' name='dep\_doc\_nam' >

<input type='text' name='dep\_doc\_otch' >

<p>Диагноз</p>

<input type='text' name='sick\_diagnos'>

<p>Состав бригады СМП</p>

<input type='checkbox' name='doctor' value='1'> Доктор

<input type='radio' name='paramedic' value='1'> 1 фельдшер

<input type='radio' name='paramedic' value='2'> 2 фельдшера

<input type='checkbox' name='attendant' value='1'> Санитар

<input type='checkbox' name='driver' value='1'> Водитель

Таким образом, на этапе разработки информационной системы учета вызовов скорой медицинской помощи был написан программный код системы. Был реализован функционал, соответствующий требованиям к системе. Для проверки корректности выполнения поставленных перед системой задач, необходимо провести тестирование.

# **ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

## **План тестирования информационной системы**

**Тестирование программного обеспечения** — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и ее ожидаемым поведением на конечном наборе тестов, выбранных определенным образом [18].

Этапы тестирования [19]:

1. Проверка требований к программному продукту на полноту.
2. Определение методов тестирования.
3. Разработка стратегии тестирования.
4. Разработка плана тестирования.
5. Создание наборов тестов.
6. Создание отчета о тестировании.

**План тестирования (тест план)** - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения [20].

Таблица №5

План тестирования

| **Вопрос** | **Ответ** |
| --- | --- |
| Основная информация | Таблица описывает инструменты, методы и подходы тестирования которые будут использоваться во время тестирования. |
| Цель | Целью составление тест-плана является описание:   * стратегии и методов тестирования; * необходимые ресурсы тестирования; * рисков тестирования; * результатов тестирования. |
| Область тестирования сайта | Объектами тестирования являются следующие компоненты и функции сайта:   * страница входа; * страница создания записей; * функция удаления записей. |
| План работы | * анализ требований к системе и базе данных; * определение области тестирования; * функциональное тестирование и разработка тест-кейсов; * сравнение результатов тестирования с требованиями; * вывод. |
| Функциональное тестирование | Цель функционального тестирования состоит в проверке соответствия фактической работы системы и составленных требований.  Для функционального тестирования будут разработаны тест-кейсы. Если тест-кейсы будут пройдены системой, то будет считаться что системой пройдено все функциональное тестирование. |
| Инструменты | Для формирования тест-кейсов будет использоваться MS Excel. |
| Список браузеров | Проверка системы будет осуществляться в браузере Opera версии: 99.0.4788.47. |
| Список устройств | Ноутбук с ОС Windows 10. |
| Критерии качества | Система должна работать в соответствии с требованиями, выполнять все основные функции и не иметь критических и серьезных дефектов, мешающих системе выполнять свои основные функции. |
| Риски процесса тестирования | На результаты тестирования могут повлиять:   * проблемы с сервером; * также ввод неполных данных. |
| Результаты | Тест-кейсы |

На данном этапе разработан план тестирования. Это позволит более организованно разработать тесты для системы.

## **Написание тест-кейсов**

**Тест-кейс** – это набор условий, действий и ожидаемых результатов, направленных на проверку какого-либо функционала. Тест-кейс представляет собой описание одной показательной проверки на соответствие требованиям, прямым или косвенным. Тест-кейсы содержат как положительные, так и негативные проверки [21].

Обязательные атрибуты:

1. **Уникальный идентификатор** — некое уникальное значение. Чаще всего применяют простую сквозную нумерацию.
2. **Краткое описание** — лаконичное описание сути тест-кейса. Может содержать ссылку на требование к системе.
3. **Входные данные** — название полей ввода данных и соответствующие полям значения вводимых данных.
4. **Шаги** — полная последовательность действий. Ее выполняют, чтобы провести описываемую тест-кейсом проверку.
5. **Ожидаемый результат** — описание планируемого поведения или результата.
6. **Фактический результат** — описание итогового поведения или результата системы.
7. **Статус** — текущее состояние тест-кейса.

Ниже в виде таблице приведен один из тест-кейсов к информационной системе.

Таблица №6

Тест-кейс

| **id** | **Краткое описание** | **Шаги** | **Входные данные** | **Ожидаемый результат** | **Фактический результат** | **Статус** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | вход в систему при вводе неверного логина и пароля | запуск системы;  ввод верного логина;  ввода верного пароля;  нажатие кнопки “ВОЙТИ” | Логин = "varya@gmail.com" Пароль = "битлс" | Вход на главную страницу | Вход на главную страницу | пройден успешно |

Если система проходит все тест кейсы успешно, то можно сказать, что она прошла и тестирование.

