**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| КП.09.02.03.22.191.16 ПЗ |  |

ИНФОРМАЦИОННАЯ СИСТЕМА

«ВИЗАЖ»

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (О.В. Литвинова) |

Иркутск 2022

# Содержание

[Введение 3](file:///G:\КП__Ильин%20Артур.docx#_Toc89852049)

[1. Описание предметной области 4](file:///G:\КП__Ильин%20Артур.docx#_Toc89852050)

[2. Анализ инструментальных средств разработки 5](file:///G:\КП__Ильин%20Артур.docx#_Toc89852051)

[3.Техническое задание 13](file:///G:\КП__Ильин%20Артур.docx#_Toc89852052)

[4. Проектирование ИС 14](file:///G:\КП__Ильин%20Артур.docx#_Toc89852053)

[4.1. Структурная схема ИС 14](file:///G:\КП__Ильин%20Артур.docx#_Toc89852054)

[4.2. Функциональная схема ИС 16](file:///G:\КП__Ильин%20Артур.docx#_Toc89852055)

[4.3. Проектирование базы данных 20](file:///G:\КП__Ильин%20Артур.docx#_Toc89852056)

[4.4. Проектирование интерфейса 24](file:///G:\КП__Ильин%20Артур.docx#_Toc89852057)

[5. Разработка ИС 28](file:///G:\КП__Ильин%20Артур.docx#_Toc89852058)

[5.1. Разработка интерфейса ИС 28](file:///G:\КП__Ильин%20Артур.docx#_Toc89852059)

[5.2. Разработка базы данных ИС 39](file:///G:\КП__Ильин%20Артур.docx#_Toc89852060)

[5.3. Разработка ИС 39](file:///G:\КП__Ильин%20Артур.docx#_Toc89852061)

[6. Документирование программного продукта 41](file:///G:\КП__Ильин%20Артур.docx#_Toc89852062)

[6.1. Руководство пользователя ИС 41](file:///G:\КП__Ильин%20Артур.docx#_Toc89852063)

[Заключение 50](file:///G:\КП__Ильин%20Артур.docx#_Toc89852064)

[Список используемых источников 51](file:///G:\КП__Ильин%20Артур.docx#_Toc89852065)

[Приложение А – Техническое задание 52](file:///G:\КП__Ильин%20Артур.docx#_Toc89852066)

[Приложение Б – Листинг 57](file:///G:\КП__Ильин%20Артур.docx#_Toc89852067)

# Введение

Визаж – это искусство. Искусство, заключающееся не только в правильном выборе средств декоративной косметики и умелом нанесении макияжа, но и в умении с их помощью создать единый образ человека, гармонично включающий в себя и причёску, и одежду, и даже характер. Данное направление является пользуется спросом и с каждым годом приобретает все большую популярность.

Данный программный продукт (далее – ПП), будет предназначен для записи и предоставление услуг клиентам.

Актуальность данной темы состоит в том, чтобы облегчить клиентам просмотр услуг, запись и бронирование на сеанс к визажисту.

Целью курсового проекта является разработка информационной системы (далее – ИС) «Визаж», основная задача которой состоит в ведении записи клиентов к визажисту с помощью базы данных (далее – БД).

Для достижения цели, необходимо решить следующие задачи:

* исследовать выбранную предметную область ИС;
* выполнить анализ инструментальных средств разработки;
* разработать техническое задание;
* выполнить проектирование ИС;
* разработать интерфейс и структуру ИС;
* подобрать литературу и интернет-источники по выбранной теме.

1. **Описание предметной области ИС**

В качестве предметной области курсового проекта был выбран визаж.

Визаж – это целое искусство, которое позволяет меняться ежедневно, иногда до неузнаваемости, но сохраняя собственную индивидуальность. То есть не просто визуально увеличивать или уменьшать черты лица, менять прическу, но и появляться перед окружающими людьми в совершенно новом образе.

Главная задача визажа – найти наиболее привлекательные черты и сделать на них акцент, замаскировав или обыграв в выигрышном свете недостатки. Визажист — это специалист, который делает макияж средствами декоративной косметики. Хороший визажист при создании макияжа ориентируется на последние тенденции в сфере визажа, пожелания клиента, его стиль, индивидуальные особенности, а также цель макияжа.

Для грамотного визажа нужно учитывать многие факторы, а именно:

* форму лица (круглая, овальная, квадратная, треугольная);
* соотношение отдельных частей лица, их пропорциональность;
* сочетаемость оттенков различных цветов.

Чаще всего визажисты работают в салонах красоты, но также, они могут иметь собственные студии. В данном случае будет рассматриваться собственная студия. Начинающие визажисты в начале своей карьеры работают в салонах красоты, чтобы получить достаточно опыта и набрать клиентскую базу. Когда визажисты становятся более опытными, они предпочитают регистрироваться, как индивидуальные предприниматели.

Работая на себя, визажисты сами регулируют запись и цены на свои услуги.

У визажистов существует несколько разных услуг, которые могут корректироваться. У каждой услуги есть описание, в которой расписывается, что в нее входит, и стоимость, которая варьируется в зависимости от сложности работы.

