**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| КП.09.02.07-5.23.202.17 ПЗ |  |

ИНФОРМАЦИОННАЯ СИСТЕМА

«МАГАЗИН КОМПЛЕКТУЮЩИХ ДЛЯ ПК»

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

Иркутск 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc150873393)

[1 Предпроектное исследование 4](#_Toc150873394)

[2 Обзор инструментальных средств для разработки 6](#_Toc150873395)

[3 Техническое задание 13](#_Toc150873396)

[4 Проектирование ИС 14](#_Toc150873397)

[4.1 Структурная схема ИС 14](#_Toc150873398)

[4.2 Функциональная схема ИС 16](#_Toc150873399)

[4.3 Проектирование базы данных 19](#_Toc150873400)

[4.4 Проектирование интерфейса 22](#_Toc150873401)

[5 Разработка ИС 24](#_Toc150873402)

[5.1 Разработка интерфейса ИС 24](#_Toc150873403)

[5.2 Разработка базы данных ИС 30](#_Toc150873404)

[5.3 Разработка ИС 32](#_Toc150873405)

[6 Тестирование ИС 37](#_Toc150873406)

[6.1 Модульное тестирование 41](#_Toc150873407)

[6.2 Автоматическое тестирование 44](#_Toc150873408)

[6.3 Тестирование базы данных 45](#_Toc150873409)

[7 Технологическая документация 50](#_Toc150873410)

[7.1 Руководство пользователя ИС 50](#_Toc150873411)

[ЗАКЛЮЧЕНИЕ 54](#_Toc150873412)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 56](#_Toc150873413)

[Приложение А – Техническое задание 57](#_Toc150873414)

[Приложение Б – Листинг регистрация 62](#_Toc150873415)

# ВВЕДЕНИЕ

В наше время компьютеры и информационные технологии стали неотъемлемой частью жизни человека. С увеличением числа пользователей ПК возросла и потребность в качественных комплектующих. Магазины комплектующих для ПК играют ключевую роль предоставляя необходимые компоненты для стабильной работы компьютеров.

С появлением новых технологий и постоянным ростом интереса к компьютерам, рынок комплектующих для ПК стал крайне конкурентоспособным. Магазины этого профиля вынуждены постоянно совершенствовать свои процессы, чтобы удовлетворить требования клиентов и оставаться конкурентоспособными.

Цель курсового проекта разработать информационную систему (далее - ИС) «Магазин комплектующих для ПК».

Основными задачами курсового проекта является:

* Описать предметную область.
* Проанализировать инструментальные средства разработки.
* Создать техническое задание.
* Спроектировать ИС.
* Разработать ИС.
* Создать документацию.
* Провести тестирование программного продукта.
* Разработать руководство пользователя.

# 1 Предпроектное исследование

Предметной областью курсового проекта является информационная система, с помощью которой пользователь сможет купить различные устройства для сборки компьютера.

Для начала клиент выбирает товары которые ему нужны, добавляет в корзину и заказывает.

Администратор может управлять товарами, то есть добавлять новые товары и удалять ненужные.

В данной предметной области документооборот включает в себя информацию о следующих компонентах:

* Товар.
* Клиент.
* Заказ.

Можно выделить следующие сущности программного продукта.

* Пользователь. Атрибуты пользователя – Код пользователя, Фамилия, Имя, Роль.
* Товар. Атрибуты товара – Код товара, Цена, Описание, Название, Картинка.

Функциональные возможности:

* регистрация пользователей;
* авторизация пользователей;
* выбор товаров;
* добавление товара в корзину;
* добавление товара администратором;
* удаление товара администратором;

# 2 Обзор инструментальных средств для разработки

Планируется разработать информационную систему, способную работать в локальном режиме и не потребляющую ресурсы устройства при его использовании.

Для рассмотрения инструментальных средств разработки программных продуктов я взял Node.js и PHP.

Среда разработки – важная часть при создании будущего приложения. От неё зависит удобство и скорость написания кода в процессе работы. Для разработки программного продукта рассмотрены такие среды разработки, как PhpStorm и Visual Studio Code.

На рассмотрение средств для управления базами данных были выбраны MySQL, SQLite, PostgreSQL.

На выбранном языке программирования будет создаваться будущий продукт, поэтому от его выбора зависит скорость работы, качество, работоспособность продукта и многое другое.

Инструментальные средства разработки программных продуктов:

1. Node.js.
2. PHP.

Node.js – программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера.

Главная особенность Node.js – то, что вместо традиционной модели параллелизма на основе потоков автор выбрал событийно-ориентированный подход. Это такая парадигма программирования, в которой выполнение программ определяется событиями. Node.js работает быстро. Это доказывают результаты тестов производительности, представленные toptal.com. Исследователи сравнили, как популярные языки, такие как GO, PHP, Java и Node.js, обрабатывают одновременные запросы. Ещё одним немаловажным преимуществом является асинхронная обработка запросов. В контексте серверной части синхронная обработка предполагает, что код выполняется последовательно.

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Язык автоматически поддерживает HTTP Cookies в соответствии со стандартами Netscape. Это позволяет проводить установку и чтение небольших сегментов данных на стороне клиента. Работа с Cookies организована посредством сеансов (сессий). У сессий есть срок действия (после его истечения данные удаляются), в сессиях можно хранить и редактировать разные типы данных, в том числе сериализованные PHP-объекты, пропущенные через serialize (процесс происходит автоматически).

Для наглядности сравнения языков программирования была составлена таблица 1.

Таблица 1 – Сравнение языков программирования для разработки

|  |  |  |
| --- | --- | --- |
| Критерии/Языки программирования | Node.js | PHP |
| Подходит для разработки серверной части | + | + |
| Понятный синтаксис | – | + |
| Хорошая типизация | + | + |
| Простота в освоении | – | + |

После рассмотрения был выбран язык PHP, так как он подходит для разработки серверной части и не сложный в освоении.

Далее рассмотрены средства разработки программных продуктов:

1. PhpStorm.
2. Visual Studio Code.

PhpStorm — коммерческая кроссплатформенная интегрированная среда разработки для PHP. Разрабатывается компанией JetBrains на основе платформы IntelliJ IDEA.

PhpStorm представляет собой интеллектуальный редактор для PHP, HTML и JavaScript с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для PHP и JavaScript. Автодополнение кода в PhpStorm поддерживает спецификацию PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2, 7.4 и 8.0. Имеется полноценный SQL-редактор с возможностью редактирования полученных результатов запросов.

PhpStorm разработан на основе платформы IntelliJ IDEA, написанной на Java. Пользователи могут расширить функциональность среды разработки за счет установки плагинов, разработанных для платформы IntelliJ, или написав собственные плагины.

Visual Studio Code – текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

Visual Studio Code основан на Electron и реализуется через веб-редактор Monaco, разработанный для Visual Studio Online.

Для наглядности сравнения IDE была составлена таблица 2.

Таблица 2 – Сравнение IDE для разработки программного продукта

|  |  |  |
| --- | --- | --- |
| Критерии/Среды разработки | PhpStorm | Visual Studio Code |
| Удобный и понятный интерфейс | + | + |
| Бесплатный доступ | – | + |
| Поддерживается на Windows | + | + |
| Поддерживается на слабых ПК | + | + |

После рассмотрения всех плюсов и минусов данных сред разработки был выбран Visual Studio Code

СУБД – неотъемлемая часть будущего продукта. Благодаря ему мы будем работать с базами данных, которые являются хранилищем необходимых данных.

Системы управления базами данных для разработки программных продуктов:

1. MySQL;
2. SQLite;
3. PostgreSQL.

MySQL – свободная реляционная система управлением 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.

MySQL является решением для малых и средних приложений. 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.

PostgreSQL – свободная объектно-реляционная система хранения и управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

SQLite – компактная встраиваемая СУБД с исходным кодом. Поддерживает динамическое типизированные данных. Возможные типы значений: INTEGER, REAL, TEXT и BLOB

Для наглядности сравнения СУБД была составлена таблица 3.

Таблица 3 – Сравнение средств управления базами данных

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

После рассмотрения СУБД для проекта была выбрана MySQL.

Таким образом, для создания программного продукта было решено использовать средства:

1. Для создания структурных схем, контекстной и диаграмм декомпозиции использовались CASE-средства – Draw.io.
2. Для структурирования, чтения, изменения и удаления информации использовался инструмент по работе с базой данных PHP MyAdmin.
3. Для разработки дизайна информационной системы использовался онлайн – сервис для разработки дизайна и прототипа сайта или приложения – Figma.
4. Средствами реализации ИС были выбраны: язык стилей CSS, язык написания разметки HTML, язык программирования серверной части PHP, язык программирования JS. Инструменты для проектирования и разработки ИС – Figma, Draw.io, Visual Studio Code, MySQL Workbench.
5. Для тестирования программного продукта был выбран инструмент для автоматизации действий веб-браузера Selenium, сценарии тестирования с чек-листами, а также база данных тестировалась с помощью метода CRUD.

Основные преимущества информационных технологий:

1. Доступность – пользователь может пользоваться информационной системой из любой точки мира при доступе в интернет.
2. Переносимость – информационную систему легко перенести с одного места хранения на другое.
3. Простота интерфейса – пользоваться информационной системой интуитивно понятно.
4. Секретность и безопасность – доступ к данным ограничен и защищён несколькими способами.
5. Масштабность – предоставление доступа большому количеству пользователей.
6. Простота обслуживания – удобное добавление новых функций в информационную систему.

3 Техническое задание

В начале разработки создавалось техническое задание, в котором указывались основные требования.

Для создания технического задания использовался стандарт ГОСТ 34.602-2020.

Согласно ГОСТ 34.602-2020 техническое задание должно включать следующие разделы:

* Программный продукт разработан в соответствии техническим заданием.
* Программный продукт должен оперативно работать с данными в данной предметной области.
* Сведения должны храниться и использоваться при составлении и отборе статистических данных.
* На этапе проектирования необходимо провести моделирование, описывающее процесс создания и эксплуатации программного продукта.
* Работа с исходными данными должна предусматривать режим коллективного доступа.
* Пользователь должен иметь возможность влиять на результаты расчетов путем выбора параметров данных, вида требуемой операции.
* Работа пользователя с программным продуктом должна строиться в виде диалогового процесса с последовательным выбором действий.
* Результатами работы программного продукта является выполнение операций, связанных с БД, такие как изменение, хранение, добавление данных и т.д.

Техническое задание на разработку информационной системы представлено в приложении А.

4 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 20 функций, часть из которых может делать несколько актёров, а часть – только определённый актёр.