## **Модульное тестирование**

**Модульное тестирование** - процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки [22].

Этапы создание юнит-тестов

1. разработчик пишет код конкретной функции — юнита;
2. проверка, изолированности функции;
3. если функции нужны реакции от других модулей, разработчик создает заглушки, которые имитируют другие модули и взаимодействие с ними.
4. написание тестов и исправление ошибок;
5. запуск юнит-тестов;
6. просмотр результатов тестирования.

Таким образом для тестирования системы были составлены тест-кейсы, которые система прошла успешно. Однако модульное тестирование проведено не было из чего можно сделать вывод, что система полностью не протестирована.

# **ЗАКЛЮЧЕНИЕ**

В ходе разработки информационной системы были приведены обоснования автоматизации учета вызовов скорой медицинской помощи и реализации системы в виде закрытого web-сайта. Были определены требования самой системе и базе данных.

На этапе разработки кода информационной системы был реализован функционал, соответствующий требованиям к системе, а на этапе реализации базы данных была спроектирована база данных, которая также соответствует перечисленным требованиям.

Был составлен план тестирования, после чего было проведено функциональное тестирование системы. Для тестирования были разработаны тест-кейсы, которые система прошла успешно. Модульное тестирование проведено не было.

В итоге была реализована система учета вызовов скорой медицинской помощи, которая соответствует требованиям и выполняет весь основной функционал.

В заключение добавлю:

В ходе создания системы я научился разрабатывать информационные системы на языке программирования php с подключением и взаимодействием с базой данных. Однако, система, конечно, не идеальна, что значит, что мне есть чему еще научиться и что еще узнать.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Автоматизированная система управления // Википедия URL: https://ru.wikipedia.org/wiki/Автоматизированная\_система\_управления (дата обращения: 12.05.23).
2. ГОСТ 34.321-96 Информационные технологии (ИТ). Система стандартов по базам данных. Эталонная модель управления данными // Кодекс URL: https://docs.cntd.ru/document/1200017662 (дата обращения: 12.05.23).
3. Скорая и скорая специализированная медицинская помощь // Скорая помощь - Уссурийск URL: https://ssmpu.ru/services/scope/acute\_care/ (дата обращения: 12.05.23).
4. URL: https://ssmpkms.ru/informatsiya-dlya-patsientov/93-о-скорой-медицинской-помощи.html?showall=1&limitstart=/(дата обращения: 10.05.23).
5. БСБД: 10 Лекция: Модель сущность-связь (ER-модель). Проектирование баз данных. // MOODLE - Виртуальная среда обучения КНИТУ (КХТИ) URL: https://moodle.kstu.ru/mod/page/view.php?id=10147&lang=ja (дата обращения: 12.05.23).
6. Требования к программному обеспечению // Википедия URL: https://ru.wikipedia.org/wiki/Требования\_к\_программному\_обеспечению (дата обращения: 12.05.23).
7. Анализ требований // Википедия URL: https://ru.wikipedia.org/wiki/Анализ\_требований#:~:text=Нефункциональные%20требования%20—%20требования%2C%20о% (дата обращения: 12.05.23).
8. ГОСТ 15.016-2016 Система разработки и постановки продукции на производство (СРПП). Техническое задание. Требования к содержанию и оформлению // Кодекс URL: https://docs.cntd.ru/document/1200144624 (дата обращения: 14.05.23).
9. ГОСТ 34.602-2020 Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы // URL: https://www.swrit.ru/doc/gost34/34.602-2020.pdf?ysclid=lbbts8qxkl958486630 (дата обращения: 14.05.23).
10. ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению // URL: https://www.swrit.ru/doc/espd/19.201-78.pdf?ysclid=lbbtyu6950545654749 (дата обращения: 14.05.23).
11. CRUD // Википедия URL: https://ru.wikipedia.org/wiki/CRUD (дата обращения: 24.05.23).
12. Лекция 2\_Жизненный цикл ИС - лекция - Жизненный цикл информационных систем // Сайт для учащихся URL: https://topuch.com/jiznennij-cikl-informacionnih-sistem-v3/index.html#:~:text=Разработка%20информационной%20системы%20включает%20в,в%20соответствии%20с%20заданными%20требованиями (дата обращения: 27.05.23).
13. 1.4 Этапы разработки ис // studfile URL: https://studfile.net/preview/5759694/page:6/ (дата обращения: 28.05.23).
14. MySQL Workbench // Википедия URL: https://ru.wikipedia.org/wiki/MySQL\_Workbench (дата обращения: 27.05.23).
15. Что такое PHP? // PHP: Hypertext Preprocessor URL: https://www.php.net/manual/ru/intro-whatis.php (дата обращения: 28.05.23).
16. CSS // Википедия URL: https://ru.wikipedia.org/wiki/CSS (дата обращения: 27.05.23).
17. Visual Studio Code // Википедия URL: https://ru.wikipedia.org/wiki/Visual\_Studio\_Code (дата обращения: 27.05.23).
18. Тестирование программного обеспечения // Википедия URL: https://ru.wikipedia.org/wiki/Тестирование\_программного\_обеспечения (дата обращения: 27.05.23).
19. 25 (14). Тестирование информационной системы. Виды тестирования ис. Разработка и выполнение тестов. // studfile URL: https://studfile.net/preview/9862725/page:10/ (дата обращения: 27.05.23).
20. testirovanie24 // Примеры тест планов по тестированию URL: https://testirovanie24.ru/109405-2/?ysclid=lji5o8rgb5868000192 (дата обращения: 30.06.23).
21. Тестовая документация: что, где, когда // hawkingbros URL: https://hawkingbros.com/article/testovaya-dokumentaciya-cto-gde-kogda?ysclid=lji5kzq93d838756951 (дата обращения: 30.06.23).
22. ОТЧЕТ. Практическая работа 1 Что такое тестирование программного обеспечения // topuch URL: https://topuch.com/prakticheskaya-rabota-1-chto-takoe-testirovanie-programmnogo-o/index.html (дата обращения: 30.06.23).