Таким образом, ИС предназначена для клиентов, которые будут иметь возможность просматривать, выбирать и бронировать время на сеанс.

1. **Анализ инструментальных средств разработки**

Инструменты разработки программного продукта определяют будущий результат. Для разработки программного продукта были рассмотрены:

1. HTML.
2. CSS.
3. JavaScript.
4. MySQL.
5. SQLite.
6. PostgreSQL.
7. Python.
8. Php.
9. PyCharm.
10. PhpStorm.
11. Visual Studio.

Проектировать структуру ПП удобно через MySQL Workbench и Draw.io, а его дизайн – через онлайн-сервис Figma. ПП будет состоять из двух частей – клиентская и серверная. Для реализации клиентской части подойдут следующие инструменты: HTML, CSS, JavaScript и Bootstrap. Серверная часть будет действовать на Python-фреймворке Django со стандартной базой данных SQLite.

MySQL Workbench – инструмент для визуального проектирования [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных [MySQL](https://ru.wikipedia.org/wiki/MySQL). В проекте используется для создания ER-диаграммы БД.

Draw.io – это бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом, разработанное на HTML5 и JavaScript, используемое для создания блок-схем, UML-диаграмм, организационных и сетевых диаграмм. В проекте используется для создания прототипа страниц.

Figma – бесплатный онлайн-сервис для дизайнеров, веб-разработчиков и маркетологов. Он предназначен для создания прототипов сайтов или приложений, иллюстраций и векторной графики. Программа нового поколения для разработки интерфейсов программ, веб-сайтов и мобильных приложений. Первый облачный инструмент для разработки пользовательского интерфейса, позволяющий совместную работу в режиме реального времени.

HyperText Markup Language («язык гипертекстовой разметки», далее – HTML) – стандартизированный язык гипертекстовой разметки документов во Всемирной паутине (World Wide Web, WWW). HTML код позволяет составлять форматированный текст, а браузер, в свою очередь, интерпретирует его и отображает на экране элементы веб-страниц.

HTML – это контент и теги. Теги – это своеобразные команды, которые преобразовываются в визуальные объекты в браузере пользователя. Теги позволяют задать способ отображения контента на веб-страницах.

Достоинства HTML:

1. прост в изучении и использовании;
2. может легко интегрироваться с другими языками;
3. удобен для пользователя;
4. достаточно легко конвертируется в другие форматы;
5. может содержать мультимедиа, активное содержимое, гиперссылки.

Cascading Style Sheets («каскадные таблицы стилей», далее – CSS) – формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML).

CSS – комплекс стилевых описаний элементов HTML (тегов HTML). Они могут быть применены не только к отдельному элементу, но и ко всем идентичным тегам на всех страницах вашего сайта. По большому счету, CSS являются дополнением к HTML, значительно расширяющим его возможности.

При помощи CSS можно изменять параметры шрифтов, управлять цветовой гаммой страницы сайта, фоновыми изображениями, полями, позиционированием элементов и так далее.

Достоинства CSS:

1. универсальность: все современные браузеры способны работать с таблицами стилей практически безошибочно;
2. кэширование: CSS-файлы и параметры, единожды загруженные в браузер, используются в дальнейшем по умолчанию, не подгружаются вторично;
3. поддержка различных устройств: таблицы стилей позволяют оптимизировать внешний вид контента страницы к любому типу устройств;
4. быстрая загрузка страниц: при использовании CSS нет необходимости постоянно повторять стилевые атрибуты языка HTML.

JavaScript – язык программирования, который позволяет вам создать динамически обновляемый контент, управляет мультимедиа, анимирует изображения.

Достоинства JavaScript:

* ни один современный браузер не обходится без поддержки JavaScript;
* с использованием плагинов справится даже не специалист;
* постоянно совершенствующийся язык;

Bootstrap — это открытый и бесплатный HTML-, CSS- и JS-фреймворк, который используют веб-разработчики для быстрой верстки адаптивных дизайнов сайтов и веб-приложений. Включает в себя CSS- и HTML-шаблоны оформления для веб-форм, меток, типографики, кнопок, блоков навигации и других компонентов веб-интерфейса. Одним из основных преимуществ фреймворков Bootstrap, является то, что они ускоряют время разработки, сохраняя при этом качество и согласованность всего сайта.

Достоинства Bootstrap:

* снижение количества времени на разработку;
* легкость в использовании и открытость;
* единство стилей;
* понятный код;
* адаптивность и кроссбраузерность.

Программный продукт будет содержать в себе информацию – её необходимо хранить, изменять, структурировать и использовать. Это реализуется благодаря базе данных. Были рассмотрены такие варианты, как MySQL, SQLite, PostgreSQL

MySQL – это реляционная система управления базами данных (далее – СУБД), которая распространяется как свободное программное обеспечение. Является одной из наиболее популярных, так как отличается гибкостью, легкостью, удобством в использовании. В основном MySQL используется для работы со средними и небольшими проектами. Она имеет открытый исходный код, поэтому любой разработчик может с ним ознакомиться. Сейчас систему поддерживают известная компания Oracle.

