**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

КП.09.02.07-5.23.201.10 ПЗ

ИНФОРМАЦИОННАЯ СИСТЕМА

«ПАСПОРТНЫЙ СТОЛ»

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.П. Морозов) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.В. Макрицкий) |
|  |  |  |

Иркутск 2023

**Содержание**

[Введение 3](#_Toc150974664)

[1 Описание предметной области ИС 4](#_Toc150974665)

[2 Анализ инструментов средств разработки 5](#_Toc150974666)

[3 Техническое задание 12](#_Toc150974667)

[4 Проектирование ИС 13](#_Toc150974668)

[4.1 Структурная схема ИС 13](#_Toc150974669)

[4.2 Функциональная схема ИС 15](#_Toc150974670)

[4.3 Проектирование базы данных 17](#_Toc150974671)

[4.4. Проектирование интерфейса 21](#_Toc150974672)

[5 Разработка ИС 23](#_Toc150974673)

[5.1. Разработка интерфейса ИС 23](#_Toc150974674)

[5.2 Разработка базы данных ИС 27](#_Toc150974675)

[5.3 Разработка ИС 27](#_Toc150974676)

[5.4 Тестирование ИС 29](#_Toc150974677)

[6 Технологическая документация ИС 31](#_Toc150974678)

[6.1 Руководство пользователя ИС 31](#_Toc150974679)

[Заключение 34](#_Toc150974680)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 35](#_Toc150974681)

[Приложение А – Техническое задание 36](#_Toc150974682)

[Приложение Б – Листинг roles.php 41](#_Toc150974683)

Введение

В настоящее время эффективное функционирование любой современной организации невозможно без применения информационных систем, так как они имеют ряд существенных отличий от стандартных прикладных программ. Создание и обработка баз данных дает возможность накапливать огромное количество информации, производить ее сортировку и выборку по различным признакам, производить формирование списков и таблиц. Все это позволяет существенно облегчить и ускорить труд людей по выполнению данных операций.

С развитием технологий и возрастанием ожиданий граждан, паспортные столы сталкиваются с необходимостью улучшить качество своих услуг. Современные информационные технологии имеют огромный потенциал для оптимизации работы паспортного стола. Автоматизация процесса оформления и выдачи паспортных документов, внедрение электронных систем записи на прием и электронного оформления документов - все это может существенно улучшить эффективность и доступность услуг паспортного стола.

Актуальностью данной работы является необходимость автоматизация деятельности паспортного стола, учитывая, что в данной области существует необходимость быстрого получения и обработки информации.

Цель курсовой работы – разработка информационной системы «Паспортный стол».

Задачи курсовой работы:

* Провести предпроектное исследование предметной области (выбрать тему, поставить цель и задачи, описать предметную область).
* Разработать техническое задание.
* Спроектировать программный продукт.
* Разработать и отладить программный продукт.
* Составить программную документацию в виде руководства пользователя.

## **Описание предметной области ИС**

Предметной областью курсовой работы является паспортный стол.

Паспортный стол – это учреждение, ответственное за оформление и выдачу паспортов гражданам. Эта предметная область охватывает различные аспекты, связанные с документами, удостоверяющими личность граждан.

Предметная информационная система "Паспортный стол" является комплексной системой, разработанной для автоматизации работы паспортного стола. Она включает в себя несколько модулей, обеспечивающих эффективное функционирование всех процессов, связанных с оформлением и выдачей паспортных документов.

Тема данного курсового проекта актуальна, поскольку паспорт является важным неотъемлемым документом, удостоверяющим личность каждого гражданина своей страны. Также практически каждый из нас сталкивается со сферой услуг, такими, как оформление частной собственности, приватизация, оформление заграничных паспортов и т.д. Специфика таких организаций связана с большим разнообразием индивидуальных интересов большого числа клиентов. В связи с этим на рабочих местах приходится вести множество записей, заявок, журналов содержащих запросы, выписывать квитанции.

Зачастую в паспортных столах работа с гражданами осуществляется в ручном режиме, что приводит к потере времени и ненадлежащему контролю.

Выбранная тема исследования является актуальной в связи с внедрением автоматизированной системы в паспортный стол, что значительно упростит и ускорит процесс обслуживания граждан.

Информация, хранимая в базе данных информационной системы паспортного стола: информация о гражданах, сведения об обращениях.

## **Анализ инструментов средств разработки**

Грамотно выбранные инструменты для создания программного продукта, заранее определяют его результат.

Проектировать структуру информационной системы удобно через MySQL Workbench и Draw.io, а дизайн – через сервис Figma. Для работников информационная система будет серверной. Для реализации клиентской части подойдут следующие инструменты: HTML5, CSS3 и JS, включая AJAX.

MySQL Workbench — это унифицированный визуальный инструмент для архитекторов баз данных, разработчиков и администраторов баз данных. MySQL Workbench предоставляет моделирование данных, разработку SQL и комплексные инструменты администрирования для настройки сервера, администрирования пользователей, резервного копирования и другое.

Draw.io — это бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом, разработанное на HTML5 и JavaScript. Его интерфейс можно использовать для создания таких диаграмм, как блок-схемы, каркасы, UML-диаграммы, организационные диаграммы и сетевые диаграммы.

Figma – графический редактор для создания прототипов сайтов и приложений. Над проектом одновременно могут работать несколько человек, так как можно выдать доступ на редактирование или комментирование любому. В Figma обычно создают прототипы сайтов и приложений, иллюстрации, векторную графику, рисуют элементы интерфейса. Ещё здесь создают макеты сайтов для тильды: есть возможность импортировать дизайн.

HTML – язык разметки гипертекста. Язык разметки дает браузеру необходимые инструкции о том, как отображать тексты и другие элементы страницы на мониторе. Язык HTML интерпретируется браузерами и отображается в виде документа, в удобной для человека форме.

CSS – каскадные таблицы стилей, которые используются для определения стилей (правил) оформления документов — включая дизайн, вёрстку и вариации макета для различных устройств и размеров экрана.

JavaScript – это мультипарадигменный язык программирования, который обычно применяется в качестве встраиваемого инструмента для программного доступа к различным объектам приложений. С точки зрения веб-разработки, без знаний этой технологии невозможно заниматься созданием современных интерактивных сайтов. Язык JS – это то, что «оживляет» разметку страниц (HTML) и пользовательский функционал (CMS) сайтов. С помощью этого языка реализуется возможность реакции страницы или отдельных ее элементов на действия посетителя.

AJAX, или, более длинно, Asynchronous Javascript And Xml - технология для взаимодействия с сервером без перезагрузки страниц. За счет этого уменьшается время отклика и веб-приложение по интерактивности больше напоминает десктоп.

Информационная система будет содержать в себе информацию – её необходимо хранить, изменять, структурировать и использовать. Это реализуется благодаря базе данных. Были рассмотрены следующие варианты реализации СУБД: MySQL, MySQL Workbench, PostgreSQL.

MySQL — одна из наиболее используемых систем управления базами данных. MySQL управляет реляционными базами данных, то есть такими, в которых таблицы связаны между собой. MySQL работает по принципу клиент-сервер. Компьютер пользователя (клиент) отправляет запрос. Сервер баз данных его обрабатывает и предоставляет ответ. Именно поэтому часто можно услышать понятие MySQL-сервер. Это сервер, на котором хранится база данных. Система MySQL написана на языках программирования C и C++. Для работы MySQL используется язык структурированных запросов SQL.

[MySQL Workbench](https://www.mysql.com/products/workbench/) — универсальный инструмент для работы и управления базами данных. С его помощью можно создавать и редактировать таблицы и другие объекты, управлять доступом пользователей и полноценно администрировать БД. Workbench может пригодиться разработчикам, администраторам БД и архитекторам систем.

Несмотря на название, он может работать не только с MySQL, но и с другими СУБД: Oracle Database, Microsoft SQL Server, PostgreSQL и другими. Одна из главных особенностей Workbench — визуальное моделирование. Оно позволяет увидеть все таблицы и связи между ними в наглядном виде. Это намного упрощает проектирование базы данных.

PostgreSQL — это объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых [СУБД](https://blog.skillfactory.ru/glossary/subd/) в мире. Имеет открытый исходный код и является альтернативой коммерческим базам данных. СУБД позволяет гибко управлять базами данных. С ее помощью можно создавать, модифицировать или удалять записи, отправлять транзакцию — набор из нескольких последовательных запросов на особом языке запросов SQL.

Для наглядности сравнения вариантов реализации базы данных была составлена таблица 1.

Таблица 1 – Сравнение средств реализации базы данных

|  |  |  |
| --- | --- | --- |
| Название СУБД | MySQL | PostgreSQL |
| Большое кол-во типов данных | + | + |
| Популярность | + | - |
| Отказоустойчивость | - | + |
| Не требует удаленного сервера | - | - |
| Простота использования | - | - |
| Портативность | - | - |

Таким образом, в качестве базы данных для будущего продукта была выбрана MySQL, так как она предоставляет весь необходимый функционал для разработки продукта, и при этом она проста в изучении и овладении.

PHP — это распространённый язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML.

Язык PHP обладает рядом неоспоримых преимуществ:

* Высокая скорость работы и, соответственно, общая производительность ресурсов.
* Бюджетность, экономичность. Найти специалиста не представляется проблемой, стоимость написания программ на php не высока.
* Простота освоения, простой синтаксис.
* Отличная совместимость и переносимость — php-коды работают одинаково хорошо с разными платформами.
* Набор текста кода и его редактирование можно осуществлять в любом текстовом или html-редакторе.

Таким образом, php будет более лучшим вариантом, ведь он более компактный и простой в освоении язык. Он вобрал все лучшие особенности таких популярных языков, как C, Java и Perl, а также его сообщество более активное.