# **СПИСОК ИЛЛЮСТРАЦИЙ**

1. [*ER-диаграмма базы данных*](#_heading=h.1ci93xb) - стр.12
2. [*Структура информационной системы*](#_heading=h.3whwml4) - стр.18
3. [*Страница входа в систему*](#_heading=h.3whwml4) - стр.18
4. [*Главная страница*](#_heading=h.3whwml4) - стр.19
5. [*Страница менеджера аккаунтов*](#_heading=h.3whwml4) - стр.20
6. [*Страница редактирования аккаунтов*](#_heading=h.3whwml4) - стр.22
7. [*Страница редактирования записей*](#_heading=h.3whwml4) - стр.23
8. [*Страница создание записей*](#_heading=h.3whwml4) - стр.24

# **СПИСОК ТАБЛИЦ**

1. [*Краткие сведения об информационной системе*](#_heading=h.1y810tw) - стр.7
2. [*Бриф на разработку информационной системы*](#_heading=h.4i7ojhp) - стр.8
3. [*Требования к базе данных*](#_heading=h.2xcytpi) - стр.11
4. [*Словарь базы данных*](#_heading=h.2xcytpi) - стр.12
5. [*План тестирования*](#_heading=h.2bn6wsx) - стр.26
6. [*Тест-кейс*](#_heading=h.1fob9te) – стр.29

# **ПРИЛОЖЕНИЯ**

1. [*Техническое задание*](https://docs.google.com/document/d/1g7O-5AGw9HcWy6PqjOQ3VI3gem-duosz/edit?usp=drive_link&ouid=108984478239207438867&rtpof=true&sd=true)
2. [*ER-диаграмма (mwb-формат)*](https://drive.google.com/file/d/1mRxHh-jW8Bs2S7xStM4Ky6EkAVWcdJ7a/view?usp=drive_link)
3. [*ER-диаграмма (png-формат)*](https://drive.google.com/file/d/16zb3ghJAx-xsZHzEOmhB-Lk036Cm9oti/view?usp=drive_link)
4. [*Словарь базы данных*](https://docs.google.com/document/d/1k4obJGCTdNrS6fb4IrbIu9mezlvCrRil/edit?usp=drive_link&ouid=108984478239207438867&rtpof=true&sd=true)
5. [*Видео-демонстрация функционала*](https://drive.google.com/file/d/1LHQlJOf58rLVkKppDJJSHpTnWpnF8C_k/view?usp=drive_link)
6. [*Код информационной системы*](https://drive.google.com/drive/folders/1Rl5lsxkgUg8wo5k8H36w-CQlEsqDTb4M?usp=drive_link)
7. [*Тест-кейсы*](https://docs.google.com/spreadsheets/d/1UjuObWHVeFOfoQmLhDY29bpf2qiebDiI/edit?usp=drive_link&ouid=108984478239207438867&rtpof=true&sd=true)