SQLite – это быстрая и легкая встраиваемая одно файловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы. SQLite напрямую хранит информацию в одном файле, что облегчает его копирование. Большая популярность в мобильной разработке и небольших автономных приложениях, поскольку она занимает меньше места на дисковом пространстве, имеет высокую скорость работы и не требует в отличии от MySQL не требует наличие сервера для запуска.

PostgreSQL — это объектно-реляционная система управления базами данных, наиболее развитая из открытых СУБД в мире. Имеет открытый исходный код и является альтернативой коммерческим базам данных. Все таблицы представлены в виде объектов, они могут наследоваться, а все действия с таблицами выполняются с помощью объектно-ориентированных функций. Реализуются все самые новые стандарты, что приводит к ущербу простоты, из-за чего PostgreSQL очень сложный и уступает в популярности MySQL. 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 взаимосвязи баз данных и северной части продукта необходимо использовать серверный язык. Для реализации этого были рассмотрены два языка программирования – Python и Php.

Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Python поддерживает как процедурно-ориентированное, так и объектно-ориентированное программирование. В процедурно-ориентированных языках программы строятся на основе процедур или функций, которые представляют собой многократно используемые фрагменты программы. B объектно-ориентированных языках программирования программы строятся на основе объектов, объединяющих в себе данные и функционал. Python предоставляет простые, но мощные средства для ООП, особенно в сравнении с такими большими языками программирования, как C++ или Java. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов.

Django – веб-фреймворк для создания интерактивных веб-сайтов на Python. В Django вы определяете тип данных, с которыми ваш сайт должен работать, а затем указываете, как пользователи могут взаимодействовать с этими данными.

PHP (Hypertext PreProcessor) – скриптовый язык программирования. Имеет открытый исходный код. Изначально создавался для разработки веб-приложений, но в процессе обновлений стал языком общего назначения.

PHP – язык с динамической типизацией. Динамическая типизация позволяет присваивать одной и той же переменной несколько значений: она может быть массивом, строкой, числом, объектом и так далее. Динамическая типизация удобная и гибкая, но приводит к потреблению большого количества оперативной памяти и уменьшает скорость работы.

Также PHP – интерпретируемый язык. Это означает, что код, который написан на PHP, не компилируется (преобразуется в машинный), а выполняется строка за строкой с помощью программы-интерпретатора, которая читает каждую из них и выполняет то, что там указано.

Для разработки программного продукта рассмотрены интегральные среды разработки (далее – IDE) программных продуктов, такие как NetBeans, PhpStorm, Visual Studio.

PyCharm - интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик и поддерживает веб-разработку на Django.

Особенности PyCharm:

* интеллектуальный редактор кода;
* доступность инструментов интеграции;
* навигация по проекту и коду;
* рефакторинг кода.

Visual Studio – стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. IDE представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

Visual Studio представляет собой линейку продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения.

PhpStorm — инструмент для разработки PHP- и веб-проектов. В PhpStorm можно работать с самыми современными технологиями: HTML5, CSS, CoffeeScript, TypeScript, JavaScript и другими.

PhpStorm включает в себя всю функциональность WebStorm, а также полноценную поддержку PHP, баз данных и SQL.

Для выбора инструментов разработки были произведены сравнения, результат которых представлен в таблицах 1, 2, 3.

Таблица 1 – Сравнение средств реализации базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Название БД | MySQL | SQLite | PostgreSQL |
| Простота использования | - | + | - |
| Популярность | + | + | - |
| Портативность | - | + | - |
| Большое количество типов данных | + | - | + |
| Не требует удаленного сервера | - | + | - |

Таким образом, в качестве базы данных для будущего продукта была выбрана SQLite, так как она предоставляет весь необходимый функционал для разработки продукта, и при этом она проста в изучении и использовании, достаточно быстрая, предоставляет возможность гибкой настройки, а также не требует удаленного сервера.

Таблица 2 – Сравнение IDE для разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Название IDE | PyCharm | Visual Studio | PhpStorm |
| Бесплатность | -  (имеется бесплатная  лицензия) | -  (имеется бесплатная  лицензия) | -  (имеется бесплатная  лицензия) |
| Автосохранение | + | + | + |
| Автодополнение | + | + | + |
| Кроссплатформенность | + | + | + |
| Поддержка Django | + | - | - |

Таким образом, после рассмотрения сред разработок, было принято решение использовать PyCharm, так как он в сравнении с другими средами разработки поддерживает все свежие версии Django.

Таблица 3 – Сравнение языков программирования для разработки ПП

|  |  |  |
| --- | --- | --- |
| Название языка программирования | Python | Php |
| Наличие библиотек | + | + |
| Инструменты для работы с БД | + | + |
| Объектно-ориентированные возможности | + | + |
| Лёгкий и понятный синтаксис | + | - |
| Динамическая типизация | + | + |
| Более лёгкая и простая модульность | + | - |

Таким образом, Python будет более лучшим вариантом, так как он имеет большое количество библиотек и имеет инструменты для работы с БД. В сравнении с языком Php, у Python более легкий и понятный синтаксис, а также более простая работа с модулями.