Для разработки программного продукта рассмотрены следующие инструментальные средства разработки программных продуктов:

1. PHPStorm.

2. Netbeans.

3. Atom.

PhpStorm — коммерческая кросс-платформенная [интегрированная среда разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) обеспечивает первоклассную поддержку PHP 5.3–8.1, мгновенно предотвращает ошибки, предоставляет точное автодополнение и безопасные рефакторинги, а также возможность редактирования кода на HTML, CSS и JavaScript.

IDE обеспечивает умное автодополнение, подсветку синтаксиса, расширенные настройки форматирования, проверку ошибок на лету, сворачивание фрагментов кода и много других возможностей для продуктивной разработки. Автоматические рефакторинги позаботятся о порядке в вашем коде и помогут легко и безопасно выполнять глобальные настройки проекта.

Сотни встроенных инспекций проверяют код прямо в режиме редактирования, а также позволяют проанализировать весь проект на наличие ошибок и проблем в структуре кода. Для большинства инспекций доступны быстрые исправления, которые позволяют откорректировать код мгновенно. Нажмите Alt+Enter, чтобы просмотреть список возможных исправлений для любой инспекции. PhpStorm помогает быстро перемещаться по коду и экономить время при работе над большими проектами. В один клик можно перейти к определению любого метода, функции или переменной, или найти их использования в проекте.

NetBeans IDE — свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и ряда других. Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведётся независимым сообществом разработчиков-энтузиастов (NetBeans Community) и компанией NetBeans Org. Последние версии NetBeans IDE поддерживают рефакторинг, профилирование, выделение синтаксических конструкций цветом, автодополнение набираемых конструкций на лету и множество предопределённых шаблонов кода. Для разработки программ в среде NetBeans и для успешной инсталляции и работы самой среды NetBeans должен быть предварительно установлен Sun JDK или J2EE SDK подходящей версии. Среда разработки NetBeans по-умолчанию поддерживала разработку для платформ J2SE и J2EE. Начиная с версии 6.0 NetBeans поддерживает разработку для мобильных платформ J2ME, C++ (только g++) и PHP без установки дополнительных компонентов.

Atom (в прошлом Atomicity) — бесплатный текстовый редактор с открытым исходным кодом для Linux, macOS, Windows с поддержкой плагинов, написанных на JavaScript, и встраиваемых под управлением Git. Большинство плагинов имеют статус свободного программного обеспечения, разрабатываются и поддерживаются сообществом.

Atom основан на Electron (ранее известный как Atom Shell) — фреймворке кроссплатформенной разработки с использованием Chromium и io.js. Редактор написан на CoffeeScript и LESS. Версия 1.0 была выпущена 25 июня 2015 г.

Сравнение IDE для разработки программного продукта наглядно представлено в таблице 2.

Таблица 2 – Сравнение IDE для разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Название IDE | PHP Storm | NetBeans | Atom |
| Распространяется бесплатно | + | - | + |
| Автоматическое сохранение | + | - | + |
| Подсказки по коду | + | + | + |
| Интеграция с системой контроля версия (GIT) | + | + | - |
| Возможность расширения функционала библиотеками | + | + | + |
| Заточен под PHP-разработку | + | + | + |
| Поддержка CSS/HTML/JS | + | + | + |
| Комфортное использование на слабых ПК | + | - | + |

Таким образом, после рассмотрения вариантов средств разработок, было принято решение использовать PHPStorm. PHPStorm намного функциональнее, чем приведённые выше Atom и NetBeans. В PHPStorm имеется автоматическое сохранение, которое не позволит случайно потерять все наработки, а также он более удобен в работе.

Обоснование выбора

Для создания программного продукта было решено использовать

средства:

1) Для создания структурных схем, контекстной и диаграмм декомпозиции использовались CASE-средства – Draw.io.

2) Для наглядного составления структуры базы данных использовался инструмент для визуального проектирования баз данных, их редактирования и полного администрирования – MySQL Workbench.

3) Для разработки дизайна web-приложения использовался онлайн-сервис для разработки дизайна и прототипа сайта – Figma.

4) На этапе разработки программного продукта использовались инструменты и среды для разработки: PHP, Laravel, HTML5, CSS3, JavaScript и AJAX, а также редактор кода PHPStorm.

Основные преимущества веб-технологий:

1. Распределенность (пользователь может работать с системой из любого  
   места, связанного с WEB-сервером по сети, находясь в любой точке земного шара);
2. Переносимость (Web-клиенты (браузеры) существуют для любых  
   платформ, от настольных компьютеров до сотовых телефонов. Web-сервера используются для большинства платформ, а Web-приложения обычно пишутся на переносимых языках);  
   привычность интерфейса (почти каждый пользователь компьютера хотя бы раз запускал браузер и работал в нем);  
   простота установки и обслуживания (новую версию web-приложения не надо устанавливать на все компьютеры – достаточно установить на сервер).
3. Простота интерфейса - Пользователи не любят гигантских окошек с  
   сотнями полей ввода, а программисты и разработчики интерфейсов  
   почему-то любят. Web как раз не поощряет сложный интерфейс,  
   скорее поощряет простой.
4. Простота программирования интерфейса - Создавать HTML из  
   шаблонов куда проще и удобнее, чем создавать графические  
   приложения визуальными редакторами интерфейсов.
5. Простота программирования в целом - 3х-уровневая архитектура:  
   база данных – логика приложения (сервер) – логика представления  
   (клиент, браузер)

## **Техническое задание**

Техническое задание (далее – ТЗ) — документ или несколько документов, определяющих цель, структуру, свойства и методы какого-либо проекта, и исключающие двусмысленное толкование различными исполнителями.

ТЗ устанавливает основное назначение разрабатываемого объекта, его технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации (конструкторской, технологической, программной и т. д.) и её состав, а также специальные требования.

Для создания технического задания использовался стандарт ГОСТ 19.201-78.

Согласно ГОСТ 19.201-78 техническое задание должно включать следующие разделы:

Введение.

1. Основание для разработки.
2. Назначение разработки.
3. Требования к программе или программному изделию.
   1. Требования к функциональным характеристикам.
   2. Требования к надежности.
   3. Требования к информационной и программной совместимости.
4. Требования к программной документации.
5. Состав и содержание работ по созданию системы.

ТЗ представлено в Приложении А.

## **Проектирование ИС**

### **4.1 Структурная схема ИС**

Проектирование информационной системы началось с построения диаграммы вариантов использования. На рисунке 1 представлена диаграмма прецедентов Uses CASE. Она содержит 2 актёра, которые могут выполнять суммарно 6 функций.

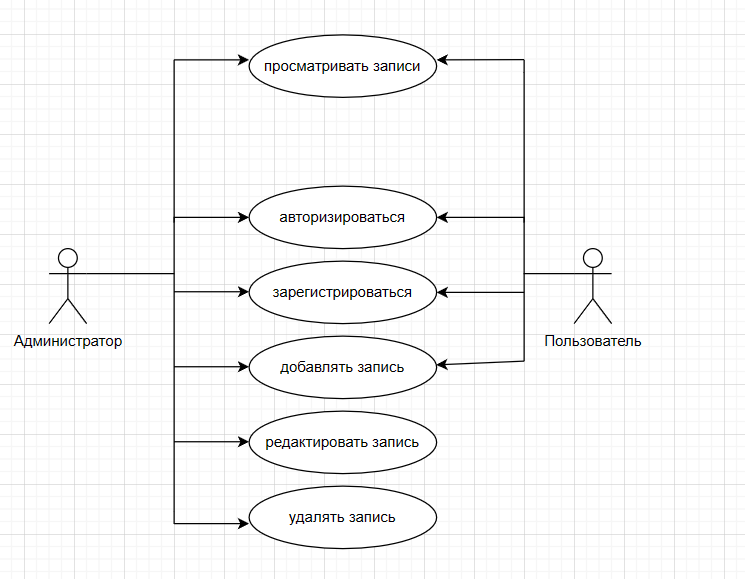


Рисунок 1 - Диаграмма прецедентов

На рисунке 2 представлена диаграмма деятельности. Она содержит 3 роли. Процесс начинается с пользователя, который добавляет запись. Далее администратор проверяет правильность записи, если запись возможна, то данные уходят к работнику, если нет, то посетитель записывается повторно.

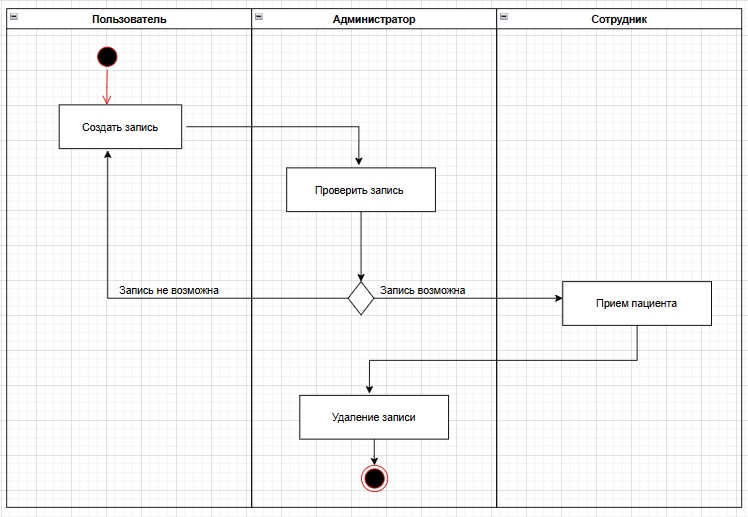


Рисунок 2 – Диаграмма деятельности

На рисунке 3 представлена диаграмма компонентов. В общей сложности 7 элементов.