Рисунок 1 – Диаграмма прецедентов

На рисунке 2 представлена диаграмма деятельности. Она содержит 2 роли. Процесс начинается с клиента, который регистрируется в системе и формирует заказ. Далее менеджер проверяет статус заказа, передает информацию.

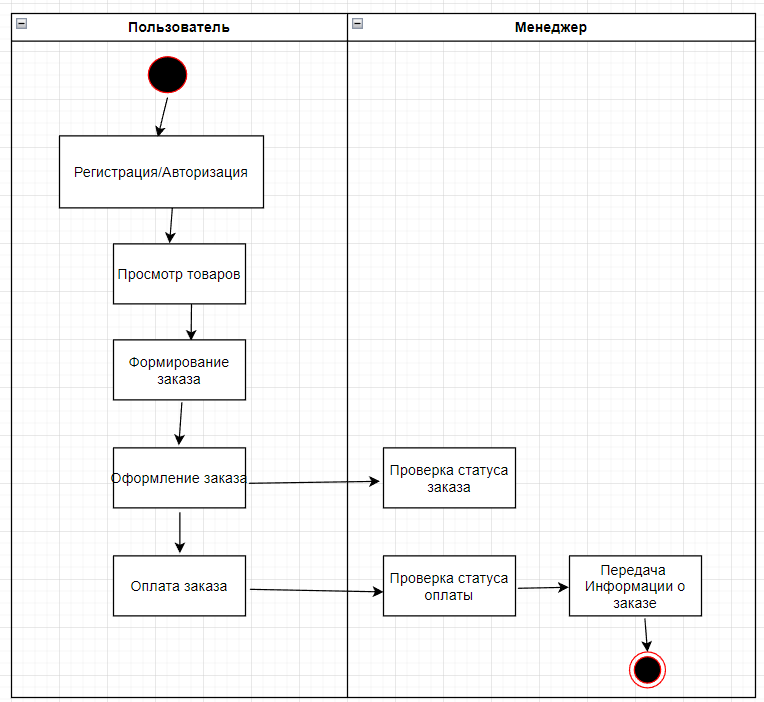


Рисунок 2 – Диаграмма деятельности

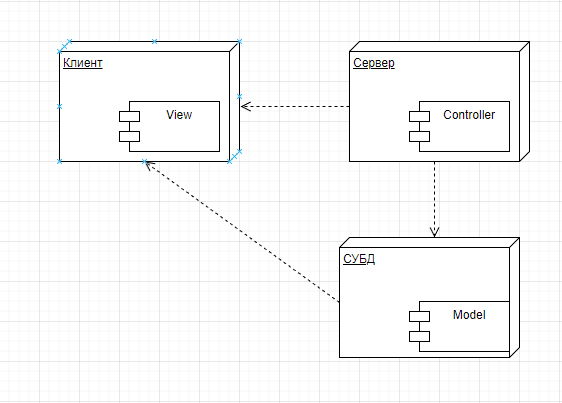


Рисунок 3 – Диаграмма развертывания

На рисунке 4 представлена диаграмма компонентов. В ней отражены элементы, которые служат для регистрации, авторизации и выхода из своей учётной записи, просмотра и создания заказов. А также элементы для административного функционала.

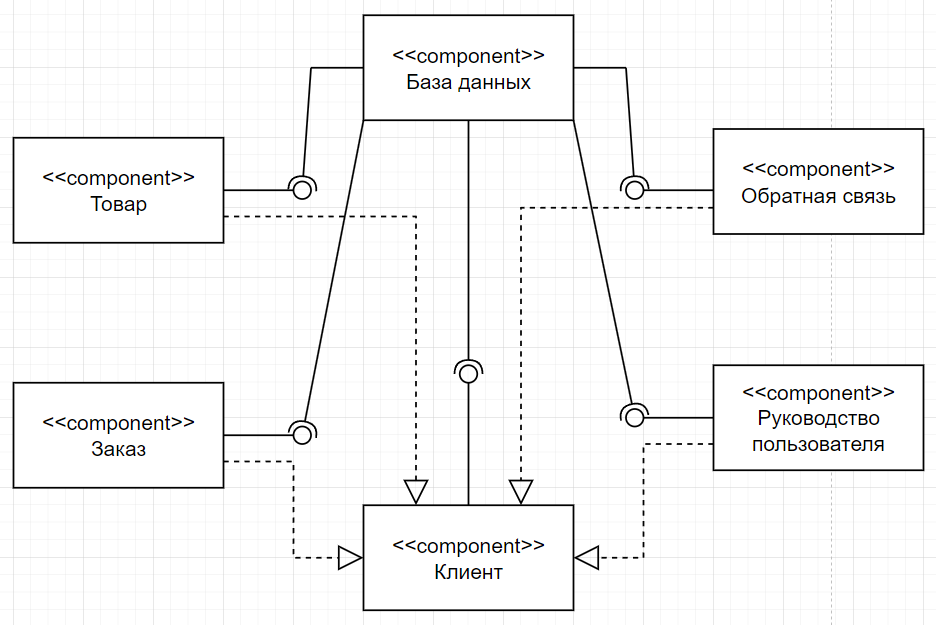


Рисунок 4 – Диаграмма компонентов

4.2 Функциональная схема ИС

На рисунке 5 представлена контекстная диаграмма, отображающая деятельность ИС.

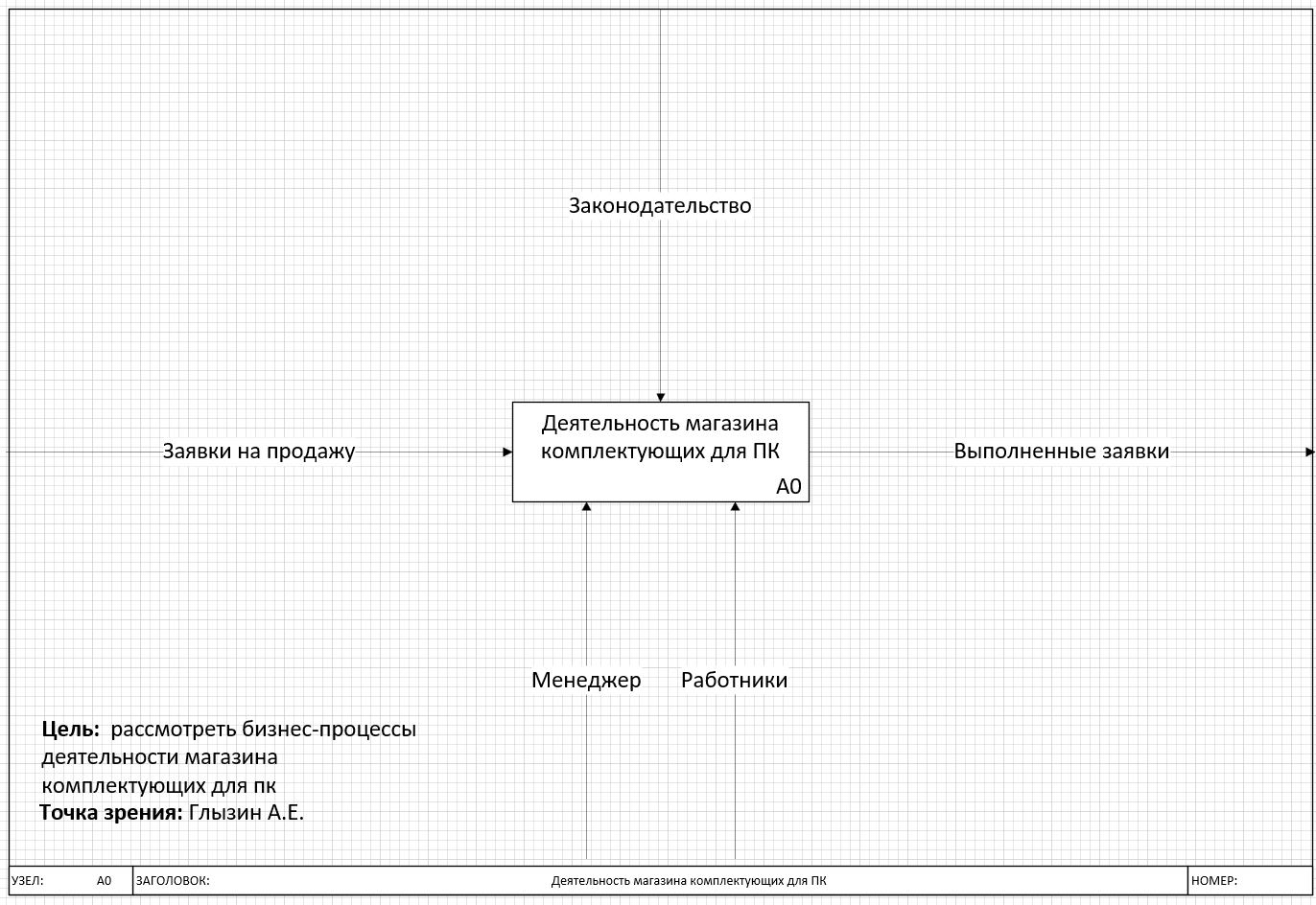
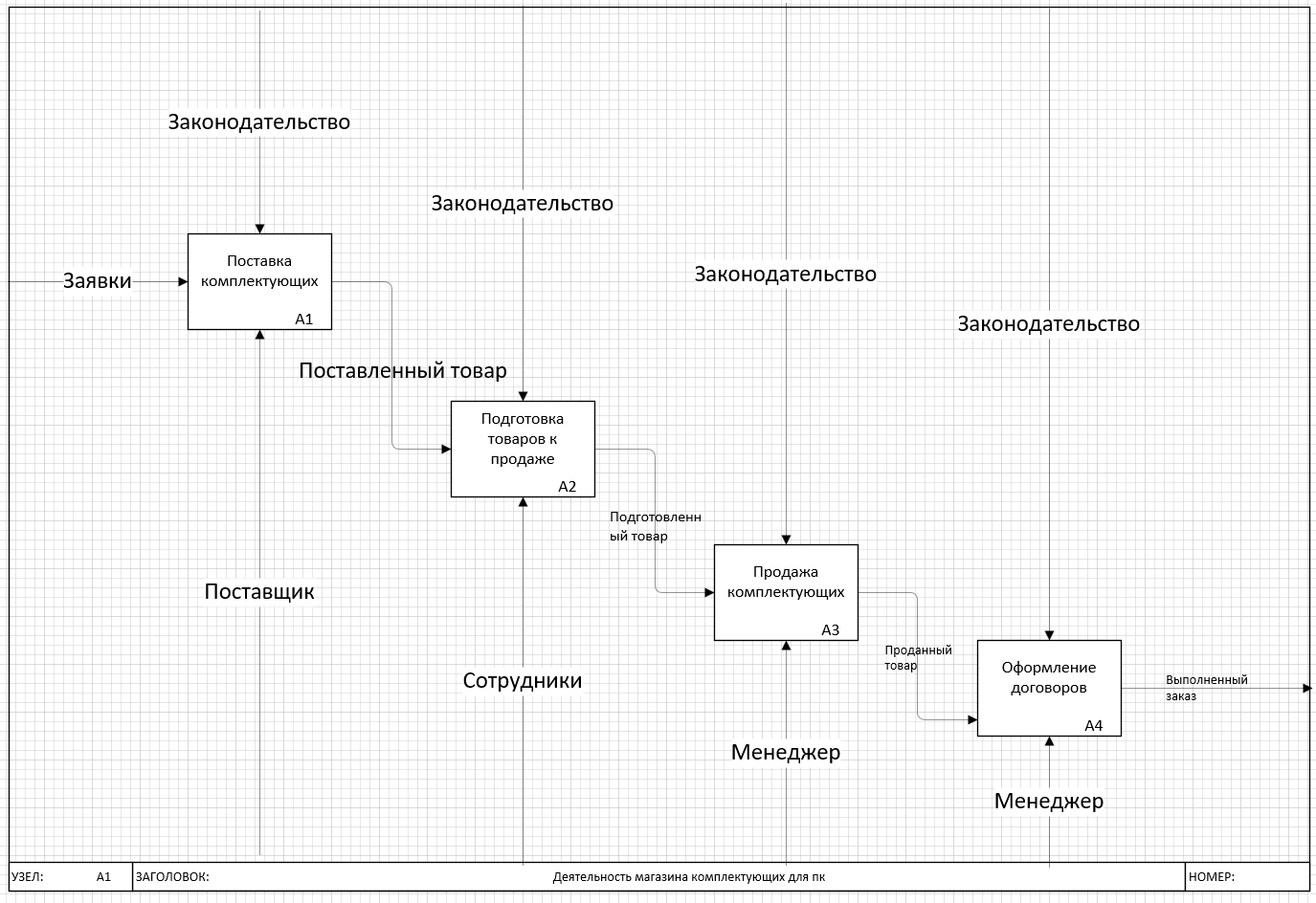