Для создания ПП было решено использовать следующие средства:

1. Для создания структурных схем, контекстной диаграммы и диаграмм декомпозиции – Draw.io.
2. Для визуального проектирования [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) – MySQL Workbench.
3. Для разработки дизайна веб-приложения – Figma.
4. Для разработки программного продукта: Python, фреймворк Django, HTML5, CSS3, а также среда разработки PyCharm.
5. Для структурирования, чтения и изменения информации – SQLite.
6. **Техническое задание**

В начале разработки создавалась техническое задание, в котором указывались основные требования. Для создания технического задания использовался стандарт ГОСТ 19.

Согласно ГОСТ 19 техническое задание должно включать следующие разделы:

* + - 1. Общие сведения.
      2. Назначение и цели создания системы.
      3. Требования к системе в целом.
  1. Требования к структуре и функционированию системы.
  2. Требования к надежности.
  3. Требования к безопасности.
  4. Требования к эксплуатации, техническому обслуживанию и хранению компонентов системы.
     + 1. Требования к документированию.
       2. Состав и содержание работ по созданию системы.

Техническое задание на разработку информационной системы представлено в Приложение А.

1. **Проектирование ИС**
   1. **Структурная схема ИС**

Диаграммы прецедентов применяются для моделирования вида системы с точки зрения внешнего наблюдателя. На диаграмме описана связь групп пользователей – актёров, и функциональных возможностей – вариантов использования. На рисунке 1 представлена диаграмма прецедентов данной информационной системы, на которой показаны две группы пользователей: клиенты и сотрудники, имеющие некоторое множество вариантов использования информационной системы.



Рисунок 1 – Диаграмма прецедентов

На рисунке 2 показана диаграмма деятельности, которая отражает общую схему происходящих на предприятии бизнес-процессов. Данная диаграмма предназначена для демонстрации рабочего процесса некоторой деятельности, основанной на поэтапных действиях и действиях с поддержкой выбора. Диаграмма деятельности позволяет узнать подробности о требованиях к информационной системе.

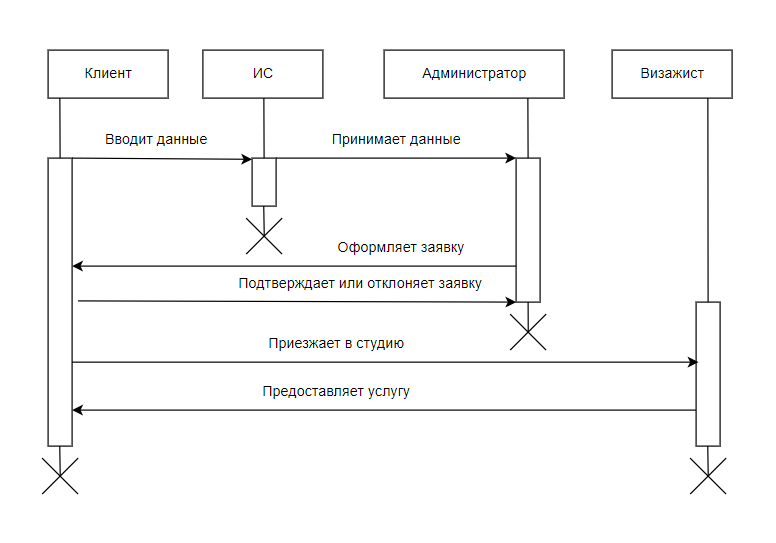


Рисунок 2 – Диаграмма деятельности

Диаграмма компонентов (рисунок 3) показывает, как компоненты соединяются вместе для формирования более крупных компонентов или программных систем, а также дает упрощенное представление о сложной системе, разбивая ее на более мелкие компоненты.

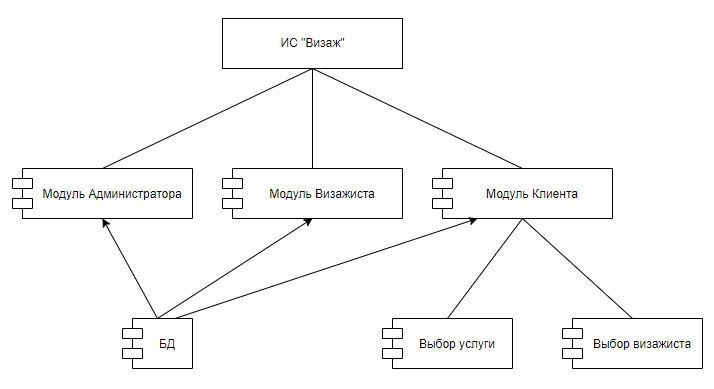


Рисунок 3 – Диаграмма компонентов

Диаграмма развертывания показывает архитектуру исполнения системы, включая в себя аппаратные или программные среды исполнения, а также промежуточное программное обеспечение, соединяющее их. Диаграммы развертывания обычно используются для визуализации физического аппаратного и программного обеспечения системы. На рисунке 4 представлена диаграмма развертывания.