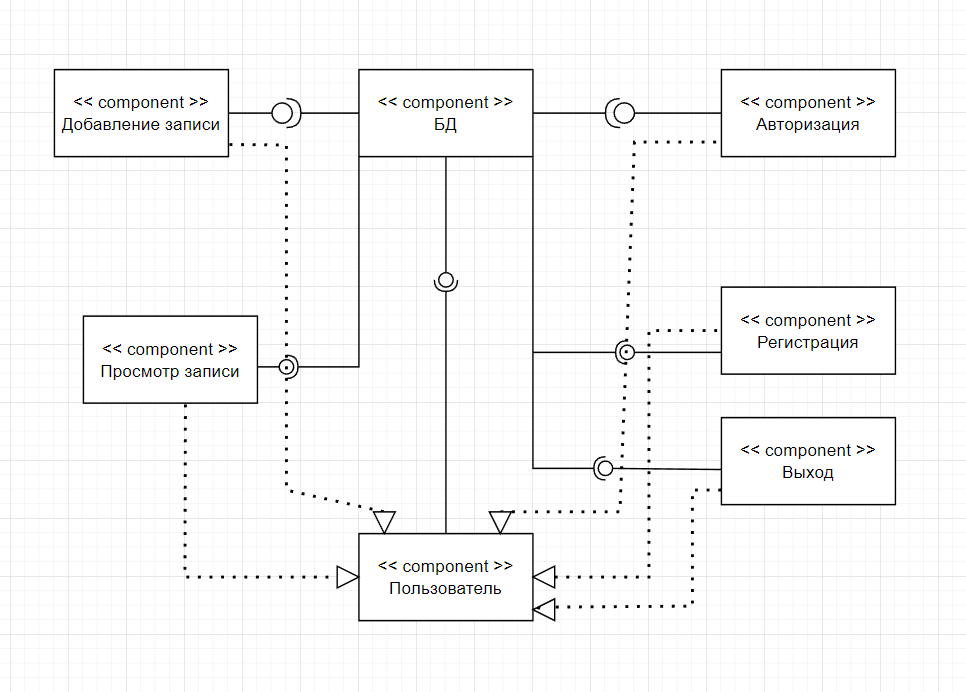


Рисунок 3 – Диаграмма компонентов

На рисунке 4 представлена диаграмма развёртывания. Она показывает, что, чтобы пользоваться программным продуктом, необходим web-сервер, на котором размещаются ИС и БД.

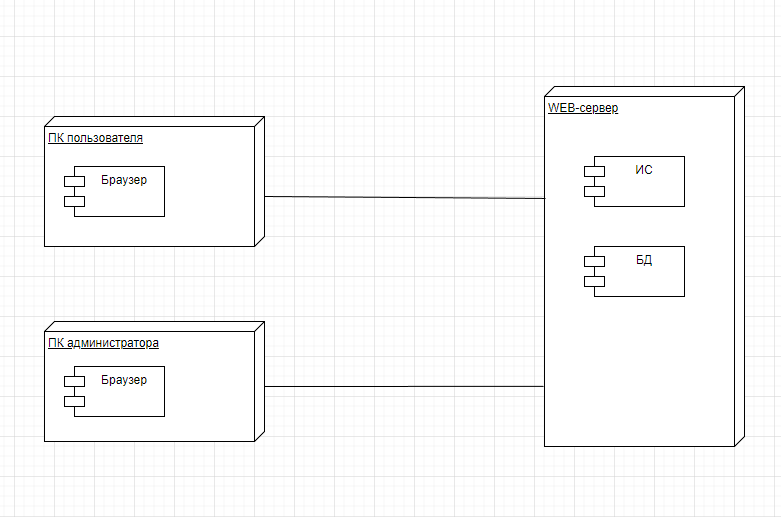


Рисунок 4 – Диаграмма развёртывания

### **4.2 Функциональная схема ИС**

На рисунке 5 представлена контекстная диаграмма, отображающая деятельность ИС.

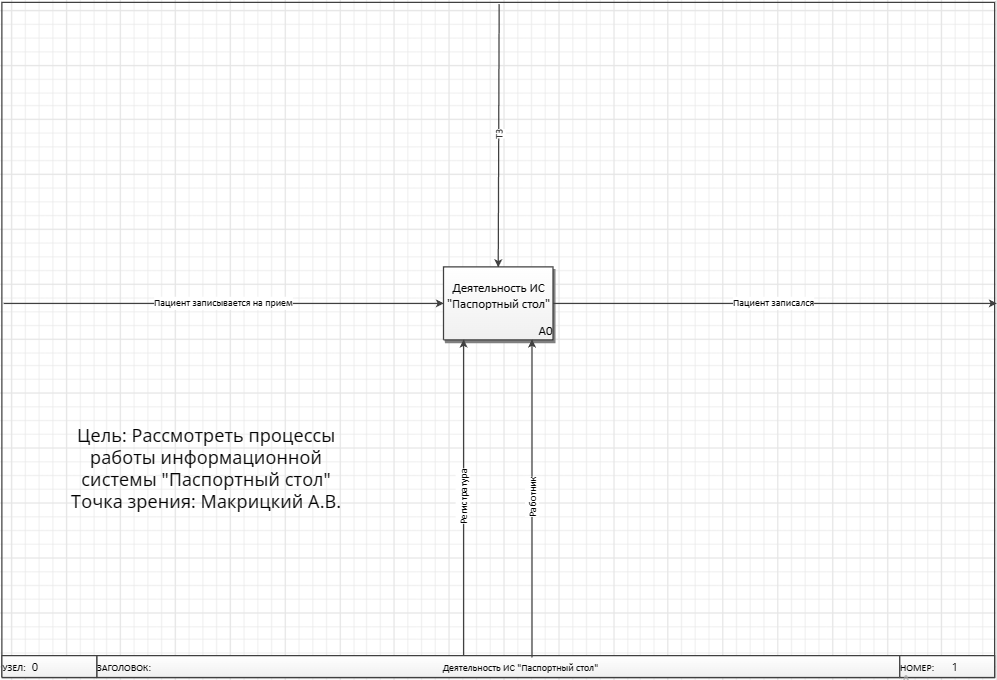


Рисунок 5 – Контекстная диаграмма IDEF0

На рисунке 6 представлена диаграмма декомпозиций (А1), отображающая деятельность ИС более подробно предыдущей.

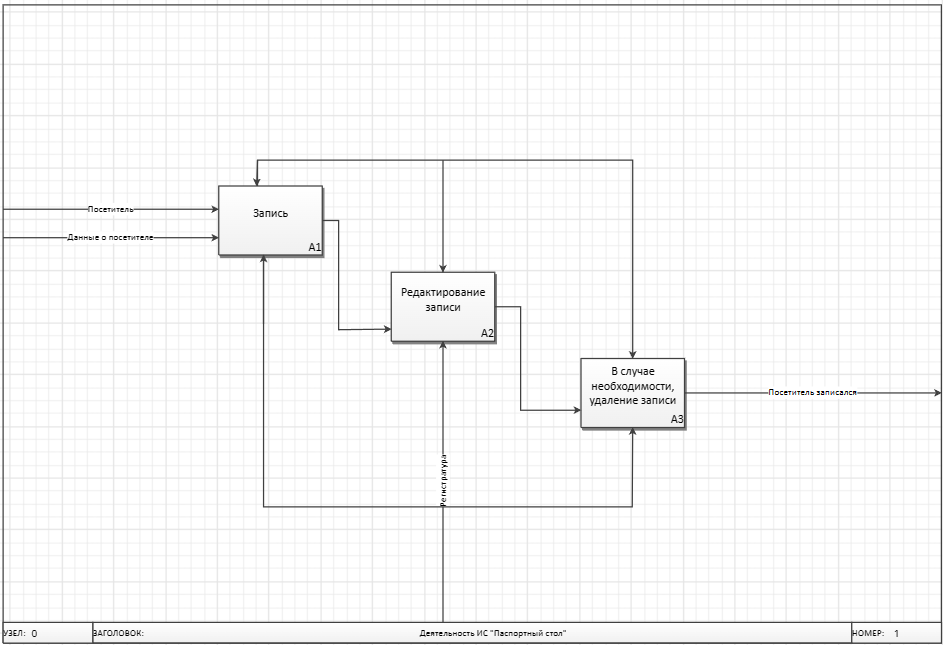


Рисунок 6 – Диаграмма декомпозиций А1

На рисунке 7 представлена диаграмма классов. Она содержит классы, среди них база данных, html-страница, информационная сеть «Паспортный стол», пользователь, сотрудник, администратор, запись.

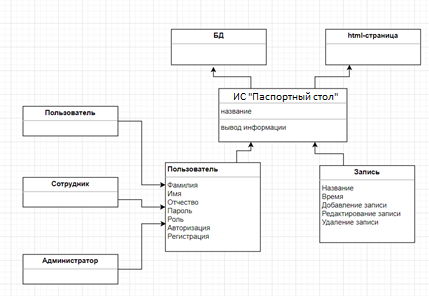


Рисунок 7 – Диаграмма классов

На рисунке 8 представлена диаграмма потоков данных. В центре всего пользователь, который может регистрироваться, просматривать услуги, добавлять, редактировать и удалять записи. Для соответствующих действий в ИС предусмотрены соответствующие БД.