Рисунок 5 – Контекстная диаграмма IDEF0

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



Отказ

Заказ

Данные

Рисунок 6 – Диаграмма декомпозиций А1

На рисунке 7 представлена диаграмма классов. Она содержит 6 классов, среди них: товар, пользователь, заказ

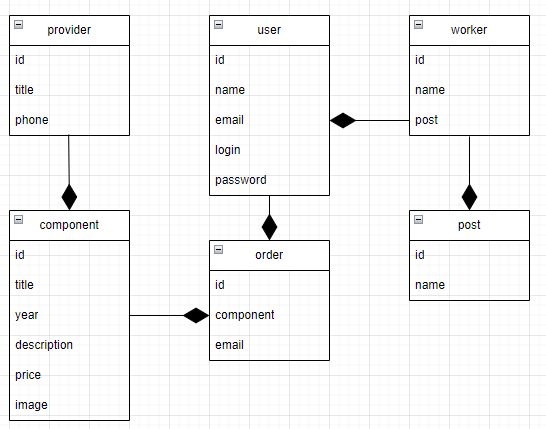


Рисунок 8 – Диаграмма классов

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.

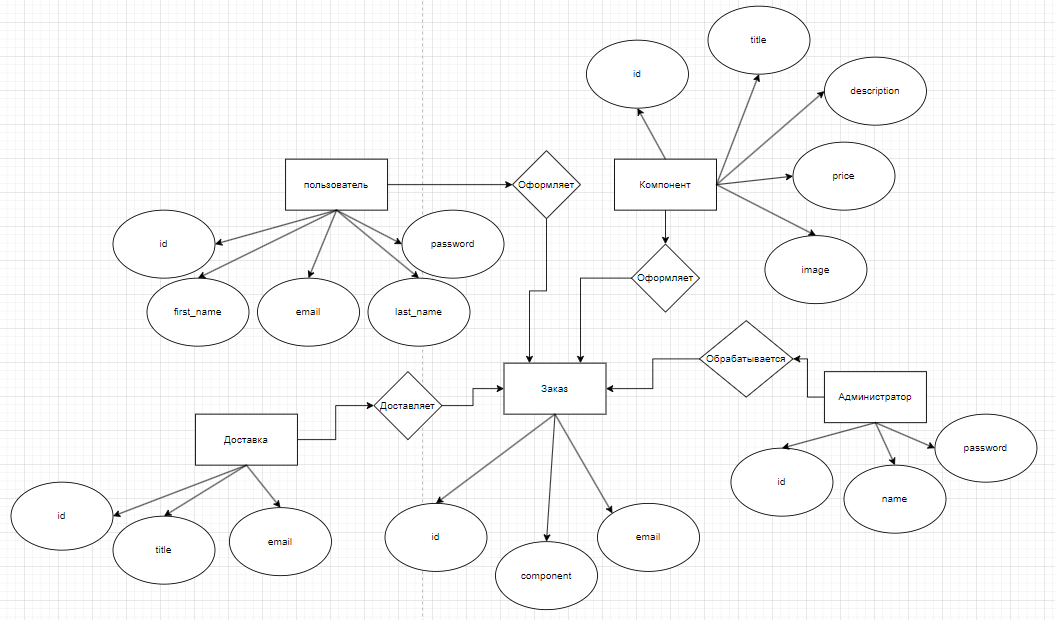


Рисунок 9 – Инфологическая модель базы данных

На инфологической модели базы данных выше схематично отображены сущности системы, их атрибуты и связи между ними. Так, в прямоугольниках отображены сущности, такие как: Пользователь, Заказ, Компонент Заказ. В овалах отображены атрибуты сущностей, например: Имя, почта, ID и другие. Ромбами изображены связи между сущностями, например, пользователь оформляет заказ. Модель содержит больше таблиц в связи с тем, что таблица users содержит как клиентов так и администраторов.

Далее происходит преобразование концептуальной модели в логическую модель, по формальным правилам. Таким образом, логическое (даталогическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных.

На даталогической модели базы данных (рисунок 10) отображены сущности приложения, а также первичные и внешние ключи, связывающие сущности между собой.

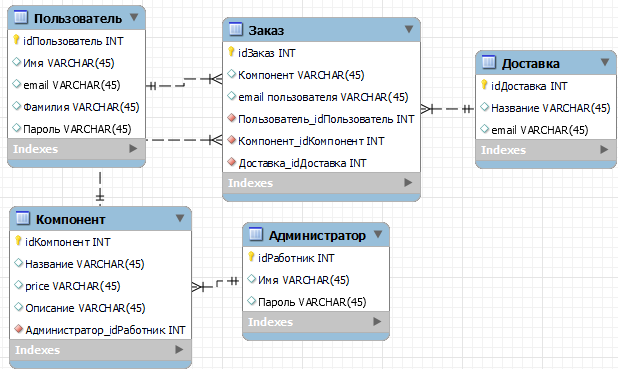


Рисунок 10 – Даталогическая модель базы данных

На рисунке 12 представлена ER-диаграмма базы данных. Она содержит 5 таблиц для полного функционирования.

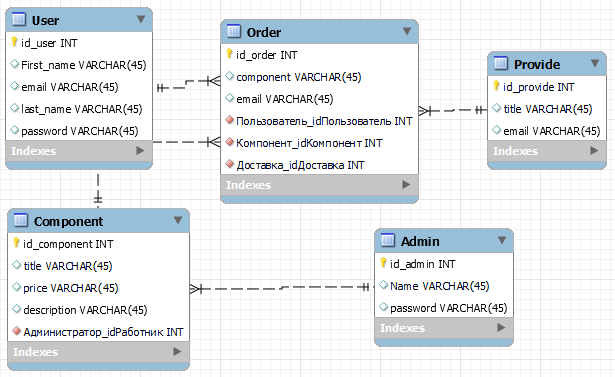


Рисунок 12 – ER-модель базы данных

С четвертой по девятую таблицу представлены основные таблицы ER-модели. В таблицах с 4 по 8 представлены поля, тип данных поля и описание.

Таблица 4 – Таблица «users»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_order | Int | ID заказа |
| component | Varchar(45) | Название товара |
| email | Int | Цена товара |

Таблица 5 – Таблица «graphics»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_component | Int | ID компонента |
| title | Varchar(45) | Название |
| price | Varchar(70) | Цена |
| description | Varchar(12) | Описание |

Таблица 6 – Таблица «genre»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_admin | Int | ID администратора |
| Name | Varchar(55) | Имя |
| Password | Varchar(55) | Пароль |

Таблица 7 – Таблица «access»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_user | Int | ID пользователя |
| First\_name | Varchar(100) | Имя |
| email | Varchar(15) | Почта |

Таблица 8 – Таблица «order»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_provide | Int | ID доставки |
| title | Varchar(55) | Название |
| email | Varchar(55) | Почта |

Для разработки даталогической и ER-модели был использован инструмент Draw.io и MySqlWorkbench.

После завершения проектирования базы данных информационной системы «Агентство по разработке игр» получится готовая схема, по которой будет разрабатываться база данных. Таким образом, представленная диаграмма демонстрирует взаимодействие основных объектов ИС и их действия.

4.4 Проектирование интерфейса

Для разработки дизайна информационной системы использовался онлайн-сервис для разработки дизайна и прототипа сайта или приложения – Figma.

В результате проектирования интерфейса будущей информационной системы были спроектированы прототипы трёх страниц: страница регистрации и страница входа (рисунок 13), а также страница с меню (рисунок 14).

Благодаря созданию прототипов, разработка непосредственно программного продукта будет значительно облегчена за счёт наглядных примеров будущих страниц ИС.