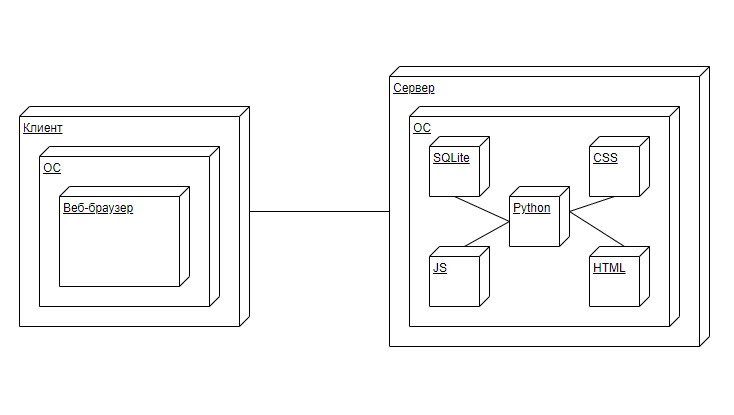


Рисунок 4 – Диаграмма развертывания

* 1. **Функциональная схема ИС**

На рисунке 5 представлена контекстная диаграмма, отображающая деятельность информационной системы.

На контекстной диаграмме показаны:

* функциональный блок: деятельность ИС «Визаж»;
* входные данные: заявка на услугу;
* управление: алгоритм работы;
* механизм: персонал;
* выходные данные: выполненная услуга.

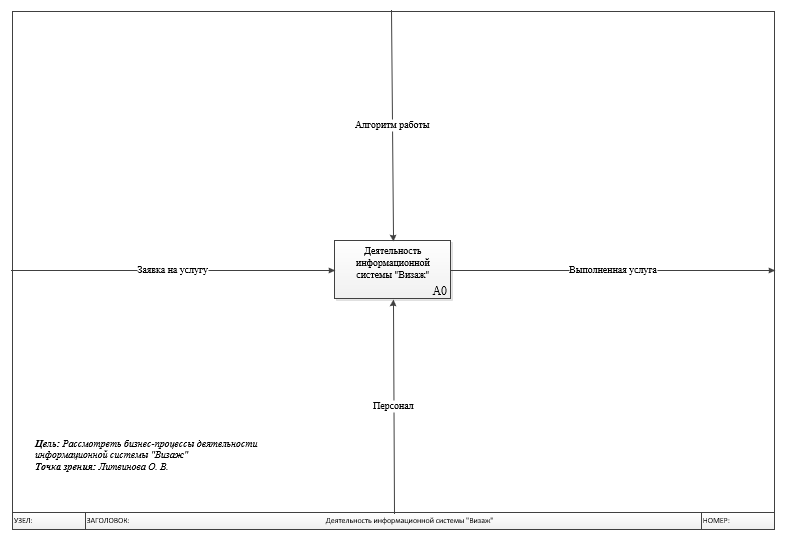


Рисунок 5 – Контекстная диаграмма

Диаграмма декомпозиции предназначена для детализации функций и получаются при разбиении контекстной диаграммы на крупные подсистемы и описывающие каждую подсистему и их взаимодействие.

Диаграмма декомпозиции показывает комплекс, процесс, организацию, предметную область данных или другой тип объекта, разбитый на более низкоуровневые, более подробные компоненты. Данные диаграммы обеспечивают логическую иерархическую декомпозицию системы.

На рисунке 6 показана диаграмма декомпозиций, которая расписывает основную функцию. На этой диаграмме показаны такие функции как: «Ввод данных», «Обработка данных», «Подтверждение заявки», «Выполнении услуги». Также, на диаграмме, изображены стрелками: входные и выходные данные, управление и механизмы.

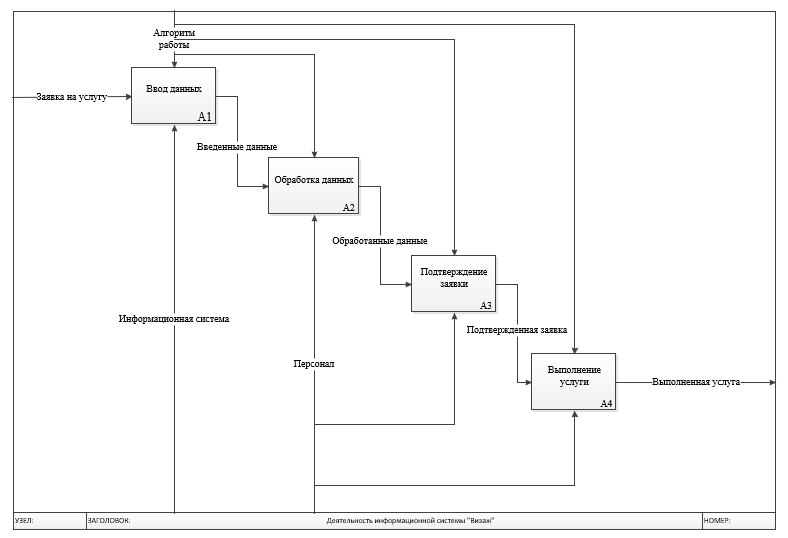


Рисунок 6 – Диаграмма декомпозиций

Диаграмма классов предназначена для представления внутренней структуры программы в виде классов и связей между ними. Все сущности должны быть представлены объектами классов в программе. При этом у каждого класса должно быть только одно назначение и уникально осмысленное имя, которое будет связано с этой целью.