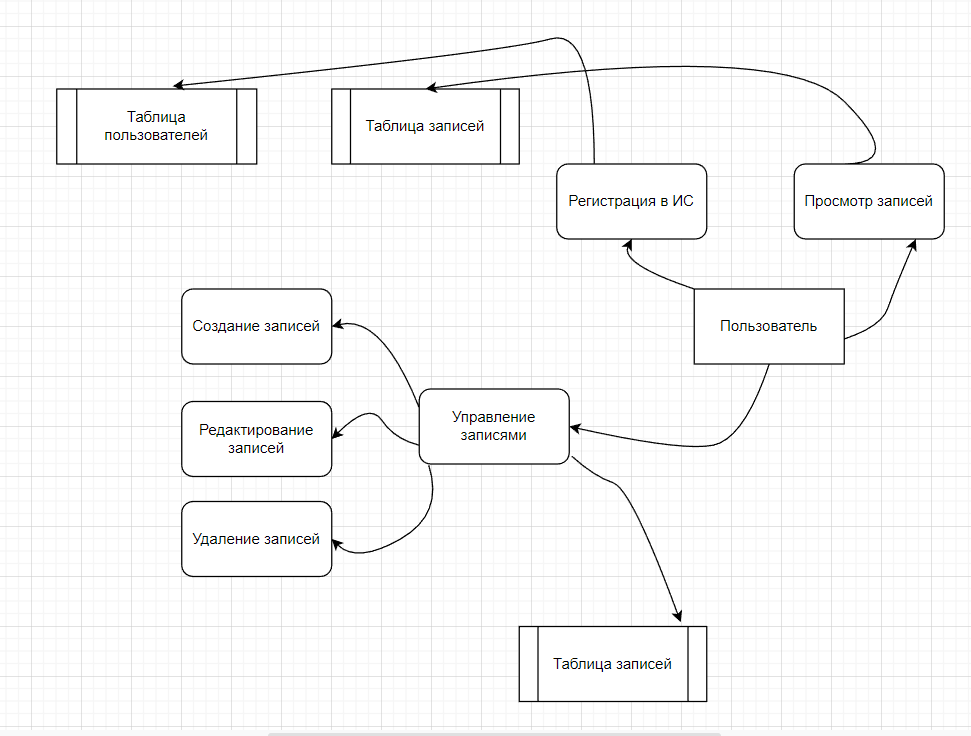


Рисунок 8 – Диаграмма потоков данных

### **4.3 Проектирование базы данных**

Проектирование базы данных начинается с концептуального проектирования базы данных.

Концептуальное проектирование – построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных.

На рисунке 9 представлена инфологическая модель базы данных.

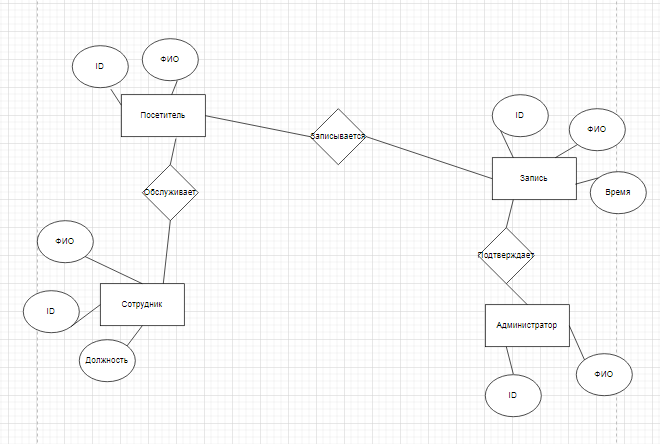


Рисунок 9 – Инфологическая модель базы данных

На инфологической модели базы данных выше схематично отображены сущности системы, их атрибуты и связи между ними. Так, в прямоугольник отображены сущности, такие как: Посетитель, Администратор, Сотрудник, Запись. В овалах отображены атрибуты сущностей, например: ФИО, ID и другие. Ромбами изображены связи между сущностями.

Далее происходит преобразование концептуальной модели в логическую модель, по формальным правилам. На даталогической модели базы данных (рисунок 10) отображены сущности приложения, а также первичные и внешние ключи, связывающие сущности между собой.

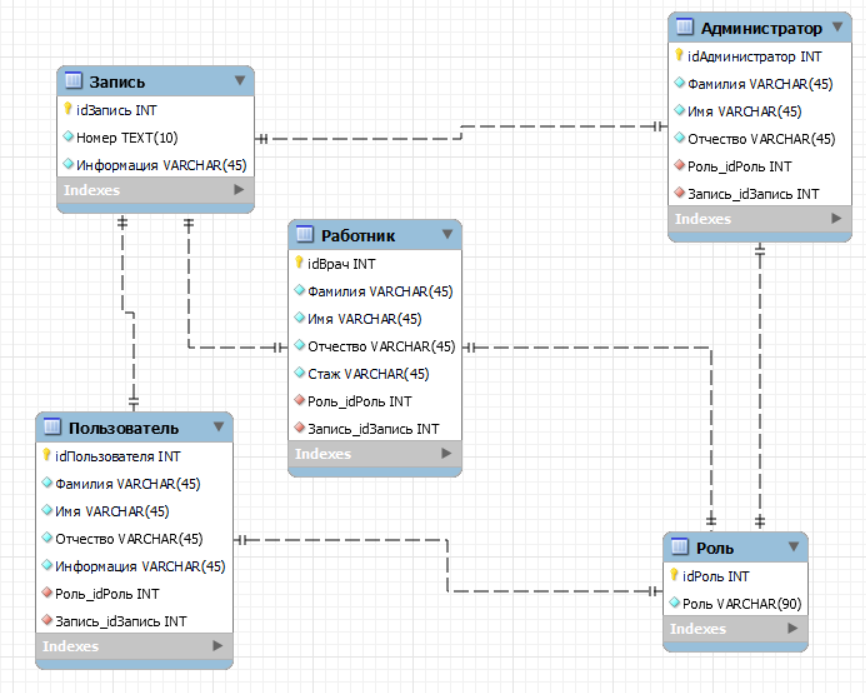


Рисунок 10 – Даталогическая модель базы данных

На рисунке 11 представлена ER-модель базы данных. Она содержит 5 таблиц для полного функционирования и качества сортировки информации.

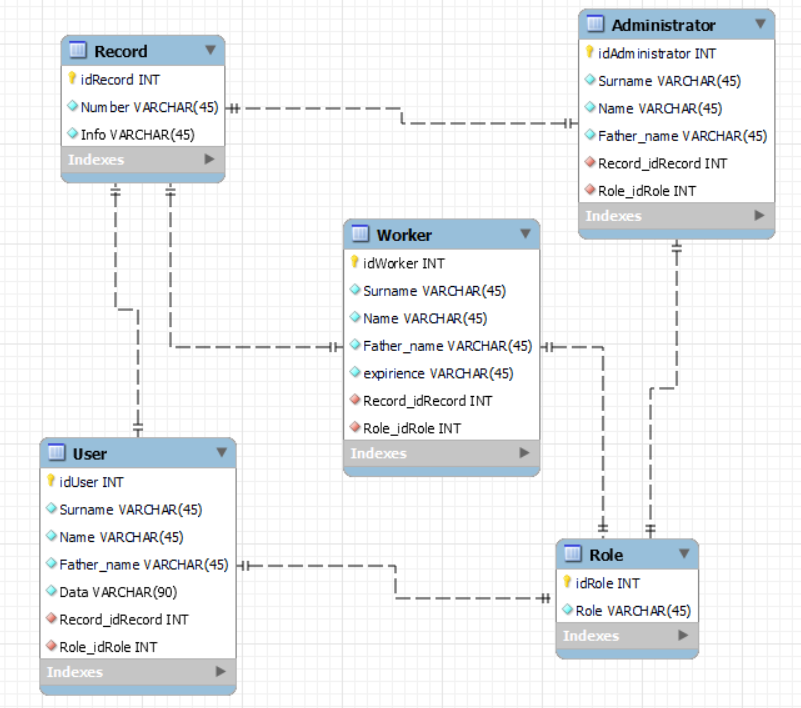


Рисунок 11 – ER-модель базы данных

Далее представлен перечень таблиц 3-7.

Таблица 3 – Таблица Record

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| idRecord | INT | id записи |
| Number | VARCHAR(45) | Фамилия |
| Info | VARCHAR(45) | Имя |

Таблица 4 – Таблица Administrator

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| IdAdministrator | INT | Id администратора |
| Surname | VARCHAR(45) | Фамилия |
| Name | VARCHAR(45) | Имя |
| Father\_name | VARCHAR(45) | Отчество |

Таблица 5 – Таблица Worker

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| IdWorker | INT | Id работника |
| Surname | VARCHAR(45) | Фамилия |
| Name | VARCHAR(45) | Имя |
| Father\_name | VARCHAR(45) | Отчество |
| Experience | VARCHAR(45) | Опыт |

Таблица 6 – Таблица User

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| IdUser | INT | Id пользователя |
| Surname | VARCHAR(45) | Фамилия |
| Name | VARCHAR(45) | Имя |
| Fathername | VARCHAR(45) | Отчество |
| Data | VARCHAR(45) | Информация |

Таблица 7 – Таблица Role

|  |  |  |
| --- | --- | --- |
| Таблица | Тип данных | Описание |
| idRole | INT | INT роли |
| Role | VARCHAR(45) | Роль |

### **4.4. Проектирование интерфейса**

Для разработки пользовательского интерфейса был выбран инструмент Figma – браузерный инструмент для создания интерфейсов.

В результате проектирование интерфейса будущей информационной системы были спроектированы прототипы главная страница (рисунок 12), окно регистрации (рисунок 13), окно авторизации (рисунок 14). Благодаря созданию прототипов, разработка непосредственно программного продукта будет значительно облегчена, за счёт наглядных примеров будущих страниц ИС.