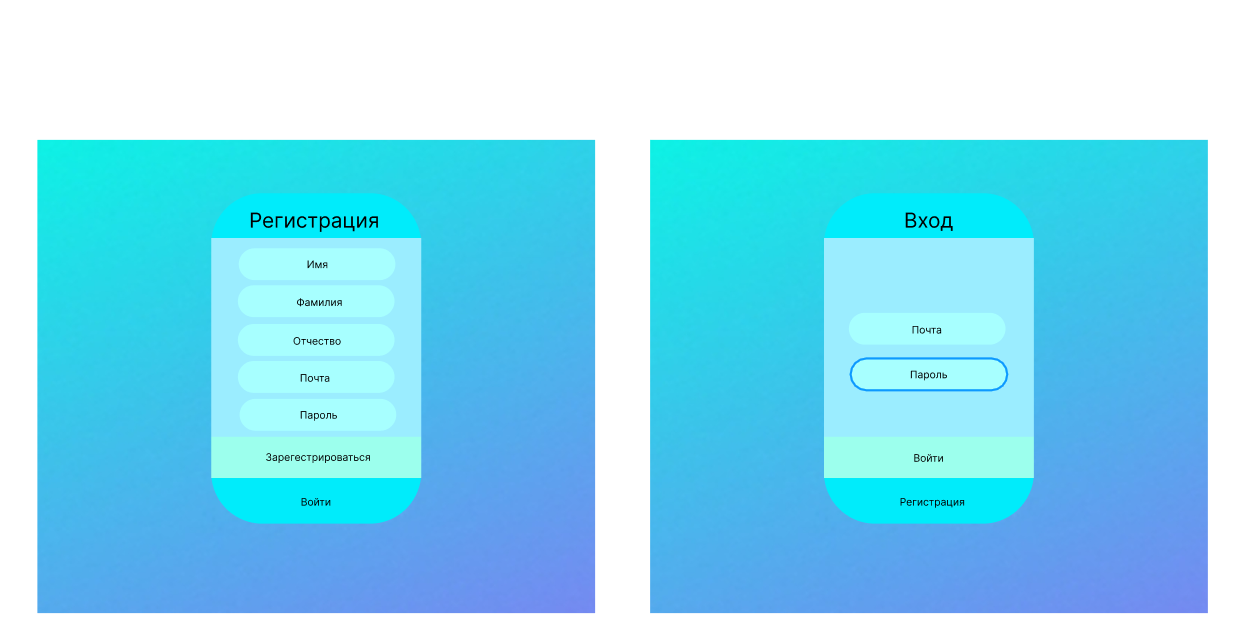


Рисунок 13 – Страницы регистрации и авторизации

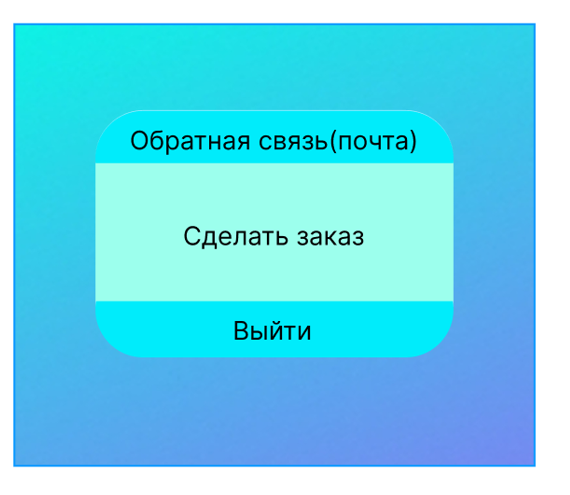


Рисунок 14 – Страница меню

5 Разработка ИС

5.1 Разработка интерфейса ИС

Разработка удобного пользовательского интерфейса – это один из важнейших этапов в процессе создания информационной системы «Агентство по разработке игр».

В разрабатываемой информационной системе будет использоваться формат - \*.php. Файл с таким форматом содержит в себе скрипт или страницу, написанную на языке “PHP”.

Вся структура работы представляет собой набор скриптовых файлов формата \*.php с возможностью переходов между ними посредством локальных ссылок, начиная с экрана авторизации и переходя последовательно к другим экранам.

С помощью PHP и CSS стилей были созданы удобные и простые окна регистрации (код показан в приложении В) и авторизации.

Регистрация и авторизация изображены на рисунках 17, 20.

HTML и CSS код страниц показан на рисунках 15, 16 и 18, 19 соответственно.



Рисунок 15 – HTML страницы авторизации пользователя



Рисунок 16 – CSS страницы авторизации пользователя



Рисунок 17 – Страница авторизации пользователя



Рисунок 18 – HTML страницы регистрации пользователя



Рисунок 19 – CSS страницы регистрации пользователя

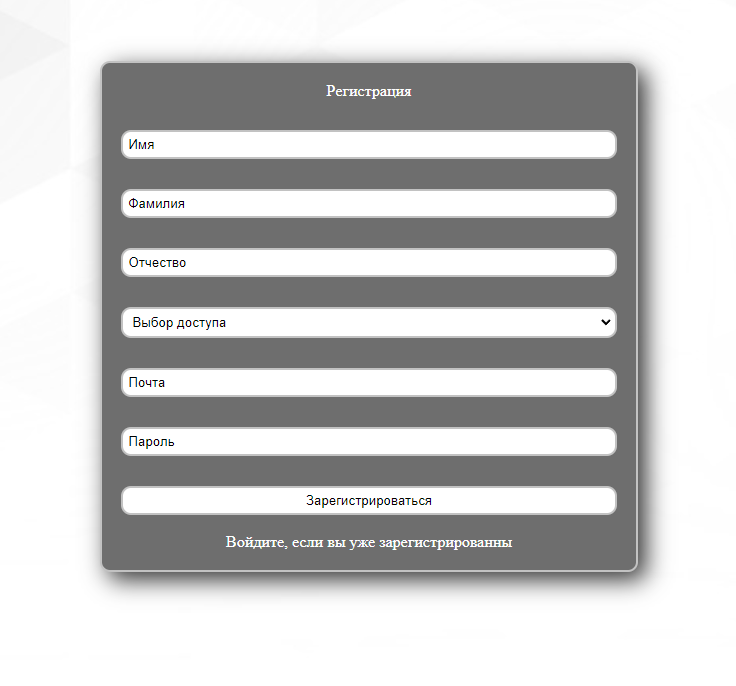


Рисунок 20 – Страница регистрации пользователя

С помощью элементов CSS и PHP были сделаны функции и интерфейс (рисунок 23 и рисунок 26). HTML и CSS этих страниц изображен на рисуках 21, 22 и 24, 25 соответственно.



Рисунок 21 – HTML страницы создания заказа

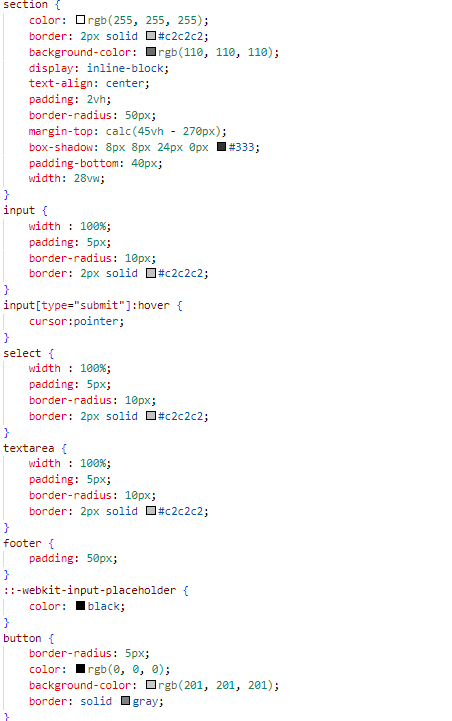


Рисунок 22 – CSS страницы создания заказа



Рисунок 23 – Страница создания заказа



Рисунок 24 – HTML страницы поиска и добавления разработчиков



Рисунок 25 – HTML страницы поиска и добавления разработчиков



Рисунок 26 – Страница поиска и добавления разработчиков

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

Все поля ввода и формы в ИС обладают проверкой вводимых данных, таким образом в случае некорректности введенных данных пользователь не получит сообщение об ошибки, а будет уведомлён о возникшем несоответствии средствами ИС. Более того, в системе предусмотрена своя система безопасности, которая предупредит пользователя, если при регистрации он использует слишком простой пароль, и попросит заменить его на более надёжный.

Информационная система доступна на любом устройстве с ОС Windows, на котором установлен любой современный браузер и есть подключение к сети Интернет. Таким образом, доступ к ИС не вызывает никаких трудностей у современного пользователя, и может быть удовлетворен без особых затрат.

Разработка ИС началась с создания основного шаблона, код которого используется для создания всех следующих страниц.

5.2 Разработка базы данных ИС

В качестве базы данных для реализации программного продукта по ряду причин, описанных в разделе «Анализ инструментальных средств разработки», была выбрана база данных MYSQL. В базе данных располагается 5 таблиц (рисунок 19).

Разработка базы данных информационной системы «Агентство по разработке игр» реализовывалась с помощью инструмента MySQLWorkbench (рисунок 27).



Рисунок 27 – MySQLWorkbench и База данных

Среди них: таблица зарегистрированных заказчиков, таблица заказов, таблица разработчиков, таблица доступа, таблица графики, таблица жанров.

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. Структуры баз данных изображены на рисунках с 28 по 32.

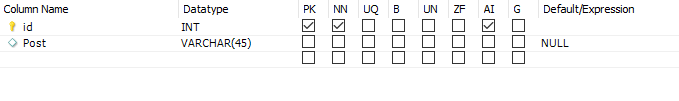


Рисунок 28 – Структура таблицы access

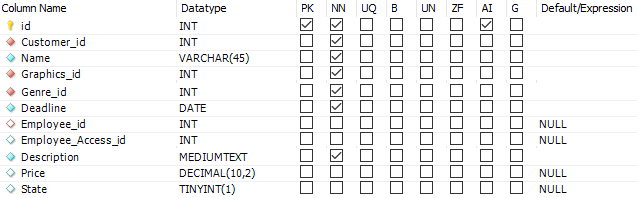


Рисунок – 29 Структура таблицы order

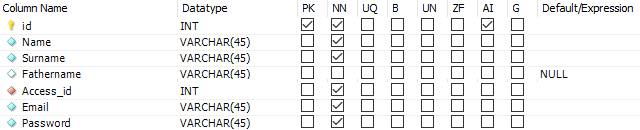


Рисунок – 30 Структура таблицы users



Рисунок – 31 Структура таблицы genre



Рисунок – 32 Структура таблицы graphics

5.3 Разработка ИС

Разработка информационной системы и подключение к базе данных (далее – БД) были выстроены в репозиторий, который содержит несколько файлов, отвечающих за работу и функционал ИС (рисунок 33).