Диаграммы классов можно использовать для моделирования объектов, составляющих систему, для демонстрации отношений между объектами и для описания ролей этих объектов и предоставляемых ими услуг.

На рисунке 7 представлена диаграмма классов, которая содержит 5 классов, таких как: «Клиент», «Визажист», «Администратор», «Заявка» и «Услуга».

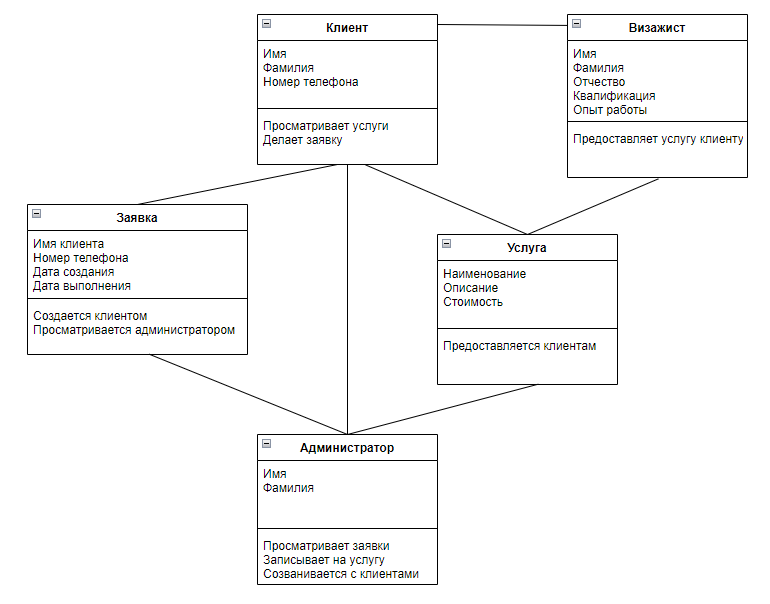


Рисунок 7 – Диаграмма классов

DFD – диаграмма потоков данных. Это метод, с помощью которого проводится графический структурный анализ, в котором описаны внешние для системы источники данных, функции, потоки и хранилища данных, к которым имеется доступ.

Диаграмма отображает потоки данных между системами, базами данных, а также логику преобразования данных в системе в каждом процессе. В данной диаграмме видны входные, промежуточные, выходные данные в каждом процессе, который протекает от внешней сущности до хранилищ данных.

Диаграмма потоков данных продемонстрирована на рисунке 8.