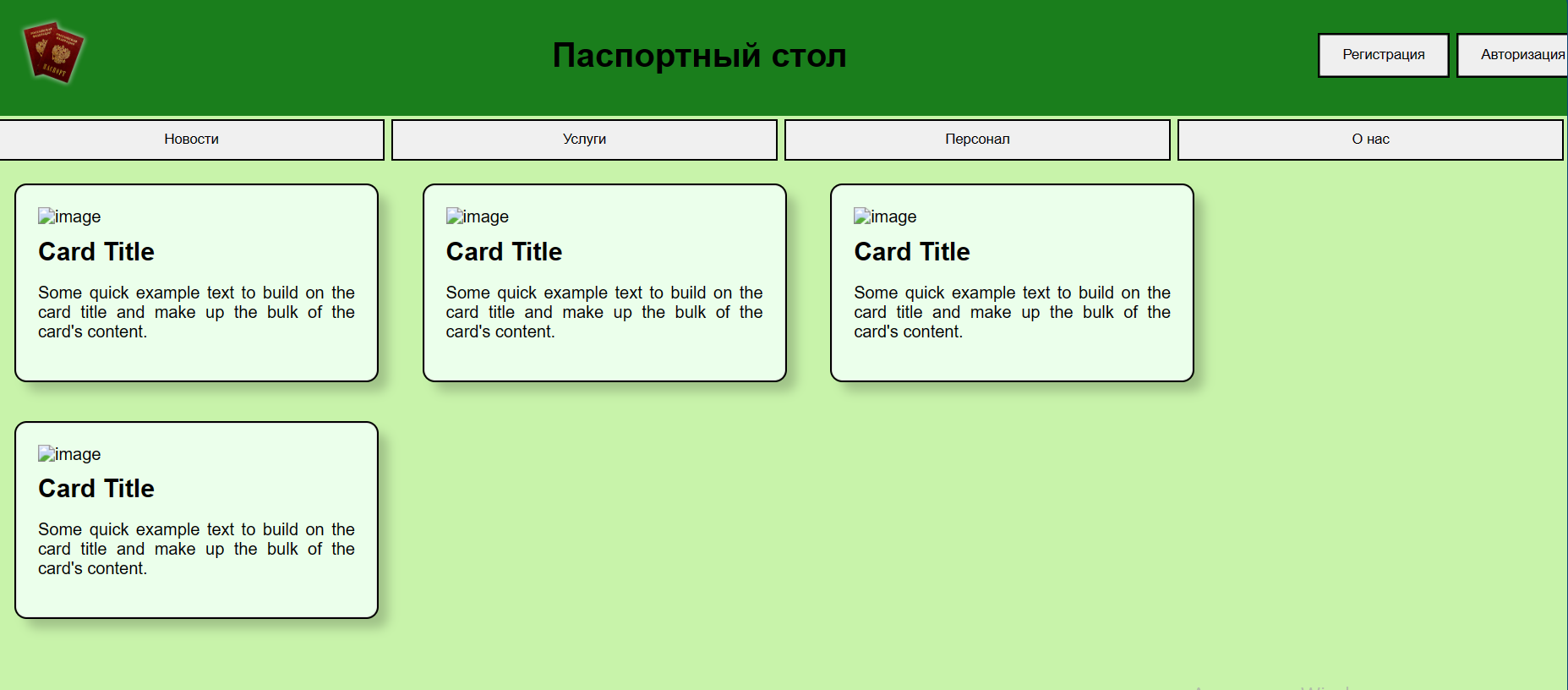


Рисунок 12 – Главная страница

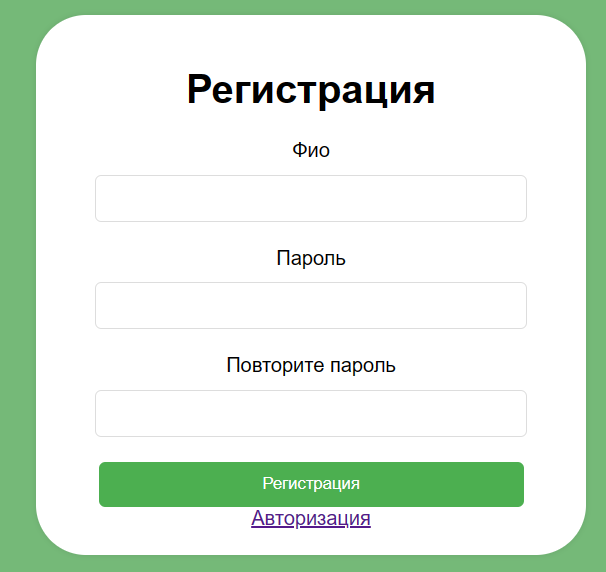


Рисунок 13 – Регистрация

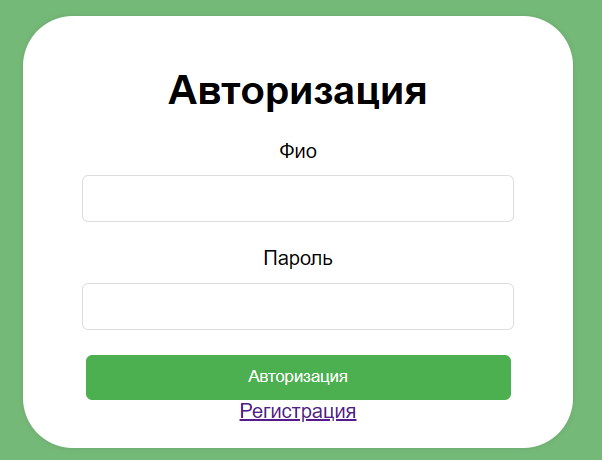


Рисунок 14 – Авторизация

## **Разработка ИС**

### **5.1. Разработка интерфейса ИС**

В информационной системе «Паспортный стол» были поставлены и выполнены следующие задачи:

− авторизация;

− регистрация;

− создание записей;

− просмотр текущих клиентов;

− удаление записей;

− редактирование записей;

− регистрация сотрудников;

− удаление сотрудников;

− редактирование сотрудников.

Программный продукт прост в освоении, т.к. имеет интуитивный интерфейс, в котором разобраться не составит труда даже начинающему пользователю ИС.

Для серверной операционной системы, на которой будет располагаться программный код ИС, выдвигаются следующие минимальные требования:

- процессор: Intel Pentium 4 или более поздней версии;

- ОЗУ 512 Мб DDR3;

- 15 ГБ на HDD;

- поддержка БД MySQL;

- операционная система Windows;

- доступ к сети Интернет.

Требования к клиентской части ИС, необходимые для пользования информационной системой:

- процессор Intel Pentium 4 или более поздней версии;

- ОЗУ 2 ГБ.

- доступ к сети Интернет;

- HDD 1ГБ;

- Браузер;

- встроенный видеоадаптер;

- клавиатура, мышь;

- операционная система: Windows.

Общий размер дисковой памяти, занимаемой информационной системой, составляет 3,5 МБ (рисунок 15).



Рисунок 15 – Объём дисковой памяти ИС

На рисунках 16 - 17 показан html код главной страницы. На ней находится лого, которое играет роль кнопки «Главная страница», кнопка «Авторизация» и «Регистрация».



Рисунок 16 – Первая часть кода

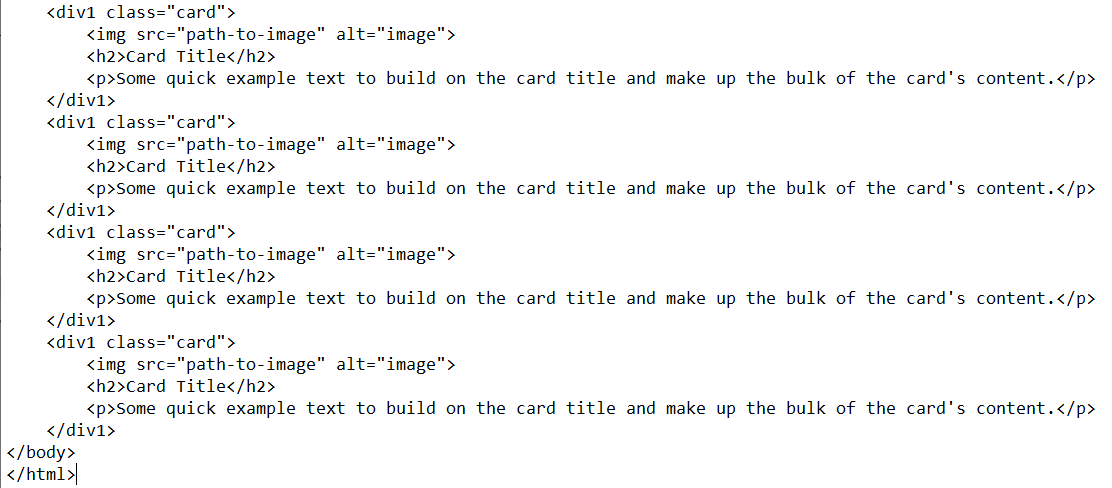


Рисунок 17 – Вторая часть кода

На рисунке 18 изображён результат создания главной страницы.