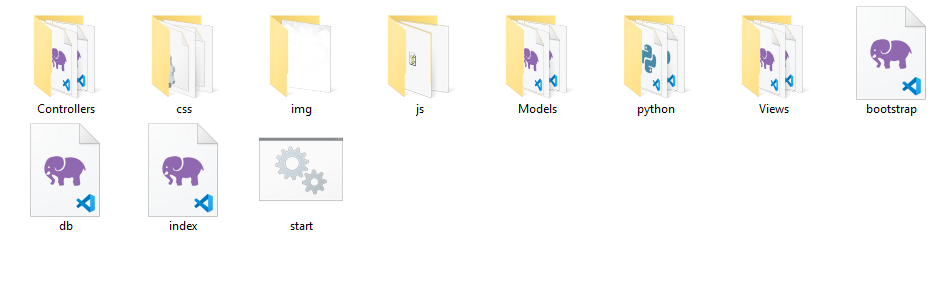


Рисунок 33 – Репозиторий информационной системы

Главный файл, c которого и начинается ход разработки к БД – это db.php.

В приложении Б представлен программный код подключения к БД из файла db.php.

Создание .php файла, отвечающего за создание заказа и получение соединения к базе данных, в котором имеется переменные $\_POST, а также функция, использующая в себе SQL запрос для добавления заказа в БД (рисунок 34) .



Рисунок – 34 Код из файла createOrder.php

Разработка файла, отвечающего за принятие решения, в котором есть окно для ввода цены и кнопка отмены заказа, и функция с SQL запросом, который выносит все заказы для оценки разрабочиком (рисунок 35).



Рисунок – 35 Код из файла employeePage.php

Проектирование завершения заказа и вывода отчёта на печать с использованием JS (рисунок 36). С помощью функции, включающей в себя SQL запрос, производится вывод всех принятых заказов (рисунок 37).



Рисунок – 36 Код из файла print.js



Рисунок – 37 Код из файла acceptedPage.php

Проектирование создания и изменения разработчика через панель администратора с использованием переменной $\_SESSION (рисунок 38), в которой содержится дата, полученная при помощи SQL запроса из файла searchEmployee.php (рисунок 39).



Рисунок – 38 Код из файла managePage.php

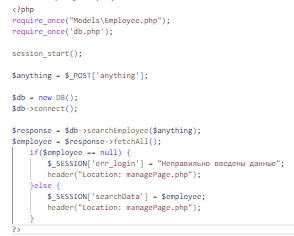


Рисунок – 39 Код из файла searchEmployee.php

Создание страницы, где реализована возможность изменения и удаления разработчиков (рисунок 40). Удаление и изменение происходит при помощи функций, содержащих в себе SQL запросы (рисунок 41 и рисунок 42).



Рисунок – 40 Код из файла editEmployee.php

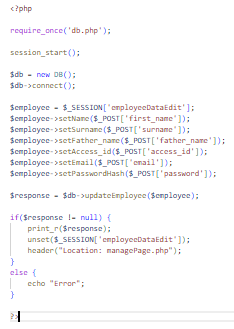


Рисунок – 41 Код из файла modifyEmployee.php



Рисунок – 42 Код из файла deleteEmployee.php

Спроектирована возможность добавления разработчиков при помощи SQL запроса, который содержится в функции insertEmployee() (рисунок 43).

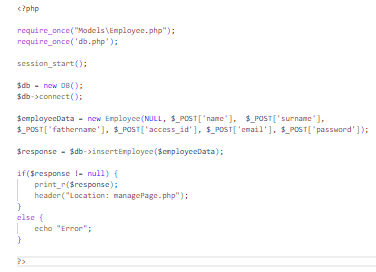


Рисунок – 43 Код из файла addEmployee.php

# 6 Тестирование ИС

Этапом разработки любого программного продукта является тестирование и отладка.

Тестирование – это процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Главная цель тестирования ПП – нахождение ошибок.

Отладка – это этап, на котором обнаруживают, локализуют и устраняют ошибки. Отладка проводилась в процессе написания программного кода, путём проверки кода в Visual Studio Code.

Данный программный продукт тестировался методом модульного тестирования.

Для тестирования ИС был разработан сценарий тестирования для каждой роли (заказчик, разработчик, администратор) предметной области.

В таблицах 9 и 10 представлены сценарии для роли Заказчик.

Таблица 9 – Сценарий тестирования создания заказа

|  |  |
| --- | --- |
| Даты теста | 27.10.23 - 27.10.23 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Создание заказа |
| Этапы теста | 1.Авторизация пользователя с ролью заказчик 2.Запись данных в поля заказа 3.Нажатие кнопки «отправить» |
| Тестовые данные | Данные заказа |
| Ожидаемый результат | Заказ отправлен разработчику |
| Фактический результат | Соответствует ожидаемому |

Таблица 10 – Сценарий тестирования редактирования профиля

|  |  |
| --- | --- |
| Даты теста | 27.10.23 - 27.10.23 |
| Приоритет тестирования | Средний |
| Заголовок/название теста | Редактирование профиля |
| Этапы теста | 1.Открытие профиля 2.Редактирование данных 3.Нажатие кнопки «сохранить» |
| Тестовые данные | Новые данные пользователя |
| Ожидаемый результат | Данные корректно изменены |
| Фактический результат | Соответствует ожидаемому |

В таблицах 11 и 12 представлены сценарии для роли Разработчик.

Таблица 11 – Сценарий тестирования одобрения заказа

|  |  |
| --- | --- |
| Даты теста | 27.10.23 - 27.10.23 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Одобрение заказа |
| Этапы теста | 1.Авторизация пользователя с ролью разработчик 2.Указание нужной цены 3.Нажатие кнопки «принять» |
| Тестовые данные | Цена заказа |
| Ожидаемый результат | Заказ получил статус принят и цена идентична указанной |
| Фактический результат | Соответствует ожидаемому |

Таблица 12 – Сценарий тестирования отклонения заказа

|  |  |
| --- | --- |
| Даты теста | 27.10.23 - 27.10.23 |
| Приоритет тестирования | Средний |
| Заголовок/название теста | Отклонение заказа |
| Этапы теста | 1.Авторизация пользователя с ролью разработчик 2.Нажатие кнопки «отклонить» у нужного заказа |
| Тестовые данные | - |
| Ожидаемый результат | Заказ удален из БД |
| Фактический результат | Соответствует ожидаемому |

В таблицах 13 и 14 представлены сценарии для роли Администратор.

Таблица 13 – Сценарий тестирования регистрации нового пользователя

|  |  |
| --- | --- |
| Даты теста | 27.10.23 - 27.10.23 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Регистрация нового пользователя |
| Этапы теста | 1.Авторизация пользователя с ролью главный разработчик 2.Открытие панели управление пользователями 3.Ввод данных о пользователе и указание его роли 4.Нажатие кнопки «добавить» |
| Тестовые данные | Данные пользователя, роль |
| Ожидаемый результат | Пользователь добавлен в систему |
| Фактический результат | Соответствует ожидаемому |

Таблица 14 – Сценарий тестирования просмотра отчётов по разработчикам

|  |  |
| --- | --- |
| Даты теста | 27.10.23 - 27.10.23 |
| Приоритет тестирования | Низкий |
| Заголовок/название теста | Просмотр отчётов по разработчикам |
| Этапы теста | 1.Авторизация пользователя с ролью главный разработчик 2.Открытие панели отчётов по разработчикам |
| Тестовые данные | - |
| Ожидаемый результат | Корректно вывелись все данные о разработчиках и количестве выполненных заказов |
| Фактический результат | Соответствует ожидаемому |

Кроме того, в рамках тестирования созданы чек-листы (таблицы 15, 16, 17).

Таблица 15 – Чек-лист тестирования функций заказчика

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тест | Входные данные | Ожидаемый результат | Фактический результат | Результат тестирования | Комментарий |
| Регистрация | Данные пользователя:  ФИО, почта, пароль | Данные пользователя добавлены в систему | Данные пользователя добавлены в систему | Неуспешно | - |
| Авторизация | Данные пользователя:  Почта, пароль | Пользователь успешно вошел в систему | Пользователь отклонен от входа в систему | Успешно | При корректно введенных данных, пользователь не смог войти в систему |
| Создание заказа | Название, тип графики, жанр, описание, дата окончания | Заказ добавлен в систему | Заказ добавлен в систему | Неуспешно | - |

Таблица 16 – Чек-лист тестирования функций разработчика

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тест | Входные данные | Ожидаемый результат | Фактический результат | Результат тестирования | Комментарий |
| Указание цены к принятому заказу | Корректная цена | У заказа в системе изменилась цена на введенную | У заказа в системе изменилась цена на введенную | Неуспешно | - |
| Отклонение заказа | - | Заказ удален из системы | Заказ удален из системы | Неуспешно | - |

Таблица 17 – Чек-лист тестирования функций администратора

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тест | Входные данные | Ожидаемый результат | Фактический результат | Результат тестирования | Комментарий |
| Добавление нового пользователя | Данные пользователя:  ФИО, уровень доступа, логин, пароль | Пользователь добавлен в систему | Пользователь не добавлен в систему | Успешно | Пользователь в введенными данными отсутствует в системе |
| Отклонение заказа | - | Заказ удален из системы | Заказ удален из системы | Неуспешно | - |

# 6.1 Модульное тестирование

Модульное тестирование — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.

Были разработаны тесты функций добавления пользователя и обновления данных о пользователе.

Было проведено тестирование добавления пользователя.

Функции, проверяющие количество пользователей до тестирования функции и во время тестирования, представлены на рисунках 44 и 45.

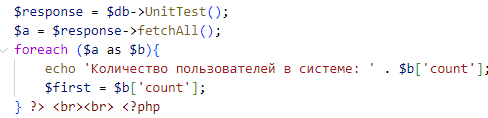


Рисунок 44 – Первое количество

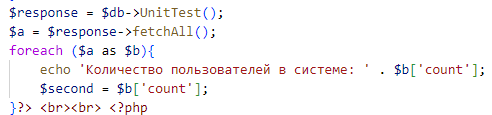


Рисунок 45 – Второе количество

На рисунке 46 изображена часть кода модульного тестирования добавления пользователя.



Рисунок 46 – Тестирование функции

Далее проводится сравнение количества пользователей до использования функции и после (рисунок 47) и результат выводится на экран (рисунок 48).

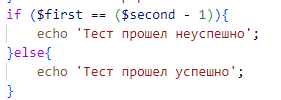


Рисунок 47 – Сравнение результатов

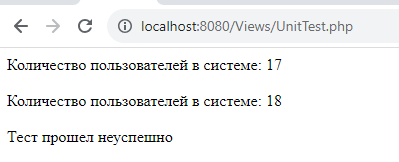


Рисунок 48 – Результат теста

Было проведено тестирование обновления данных о пользователе.

Функции, проверяющие данные об изменяемом пользователе до тестирования функции и во время тестирования, представлены на рисунках 49 и 50.