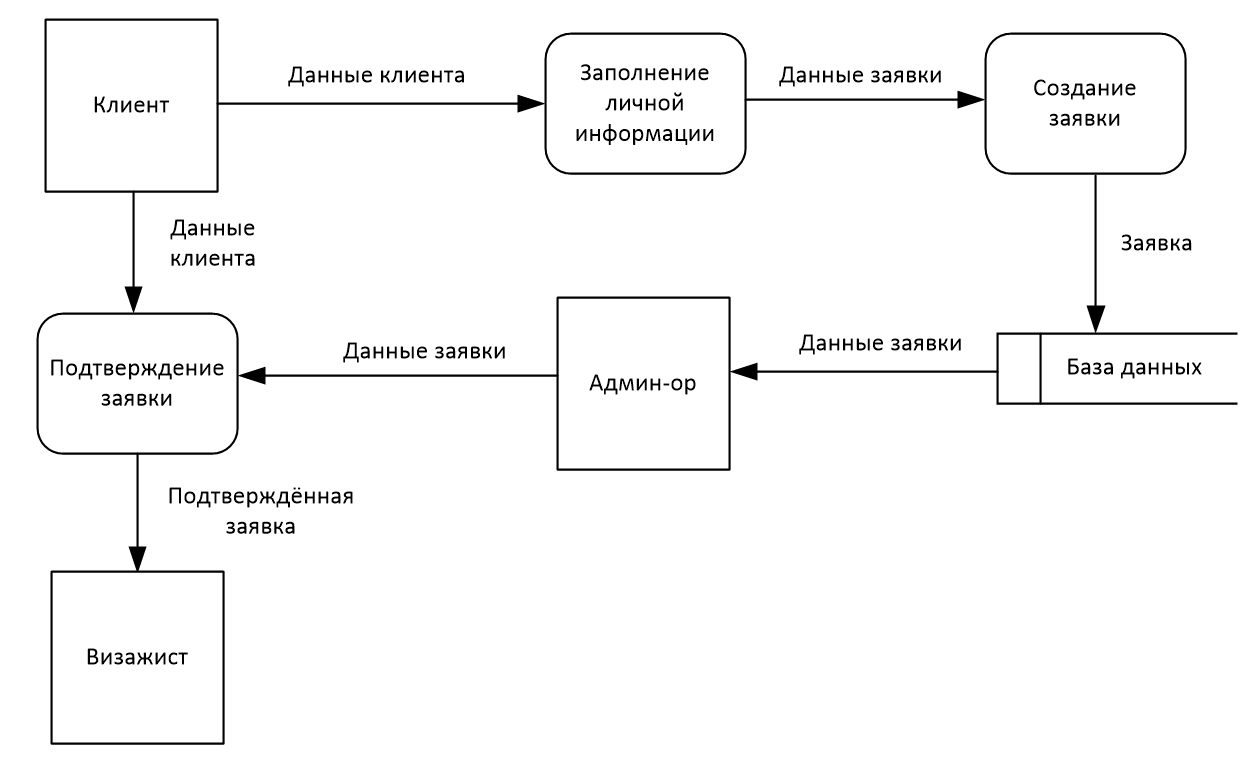


Рисунок 8 – Диаграмма потоков данных DFD

Таким образом, представленные диаграммы демонстрируют взаимодействие между основными функциями информационной системы «Визаж» и механизмами, влияющими на выполнение той или иной функции.

* 1. **Проектирование базы данных**

Цель инфологического моделирования – выявление и объединение информационных требований пользователя, связей между элементами данных без относительно к их содержанию и среде их хранения. Инфологическая модель должна обладать свойством легкой расширяемости, обеспечивающим ввод новых данных без изменений ранее определенных.

На рисунке 9 изображена инфологическая модель БД, основными конструктивными элементами которой являются сущности, связи между ними и их свойства (атрибуты).

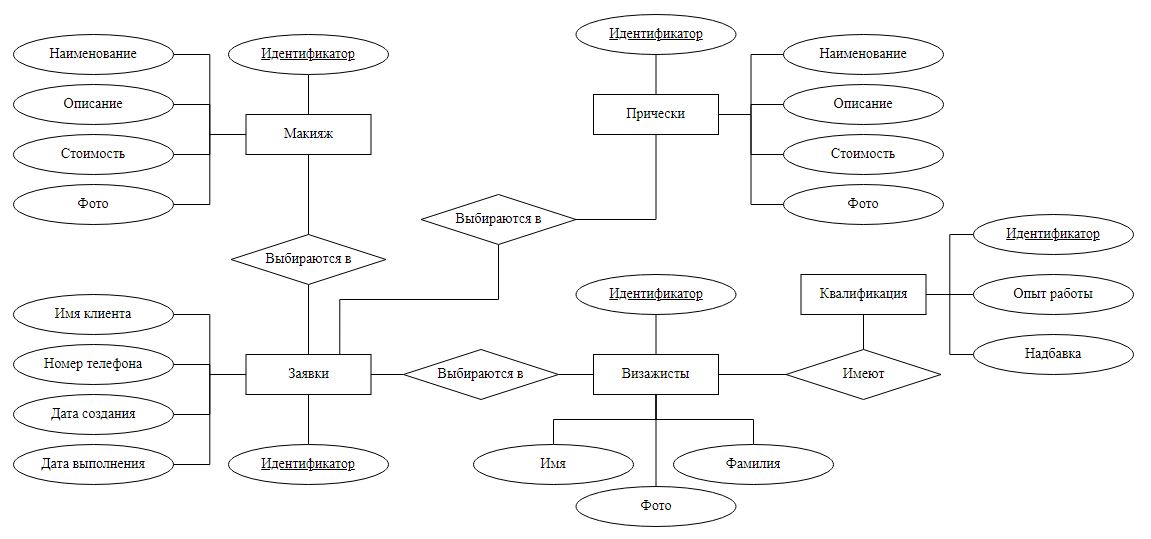


Рисунок 9 – Инфологическая модель БД

Даталогическая модель (рисунок 10) разрабатывается с учетом конкретной реализации СУБД, а также с учетом специфики конкретной предметной области на основе ее инфологической модели.