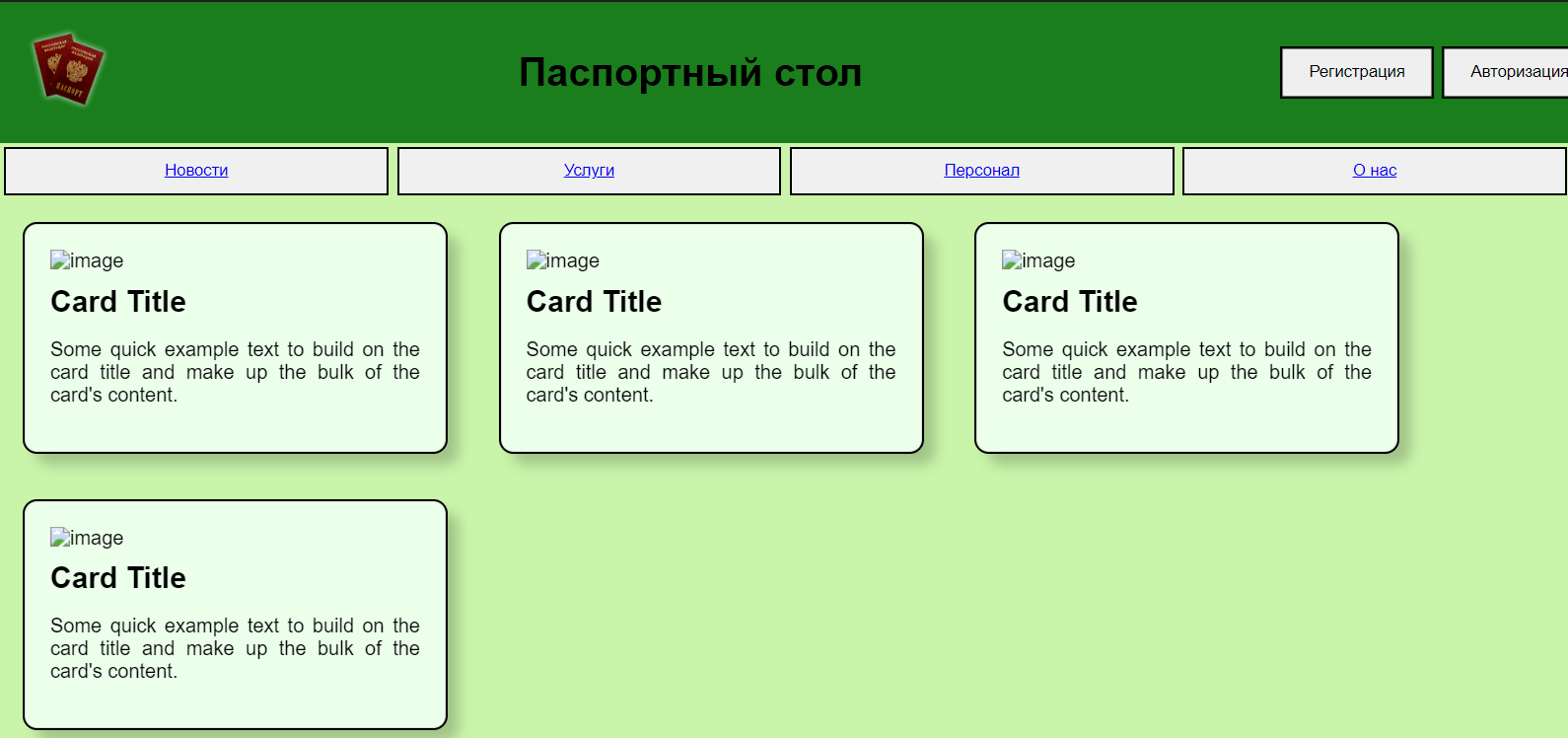


Рисунок 18 – Главная страница

На рисунках 19 изображен html код страницы авторизации.



Рисунок 19 – Кода страницы авторизации

На рисунке 20 изображен результат создания страницы авторизации.

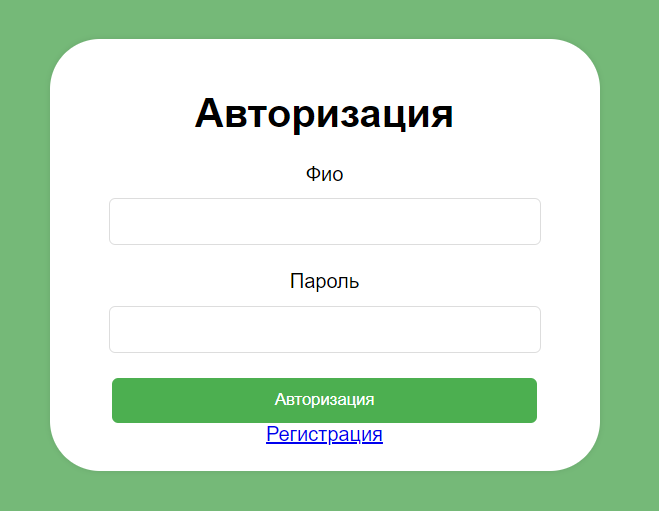


Рисунок 20 – Результат кода страницы авторизации

Так же было создано 2 личных кабинета (далее – ЛК) общего вида: для администратора для пользователя.



Рисунок 21 – Пример ЛК администратора

### **5.2 Разработка базы данных ИС**

В качестве базы данных для реализации программного продукта была выбрана база данных MySQL. Всего в базе данных располагается 5 таблиц (ER ­– модель представлена на рисунке 11).

Все таблицы спроектированы так, чтобы избежать избыточности, при этом достичь максимальной скорости работы, БД нормализирована, и приведена к третьей нормальной форме, таблицы в которой связываются друг с другом посредством внешних ключей.

### **5.3 Разработка ИС**

Подключение к базе данных осуществляется в файле db.php.



Рисунок 22 – db.php

На рисунке 23 представлена модель order. Через модель order осуществляться работа с записями.

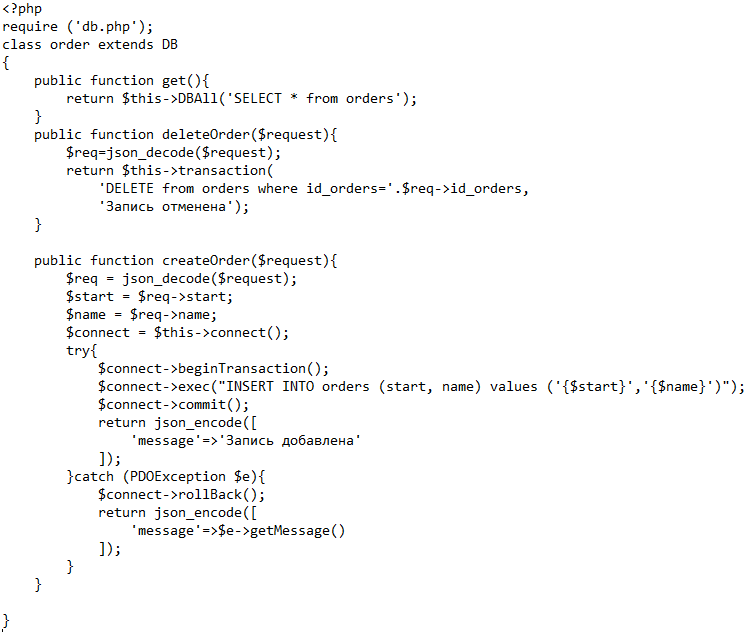


Рисунок 23 – order.php

На рисунке 24 представлена модель services, через которую будет осуществляться работа с услугами.

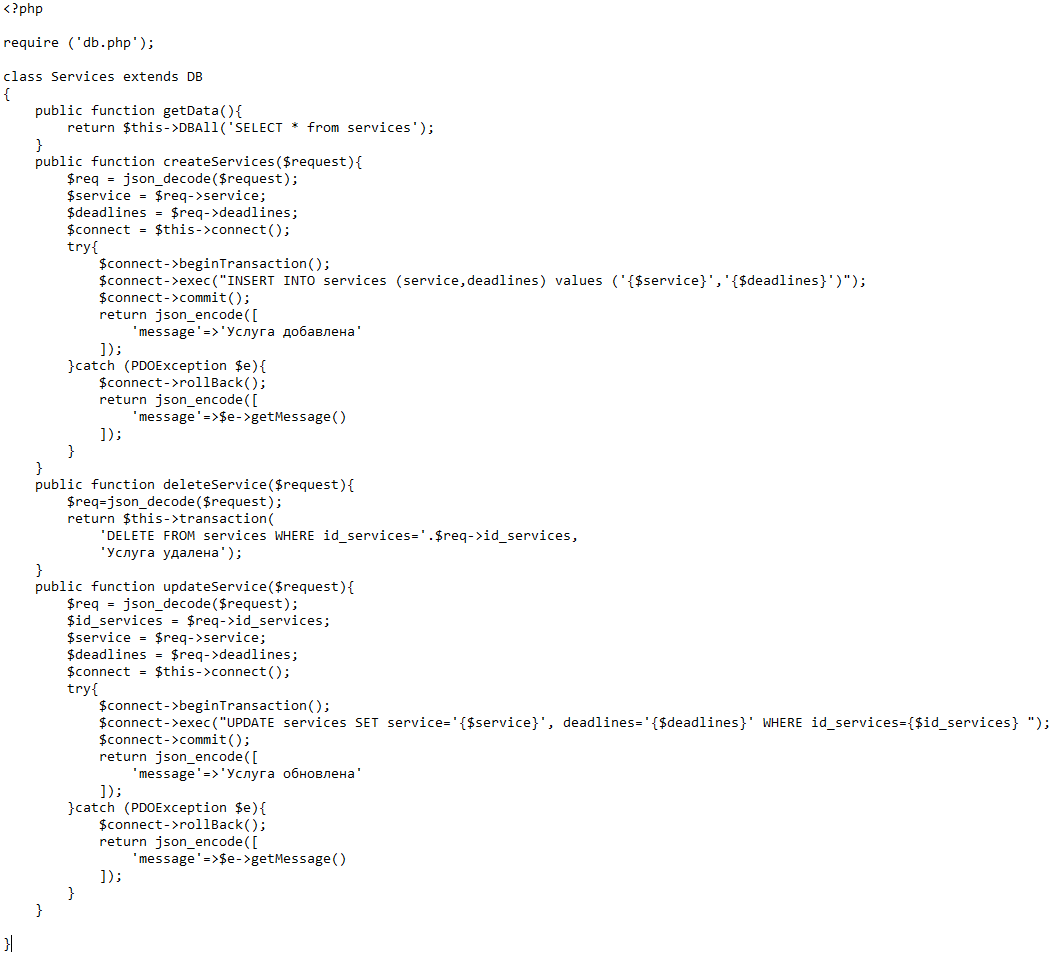


Рисунок 24 – services.php

* 1. Тестирование ИС

Одним из этапов разработки любого программного продукта является тестирование и отладка.