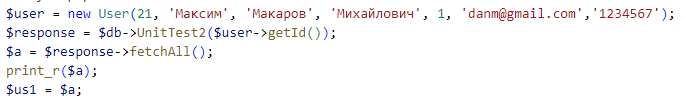


Рисунок 49 – Первые данные

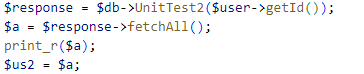


Рисунок 50 – Вторые данные

На рисунке 51 изображена часть кода модульного тестирования изменения пользователя.



Рисунок 51 – Тестирование функции

Далее проводится сравнение количества пользователей до использования функции и после (рисунок 52) и результат выводится на экран (рисунок 53).

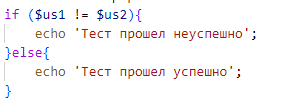


Рисунок 52 – Сравнение результатов

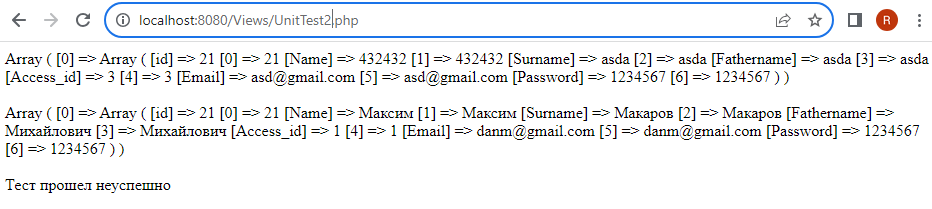


Рисунок 53 – Результат теста

После проведения модульного тестирования был сделан вывод, что функции добавления и изменения пользователей работают корректно.

# 6.2 Автоматическое тестирование

Автоматическое тестирование — процесс, представляющий собой имитацию интерактивных действий пользователя и проверку результатов этих действий.

Был использован инструмент для автоматизации действий веб-браузера Selenium.

Selenium – это набор программ с открытым исходным кодом, которые применяют для тестирования веб-приложений и администрирования сайтов локально и в сети.

С помощью автоматического тестирования был протестирован вход за Администратора (рисунок 54, 55, 56).

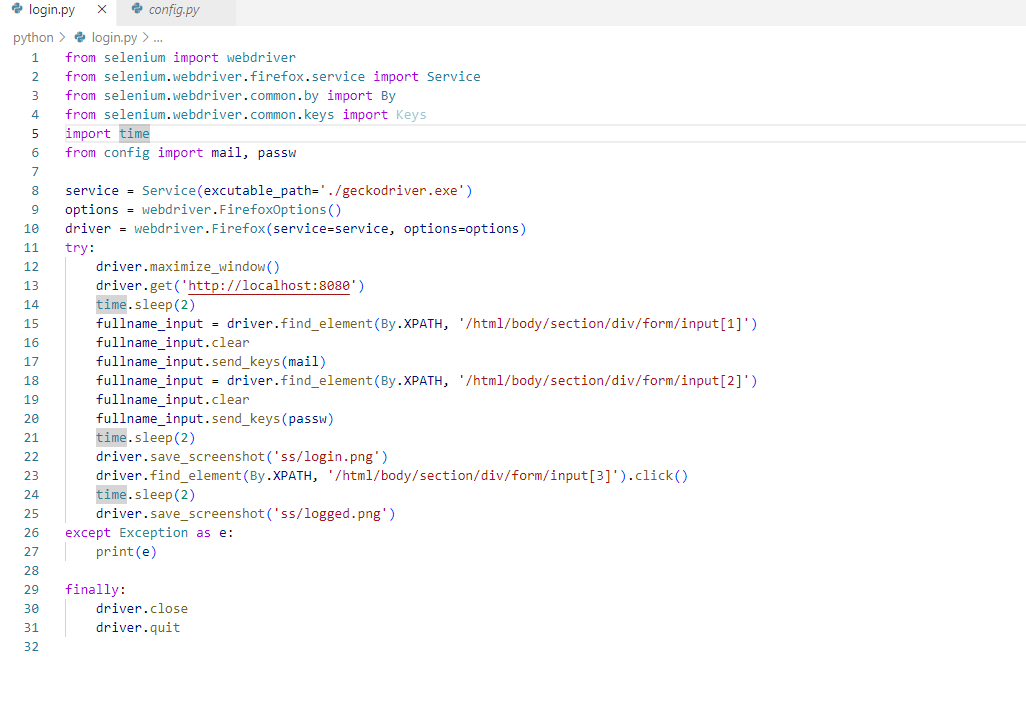


Рисунок 54 – login.py



Рисунок 55 – login.png

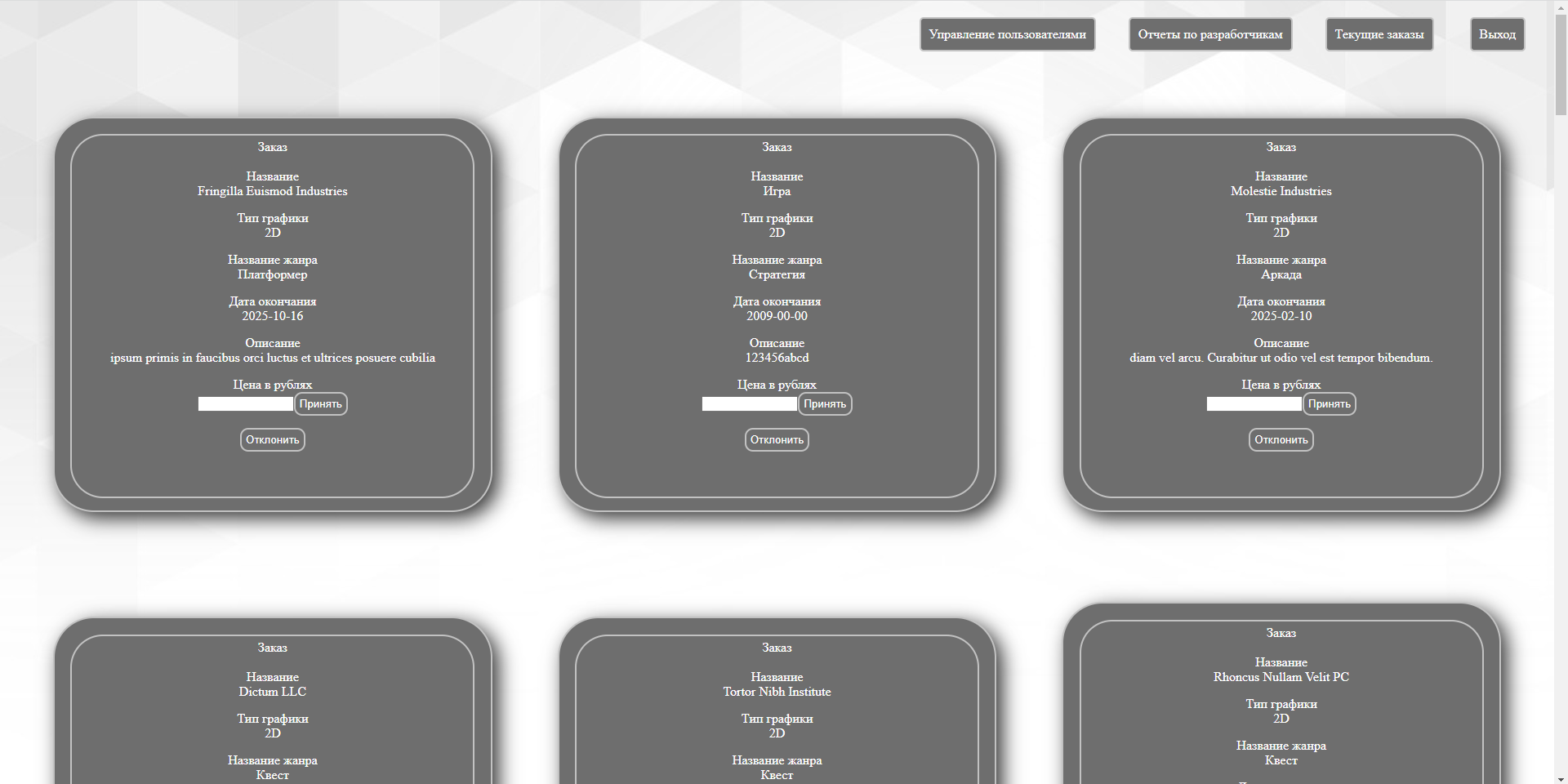


Рисунок 56 – logged.png

В ходе автоматического тестирования был протестирован вход в ИС с ролью администратора. Тест прошел неуспешно.

# 6.3 Тестирование базы данных

Также, для тестирования функциональности базы данных использовался подход CRUD.

Определение CRUD в программировании CRUD означает создание, чтение, обновление и удаление. Это четыре основные функции постоянного хранения. Они часто выполняются в программных приложениях через формы.

* + Создание - ВСТАВИТЬ запись в базу данных.
  + Чтение - ВЫБРАТЬ запись из базы данных и просмотреть её.
  + Обновление - ОБНОВИТЬ запись полностью или частично.
  + Удалить - СТЕРЕТЬ запись.

Для рассмотренных выше функций в базе данных есть соответствующий запрос. На основе требований к ПО циклы CRUD могут различаться.

Были созданы запросы, которые покрывают технологию удаления, добавления и поиска в БД (рисунки с 57 по 66).