Тестирование – это процесс проверки программного продукта с целью выявления ошибок и подтверждения соответствия требованиям.

Отладка – это процесс выявления, анализа и устранения ошибок в программном коде для обеспечения его правильной работы.

Модульное тестирования - 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.

С помощью метода модульного тестирования на начальном этапе разработки, были обнаружены ошибки в методе создания кабинета.

Для тестирования ИС был разработан сценарий тестирования для каждой модели и представления выбранной предметной области.

Таблица 8 – Сценарий тестирования

|  |  |  |
| --- | --- | --- |
| Тест | Входные данные | Ожидаемый результат |
| Авторизация | Данные пользователя: логин, пароль | Данные добавляются в систему |
| Регистрация | Данные пользователя: логин, пароль | Данные добавляются в систему |
| Добавление заявки | Данные заявки: дата, время, пользователь, сотрудник | Данные заявки сохраняются в базу данных |

Был создан чек-лист, представленный в таблице 9.

Таблица 9 – Чек-лист тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тест | Вводные данные | Ожидаемый результат | Фактический результат | Результат тестирования | Комментарий |
| Авторизация | Данные пользователя: логин, пароль | Пользователь вошел в систему | Пользователь авторизирован | Успешно | - |
| Авторизация | Данные пользователя: логин, пароль | Пользователь вошел в систему | Пользователь не авторизирован | Не успешно | Ошибка в коде |
| Регистрация | Данные пользователя: логин, пароль | Данные добавляются в систему | Данные добавлены | Успешно | - |
| Добавление заявки | Данные заявки: дата, время, пользователь, сотрудник | Данные заявки сохраняются в базу данных | Данные сохранены | Успешно | - |

## **Технологическая документация ИС**

### **6.1 Руководство пользователя ИС**

Для запуска данной информационной системы необходимо открыть файл start.bat и скопировать адрес сайта, далее вставить в адресную строку браузера. Так же нужно запустить XAMPP и запустить Apache и MySQL.

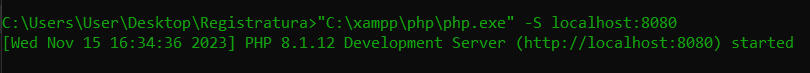


Рисунок 25 – Start.bat

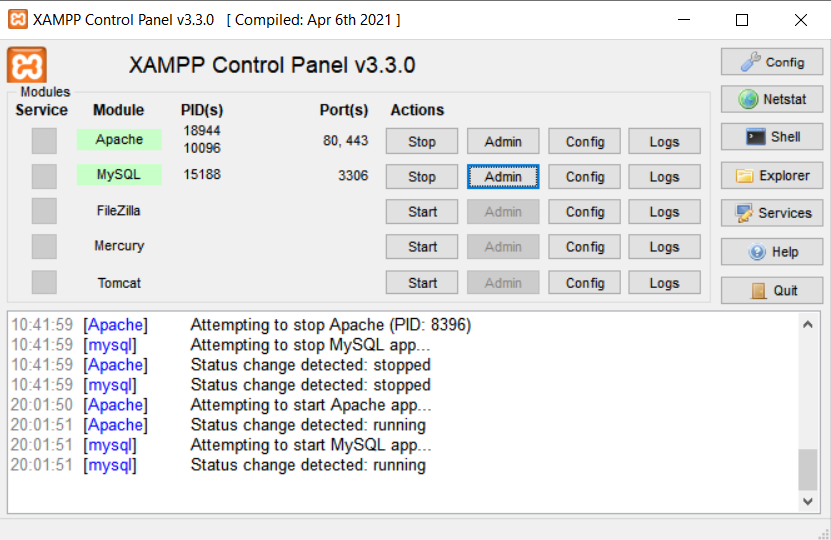


Рисунок 26 - XAMPP

Для использования функционала информационной системы необходимо авторизоваться или зарегистрироваться. При регистрации новый пользователь получает права клиента, и только администратор может выдать права на сотрудника. Страница регистрации представлена на рисунке 27, а страница авторизации на рисунке 28.

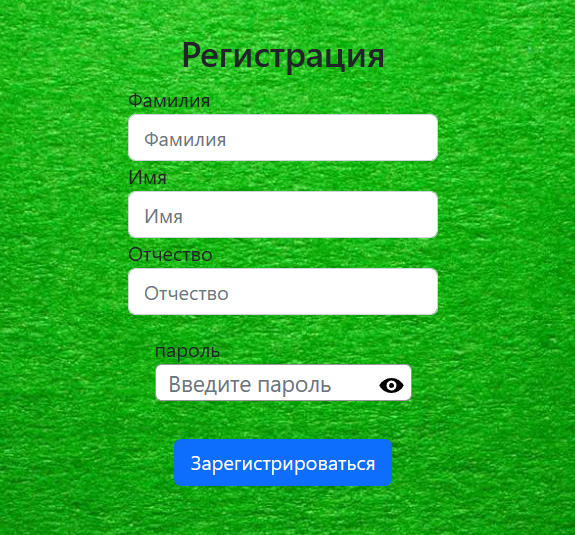


Рисунок 27 – Страница регистрации

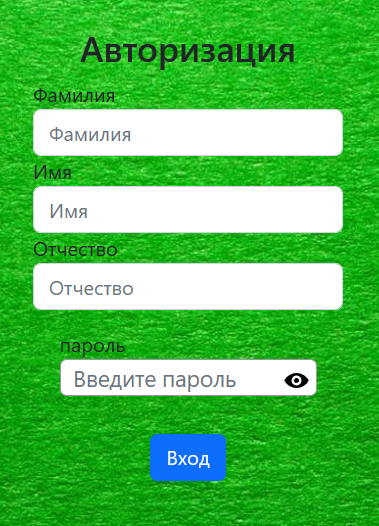


Рисунок 28 – Страница авторизации

После процесса авторизации для заявителя открывается доступ к странице «Мои записи».

Страница «Мои записи» представлена на рисунке 29. На странице имеется 2 кнопки: «Добавить запись» «Отменить запись».

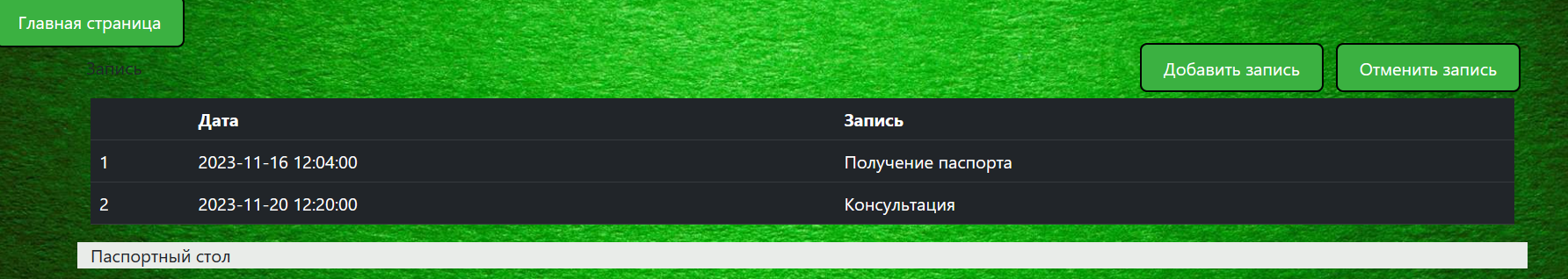


Рисунок 29 – Страница «Мои записи»

При нажатии кнопки «Добавить запись», открывается страница добавления записи, в 2 формах выбирается услуга и время. Страница «Добавить запись» представлена на рисунке 30.

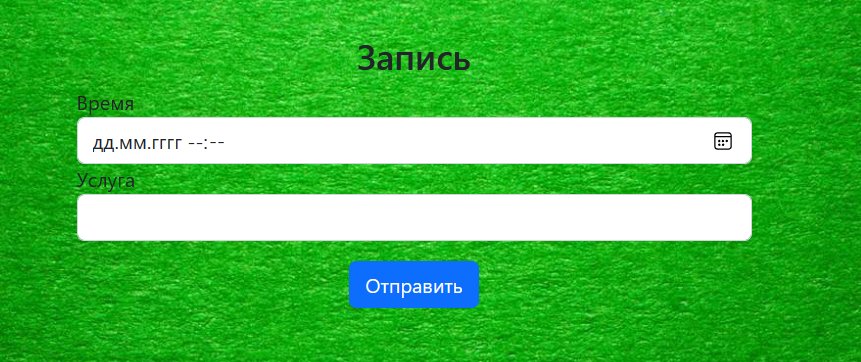


Рисунок 29 – Страница «Добавить запись»

## **Заключение**

В ходе выполнения данной курсовой работы была разработана информационная система «Паспортный стол». Был определен и реализован следующий функционал информационной системы:

− авторизация;

− регистрация;

− создание заказов;

− удаление заказов;

− редактирование заказов;

− создание сотрудников;

− удаление сотрудников;

− редактирование сотрудников;

− создание пользователей;

− удаление пользователей;

− редактирование пользователей.

Были рассмотрены возможные реализации ИС с использованием разных технологий и языков программирования, но в результате анализа инструментальных средств разработки выбор остановился на языке программирования PHP.