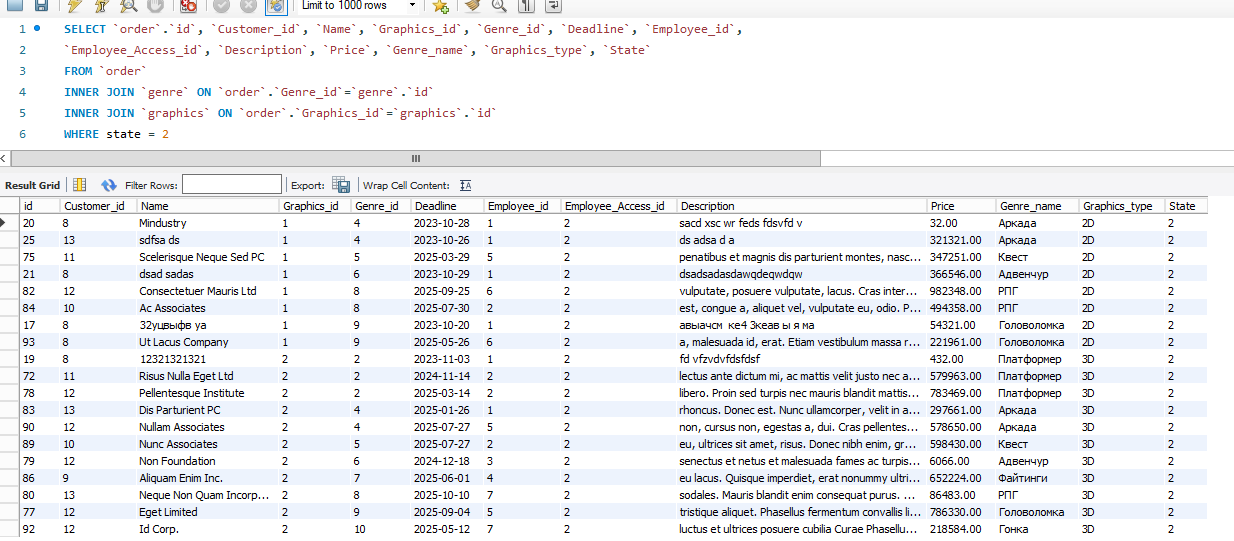


Рисунок 57 - Выборка заказов в статусе «готов»

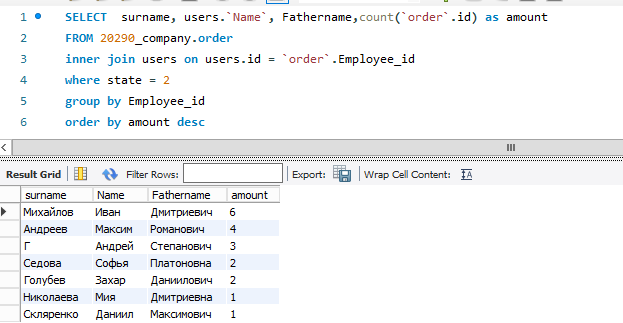


Рисунок 58 - Вывод ФИО разработчиков и количество выполненных ими заказов, отсортированных по убыванию

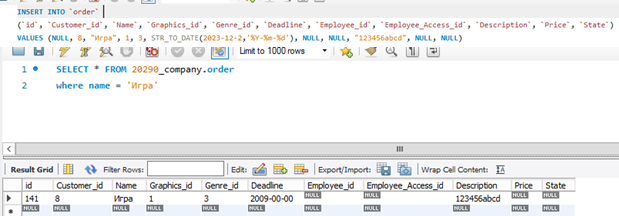


Рисунок 59 - Внесение в базу данных заказа, в котором название «Игра», заказчик под id 8, с датой окончания 2 декабря 2023 года, id графики 1, id жанра 3, c описанием «123456abcd»

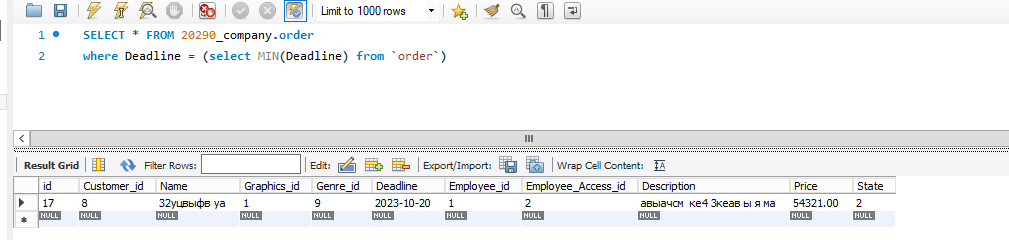


Рисунок 60 - Вывод заказа с самым ранним сроком выполнения

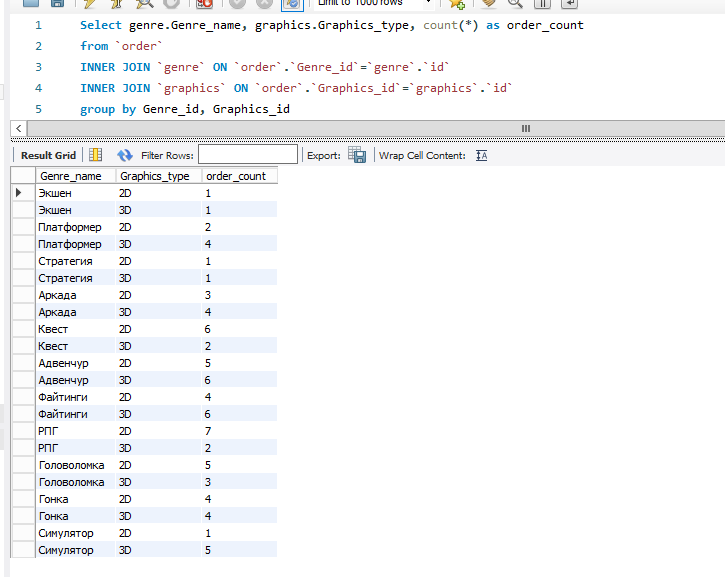


Рисунок 61 - Количество заказов в каждом жанре, выполненных в определенном графическом стиле

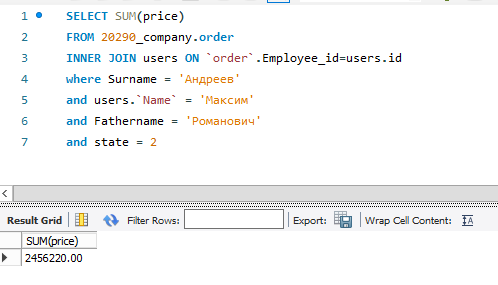


Рисунок 62 - Сумма цен готовых заказов, разработчиком которых был Андреев Максим Романович

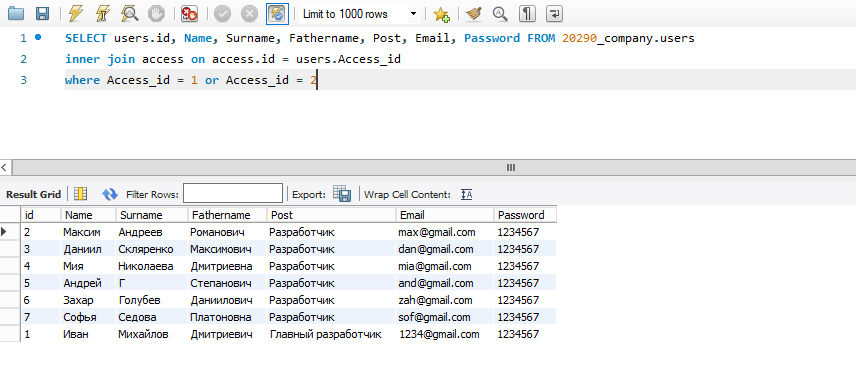


Рисунок 63 - Выборка только разработчиков

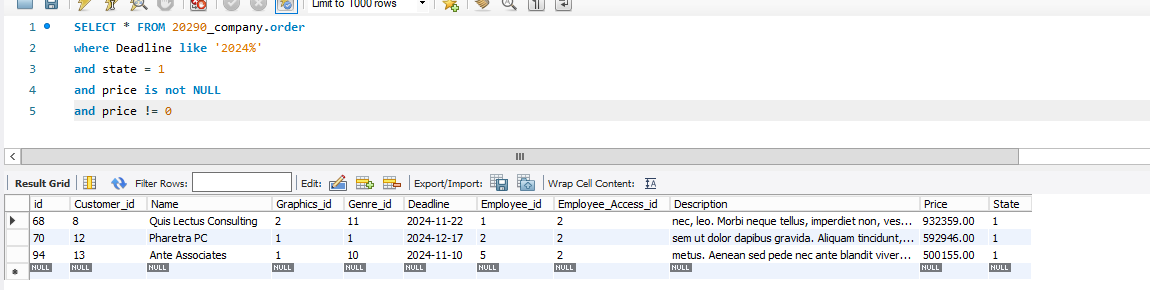


Рисунок 64 - Вывести заказы, которые будут выполнены в 2024 году

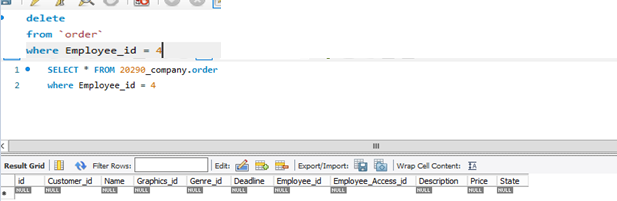


Рисунок 65 - Удалить все заказы от разработчика под id = 4

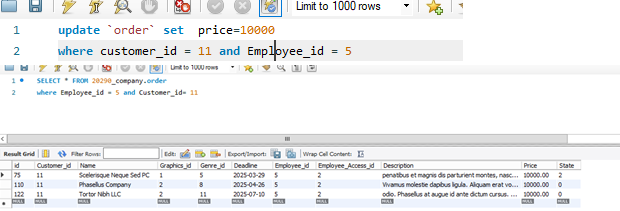


Рисунок 66 - Изменить цены у заказов, где id заказчика 11 и id разработчика 5

После тестирования базы данных методом CRUD все функции выполнились успешно и ошибок найдено не было.

7 Технологическая документация

7.1 Руководство пользователя ИС

Загрузка системы происходит из источника почтового ящика, либо же загрузчика кода gitlab, и после – разархивирование файлов на рабочий стол.

После запуска файла start.bat из разархивированной папки нужно открыть браузер и ввести в строку поиска localhost:8080, информационная система будет готова к работе.

При открытии информационной системы перед пользователем открывается страница авторизации (рисунок 67). Чтобы приступить к взаимодействию с программой и получить доступ к функциям, необходимо пройти авторизацию.



Рисунок 67 – Страница авторизации пользователя

При входе под ролью заказчика происходит переход на страницу создания заказа, где необходимо заполнить пустые поля (рисунок 68).



Рисунок 68 – Страница создания заказа

Далее, после рассмотрения заказа, разработчик указывает цену заказа (рисунок 69).

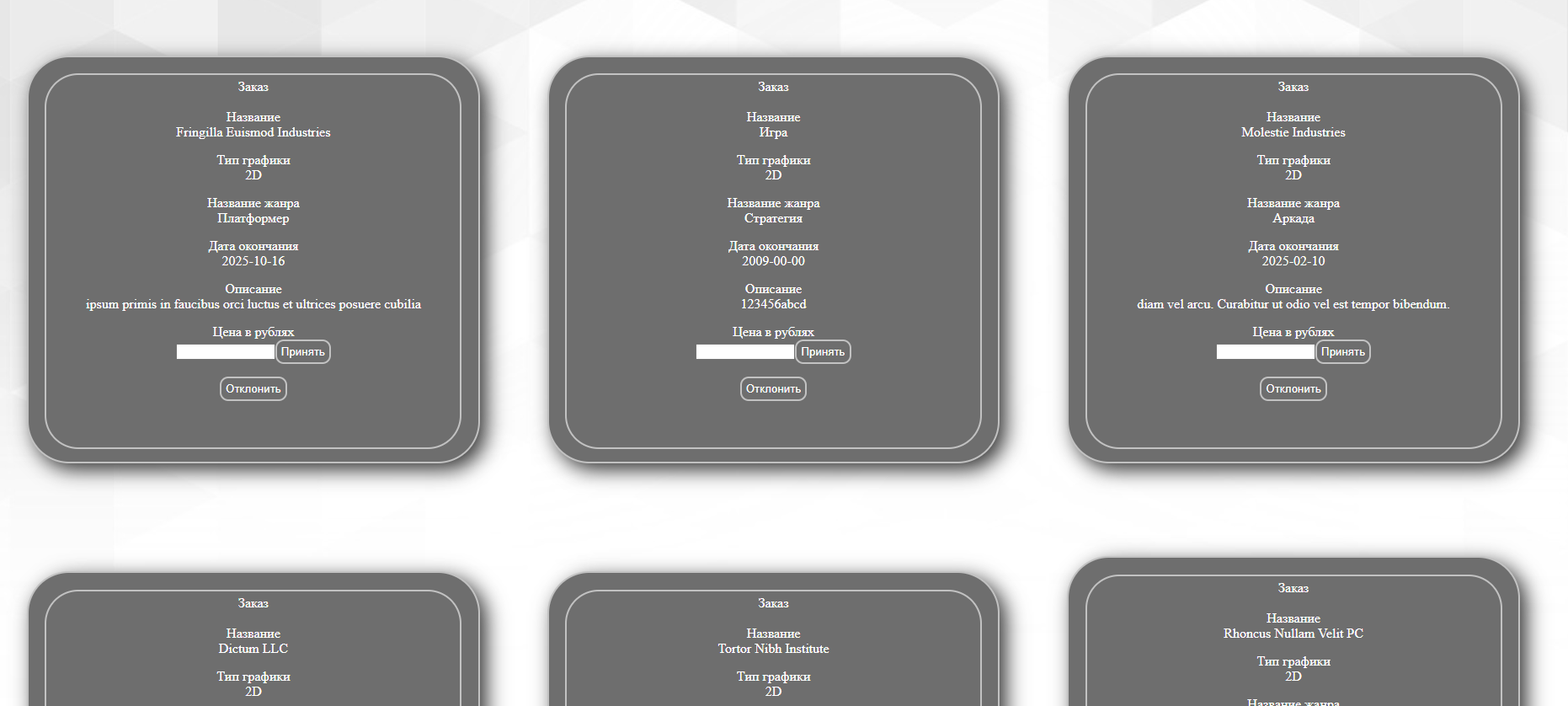


Рисунок 69 – Страница оценки заказа со стороны разработчика

И заказчику предоставляется выбор принять указанную цену, либо отказаться от заказа (рисунок 70).



Рисунок 70 – Страница принятия заказа со стороны заказчика

После принятия заказа заказчик ожидает его выполнения, и после выполнения заказа разработчик отправляет все файлы в открытом виде ему на почту.

Так же разработчик может изменять данные о других разработчиках с помощью панели управления (рисунок 71).



Рисунок 71 – Страница поиска и добавления разработчиков

ЗАКЛЮЧЕНИЕ

В ходе курсового проекта была поставлена задача разработать информационную систему «Агентство по разработке игр». Были определены требования для разработанного продукта. Исходя из требований были рассмотрены возможные реализации системы на различных языках программирования. Выбор остановлен на языке php. Разработаны ER диаграмма предметной области и шаблоны построения веб страниц, которые затем были реализованы на практике. Разработана удобная и доступная для потенциального клиента информационная система, имеющая доступные и понятные диалоговые окна интерфейса.

Разработанная ИС обеспечивает возможность выполнения следующих функций:

* авторизация;
* регистрация;
* создание заказов;
* просмотр текущих заказов;
* отмена и оценка заказов;
* администрирование пользователей.

Были выполнены цели курсового проекта – разработана информационная система «Агентство по разработке игр».

В процессе работы над проектом были решены следующие задачи:

* описана предметная область;
* выполнен полный анализ средств;
* составлено техническое задание;
* спроектирована информационная система;
* разработаны задания;
* проведено тестирование;
* создано документирование продукта.

Интерфейс программы является интуитивно понятным, требующим от пользователя самых базовых компьютерных навыков. Программа обладает удобным поиском данных о разработчиках. Данные можно отбирать по почте, фамилии или id разработчика, не зная, например, id разработчика можно ввести его почту или фамилию, и система выдаст список всех разработчиков, подходящих к параметрам поиска. Также, есть возможность распечатать информацию о готовых заказах.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Documentation: mysql\_connect . – URL: https://www.php.net/manual/ru/function.mysql-connect.php . – (дата обращения 05.10.23) – Текст: электронный.
2. Documentation: Объекты данных PHP . – URL: https://www.php.net/manual/ru/book.pdo.php – (дата обращения 05.10.23) – Текст: электронный.
3. github.com: github.com – URL: https://github.com/Cookitora/ Kursovaya/blob/main/README.md – (дата обращения: 03.10.2023) – Текст: электронный.
4. METANIT.COM: Графический клиент MySQL Workbench . – URL: https://metanit.com/sql/mysql/1.3.php?ysclid=lac1onv11d961923627 – (дата обращения 05.10.23) – Текст: электронный.
5. SQL Syntax . – URL: https://www.w3schools.com/sql/sql\_syntax.asp – (дата обращения 05.10.23) – Текст: электронный.
6. SQLite или MySQL: В чем различие SQLite или MySQL . – URL: https://ilso.ru/sqlite-vs-mysql. (дата обращения 05.10.23) – Текст: электронный.
7. Timeweb: Что такое CSS и зачем нужны каскадные таблицы стилей . – URL: https://timeweb.com/ru/community/articles/chto-takoe-css. – (дата обращения 25.10.23) – Текст: электронный.
8. Visual Studio Code Docs: PHP in Visual Studio Code . – URL: https://code.visualstudio.com/docs/languages/php asp – (дата обращения 05.10.23) – Текст: электронный.
9. Определение и особенности языка PHP: что такое PHP . – URL: https://htmlacademy.ru/php/ – (дата обращения 21.10.23) – Текст: электронный.
10. Оформление и создание веб-приложений и сайтов: Проекты . – URL: https://www.figma.com/file/NoltaosLLGvJ0QlCFWAnOs/Untitled?t=AHXzszkMvbZEtqGH-0. – (дата обращения 05.10.23) – 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

«АГЕНТСТВО ПО РАЗРАБОТКЕ ИГР»

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.П. Морозов) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Д.М. Скляренко) |

Иркутск 2023

**1 Общие сведения**

Наименование работы: Информационная система «Агентство по разработке игр».

Исполнитель: студент иркутского авиационного техникума, группы ИС-20-2, Скляренко Д.М.

Разработка информационной системы проходит в рамках курсового проекта.

Сроки разработки информационной системы с 15.09.2023 по 22.11.2023 года.

**2 Цели и назначение создания автоматизированной системы**

Целью курсового проекта является создание информационной системы «Агентство по разработке игр», которая включает в себя возможность создания заказов и редактирование данных пользователей.

В данной информационной системе реализуется следующий функционал:

− авторизация;

− регистрация;

− создание заказов;

− просмотр текущих заказов;

− просмотр отчётов;

− удаление записей.

**3 Характеристика объекта автоматизации**

Данная информационная система разрабатывается для помощи в создании или полной разработке игр. При помощи информационной системы будет проще создавать заказ, отслеживать заказы, статус заказов и параметры их выполнения.

**4 Требования к системе в целом**

**4.1 Требования к структуре и функционированию информационной системы**

Функции информационной системы:

1. Окно «Авторизации»:
   1. авторизация пользователей;
2. Окно «Регистрация»:
   1. регистрация пользователей;
3. Окно «Страница заказчика»:
   1. Создание заказа;
   2. Профиль;
4. Окно «Страница разработчика»:
   1. Указание цен на заказы;
5. Окно «Главного разработчика»:
   * 1. добавление сотрудников;
     2. удаление сотрудников;
     3. редактирование данных.

**4.2 Требования к надёжности**

Для обеспечения надёжности необходимо проверять корректность получаемых данных и реализовать валидность полей. 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 Требования к документированию**

Основным документом, регламентирующим использование информационной системы, является руководство пользователя.

Основным документом, регламентирующим разработку информационной системы, является техническое задание.

**6 Состав и содержание работ по созданию системы**

В таблице 18 представлены плановые сроки начала и окончания работы по созданию информационной системы.

Таблица 18 – 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 регистрация

<?php

require\_once("Models\User.php");

require\_once('db.php');

session\_start();

$first\_name = $\_POST['first\_name'];

$last\_name = $\_POST['last\_name'];

$father\_name = $\_POST['father\_name'];

$access = $\_POST['access'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$ALL\_OK = true;

if(strlen($first\_name) <= 1) {

$\_SESSION['err\_register\_first\_name'] = "Имя не может быть меньше двух букв";

header("Location: ../Views/registerform.php");

$ALL\_OK = false;

}

else {

unset($\_SESSION['err\_register\_first\_name']);

}

if(strlen($last\_name) <= 1) {

$\_SESSION['err\_register\_last\_name'] = "Фамилия не может быть меньше двух букв";

header("Location: ../Views/registerform.php");

$ALL\_OK = false;

}

else {

unset($\_SESSION['err\_register\_last\_name']);

}

if($access == 0) {

$\_SESSION['err\_register\_access'] = "Выберите уровень доступа";

header("Location: ../Views/registerform.php");

$ALL\_OK = false;

}

else {

unset($\_SESSION['err\_register\_access']);

}

//email

if(!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$\_SESSION['err\_register\_email'] = "Неверно введен адрес элетронной почты";

header("Location: ../Views/registerform.php");

$ALL\_OK = false;

}

else {

unset($\_SESSION['err\_register\_email']);

}

if(strlen($password) <= 6) {

$\_SESSION['err\_register\_password'] = "Пароль не может быть меньше шести символов";

header("Location: ../Views/registerform.php");

$ALL\_OK = false;

}

else {

unset($\_SESSION['err\_register\_password']);

}

if ($ALL\_OK)

{

$db = new DB();

$db->connect();

$response = $db->insertUser(new User(NULL, $first\_name, $last\_name, $father\_name, $access, $email, $password));

if($response != null) {

print\_r($response);

}

else {

echo "Error";

}

header("Location: ../index.php");

}

else

{

header("Location: ../Views/registerform.php");

}

?>