В результате разработан браузерный программный продукт, с интуитивным и современным интерфейсом и широким функционалом.

В дальнейшем созданная в ходе реализации проекта ИС может развиваться путём расширения функционала и совершенствования интерфейса, тем самым набирая большее сообщество пользователей.

Таким образом, выполнение поставленных целей и задач курсового проекта позволяет сделать вывод о его успешности и значимости для практического применения.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

* 1. Studbooks – ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ – URL: https://studbooks.net/2191158/informatika/obsledovanie\_predmetnoy\_oblasti?ysclid=loy8h9sshc392026343 (дата обращения: 26.09.2023). – Текст: электронный.
  2. Metanit – Руководство по HTML5 и CSS3 – URL: https://metanit.com/web/html5/ (дата обращения: 05.10.2023). – Текст: электронный.
  3. Metanit – Руководство по PHP и MySQL – URL: https://metanit.com/php/mysql/ (дата обращения: 15.10.2023). – Текст: электронный.
  4. Htmlacademy.ru – Определение и особенности языка PHP. – URL: https://htmlacademy.ru/tutorial/php/basics. (дата обращения 07.10.23) – Текст: электронный.
  5. HTML5book – Основы CSS – URL: https://html5book.ru/osnovy-css/?ysclid=lozmft0x58167998961(дата обращения:10.10.2023). – Текст: электронный.

## **Приложение А – Техническое задание**

**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ИНФОРМАЦИОННАЯ СИСТЕМА

«Паспортный стол»

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.П. Морозов) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.В. Макрицкий) |
|  |  |  |

Иркутск 2023

**1. Общие сведения**

Наименование работы: информационная система «Паспортный стол».

Исполнитель: студент ГБПОУ «Иркутского Авиационного техникума», группы ИС-20-1, Макрицкий Андрей Владимирович

Разработка информационной системы проходит в рамках курсовой работы.

Сроки разработки информационной системы с 15.09.2023 по 23.11.2023 года.

**2. Цели и назначение создания автоматизированной системы**

Целью курсовой работы является создание информационной системы «Паспортный стол», которое включает в себя возможность записи на прием.

В данной информационной реализуется следующий функционал:

– ­авторизация;

– регистрация;

– запись на прием;

– просмотр записей;

– удаление записей.

**3. Характеристика объекта автоматизации**

Данная информационная система разрабатывается для облегчения записи в паспортный стол. При помощи сайта будет проще записаться к специалисту.

**4. Требования к системе в целом**

**4.1 Требования к структуре и функционированию сайта**

Функции информационная система:

1. Окно «Авторизации»:
   1. авторизация пользователей.
2. Окно «Регистрация»:
   1. регистрация пользователей.
3. Окно «Главная страница»:
   1. Информация о записях;
4. Окно «Услуги»:
   1. Таблица «Услуги»:
      1. добавление записи;
      2. удаление записи;
      3. редактирование данных.
   2. Окно «Посетители»:
      1. добавление посетителей;
      2. удаление посетителей;
      3. редактирование данных посетителей.
   3. Окно «Сотрудники»:
      1. добавление сотрудников;
      2. удаление сотрудников;
      3. редактирование данных.

**4.2 Требования к надежности**

Для обеспечения надежности необходимо проверять корректность получаемых данных. Входные данные поступают в виде значений c клавиатуры. Эти значения отображаются в отдельных полях таблицы.

**4.3 Требования к безопасности**

Для обеспечения безопасности в информационной системе, необходимо реализовать разграничение прав доступа.

**4.4 Требования к эксплуатации, техническому обслуживанию, ремонту и** **хранению компонентов системы**

Минимальные системные требования для сервера:

1. Процессор: Intel Pentium 4 2.0Ghz / AMD XP 2200+;
2. Оперативная память: 512 Мб;
3. Жёсткий диск: 150мб;
4. Операционная система: Windows 7/8/10.
5. Версия MySQL 5.0 и выше;

Минимальные системные требования для рабочей станции:

1. Процессор: Intel Pentium 4 2.0Ghz / AMD XP 2200+;
2. Оперативная память: 512 Мб;
3. Жёсткий диск: 150мб;
4. Операционная система: Windows 7/8/10.

**5. Состав и содержание работ по созданию системы**

В таблице 8 представлены плановые сроки начала и окончания работы по созданию веб-приложения.

Таблица 8 – Этапы выполнения разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов курсового проекта | Срок | | Объем |
| Начало | Окончание |
| Предпроектное исследование предметной области (выбор темы, постановка цели, задач, описание области применения, исследование предметной области) | 15.09.23 | 21.09.23 | 10% |
| Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования) | 22.09.23 | 30.09.23 | 20% |
| Проектирование программного обеспечения.  (разработка структурной и функциональной схемы ПО, проектирование базы данных (инфологическое, ER-модель, физическая модель) | 02.10.23 | 12.10.23 | 50% |
| Разработка (программирование) программного продукта | 13.10.23 | 06.11.23 | 80% |
| Тестирование и отладка программного продукта (функциональное тестирование, составление тест-планов) | 07.11.23 | 14.11.23 | 90% |
| Составление программной документации (оформление ПЗ, написание руководства пользователя, составление презентации и речи) | 15.11.23 | 22.11.23 | 100% |
| Защита курсового проекта | 23.11-30.11 | |

## **Приложение Б – Листинг roles.php**

<?php  
require('db.php');  
class roles extends DB  
{  
 public function login($request)  
 {  
 $req = json\_decode($request);  
 $last\_name = $req->last\_name;  
 $name = $req->name;  
 $father\_name = $req->father\_name;  
 $password = $req->password;  
 $connect = $this->connect();  
 $sql = $connect->prepare('SELECT \* from users where last\_name=:last\_name and name=:name and father\_name=:father\_name and password=:pass');  
 $sql->execute([  
 'last\_name' => $last\_name,  
 'name' => $name,  
 'father\_name' => $father\_name,  
 'pass' => $password,  
 ]);  
 $data = $sql->fetch(PDO::*FETCH\_OBJ*);  
 if ($data) {  
 session\_start();  
 $\_SESSION['user'] = (object)[  
 'last\_name' => $data->last\_name,  
 'name' => $data->name,  
 'father\_name' => $data->father\_name,  
 'role' => $data->role  
 ];  
 }  
 }  
 public function registration($request)  
 {  
 $req = json\_decode($request);  
 $last\_name = $req->last\_name;  
 $name = $req->name;  
 $father\_name = $req->father\_name;  
 $password = $req->password;  
 $connect = $this->connect();  
 $sql = $connect->prepare('SELECT \* from users where last\_name=:last\_name and name=:name and father\_name=:father\_name and password=:pass');  
 $sql->execute(array(  
 'last\_name' => $last\_name,  
 'name' => $name,  
 'father\_name' => $father\_name,  
 "pass" => $password,  
 ));  
 $data = $sql->fetch(PDO::*FETCH\_OBJ*);  
 if ($data) {  
 return json\_encode([  
 'message' => "Такой пользователь существует"  
 ]);  
 }  
 $req = json\_decode($request);  
 $last\_name = $req->last\_name;  
 $name = $req->name;  
 $father\_name = $req->father\_name;  
 $connect = $this->connect();  
 $sql = $connect->prepare('SELECT \* from patients where last\_name=:last\_name and name=:name and father\_name=:father\_name');  
 $sql->execute(array(  
 'last\_name' => $last\_name,  
 'name' => $name,  
 'father\_name' => $father\_name,  
 ));  
 $data = $sql->fetch(PDO::*FETCH\_OBJ*);  
 if ($data) {  
 return json\_encode([  
 'message' => "Такой пользователь существует"  
 ]);  
 }  
 $sql = $connect->prepare("INSERT INTO users(last\_name,name,father\_name,password,role) values (:last\_name,:name,:father\_name,:pass,:role)");  
 $sql->execute([  
 'last\_name' => $last\_name,  
 'name' => $name,  
 'father\_name' => $father\_name,  
 "pass" => $password,  
 "role" => 3  
 ]);  
 $sql = $connect->prepare("INSERT INTO patients (last\_name,name,father\_name) values (:last\_name,:name,:father\_name)");  
 $sql->execute([  
 'last\_name' => $last\_name,  
 'name' => $name,  
 'father\_name' => $father\_name,  
 ]);  
 $sql = $connect->prepare('SELECT \* from users where last\_name=:last\_name and name=:name and father\_name=:father\_name and password=:pass');  
 $sql->execute([  
 'last\_name' => $last\_name,  
 'name' => $name,  
 'father\_name' => $father\_name,  
 "pass" => $password,  
 ]);  
 $data = $sql->fetch(PDO::*FETCH\_OBJ*);  
 if ($data) {  
 session\_start();  
 $\_SESSION['user'] = (object)[  
 'last\_name' => $data->last\_name,  
 'name' => $data->name,  
 'father\_name' => $data->father\_name,  
 'role' => $data->role  
 ];  
 return json\_encode([  
 'message' => 'Пользователь добавлен'  
 ]);  
 }  
 $sql = $connect->prepare('SELECT \* from patients where last\_name=:last\_name and name=:name and father\_name=:father\_name');  
 $sql->execute([  
 'last\_name' => $last\_name,  
 'name' => $name,  
 'father\_name' => $father\_name,  
 ]);  
 $data = $sql->fetch(PDO::*FETCH\_OBJ*);  
 if ($data) {  
 session\_start();  
 $\_SESSION['user'] = (object)[  
 'last\_name' => $data->last\_name,  
 'name' => $data->name,  
 'father\_name' => $data->father\_name,  
 ];  
 return json\_encode([  
 'message' => 'Пользователь добавлен'  
 ]);  
}