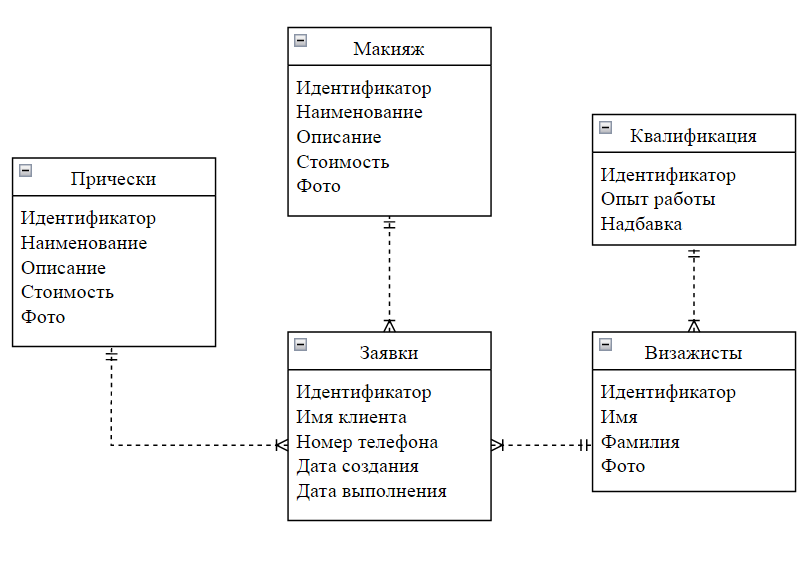


Рисунок 10 – Даталогическая модель БД

ER-модель – модель данных, которая предназначена для упрощения процесса проектирования базы данных. Из ER-модели могут быть порождены все виды баз данных: реляционные, иерархические, сетевые, объектные.

В основе ER-модели лежат понятия «сущность», «связь» и «атрибут». Диаграмма «сущность»-«связь» – это диаграмма, которая представляет в графическом виде сущности, атрибуты и связи.

ER-модель БД продемонстрирована на рисунке 11.

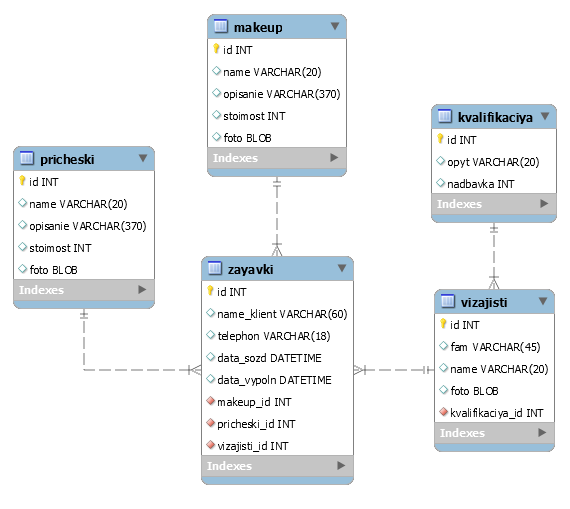


Рисунок 11 – ER-модель БД

На схеме базы данных представлены 5 таблиц: «makeup», «pricheski», «vizajisti», «kvalifikaciya», «zayavki».

Поля id являются первичными ключами своих таблицы.

БД приведена к 1 начальной форме (далее – НФ), т.к. все поля, принимаемые больше одного значения, декомпозированы. Также БД имеет 2 и 3 НФ, т.к. каждый не ключевой атрибут приведен к неприводимости, и каждый не ключевой атрибут не транзитивно (непосредственно) зависит от первичного ключа.

В таблицах 4, 5, 6, 7, 8 представлены описания полей с типами данных для каждой таблицы данной ER-модели БД.

Таблица 4 – «makeup»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | INT | Уникальный идентификатор макияжа |
| name | VARCHAR(20) | Наименование макияжа |
| opisanie | TEXT(370) | Описание макияжа |
| stoimost | INT | Стоимость макияжа |
| foto | BLOB | Фотография макияжа |

Таблица 5 – «pricheski»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | INT | Уникальный идентификатор причёски |
| name | VARCHAR(20) | Наименование причёски |
| opisanie | TEXT(370) | Описание причёски |
| stoimost | INT | Стоимость причёски |
| foto | BLOB | Фотография причёски |

Таблица 6 – «vizajisti»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | INT | Уникальный идентификатор визажиста |
| fam | VARCHAR(45) | Фамилия визажиста |
| name | VARCHAR(20) | Имя визажиста |
| kvalifikaciya\_id | INT | Идентификатор квалификации (внешний ключ) |
| foto | BLOB | Фотография визажиста |

Таблица 7 – «kvalifikaciya»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | INT | Уникальный идентификатор квалификации |
| opyt | VARCHAR(20) | Опыт визажиста |
| nadbavka | INT | Надбавка к стоимости услуг |

Таблица 8 – «zayavki»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id | INT | Уникальный идентификатор заявки |
| name\_klient | VARCHAR(60) | Имя клиента |
| telephon | VARCHAR(18) | Номер телефона клиента |
| data\_sozd | DATETIME | Дата создания заявки |
| data\_vypoln | DATETIME | Дата выполнения услуги |
| makeup\_id | INT | Идентификатор макияжа (внешний ключ) |
| pricheski\_id | INT | Идентификатор причёски (внешний ключ) |
| vizajisti\_id | INT | Идентификатор визажиста (внешний ключ) |

* 1. **Проектирование интерфейса**

Для проектирования интерфейса был выбран инструмент Figma – бесплатный онлайн-сервис для создания прототипов сайтов или приложений.

В результате проектирования интерфейса будущей информационной системы были спроектированы прототипы трёх страниц: страница «О нас» (рисунок 12), страница «Услуги» (рисунок 13), страница «Контакты» (рисунок 14).

На данных прототипах страниц также наглядно показаны навигация и футер для всех страниц будущей информационной системы.

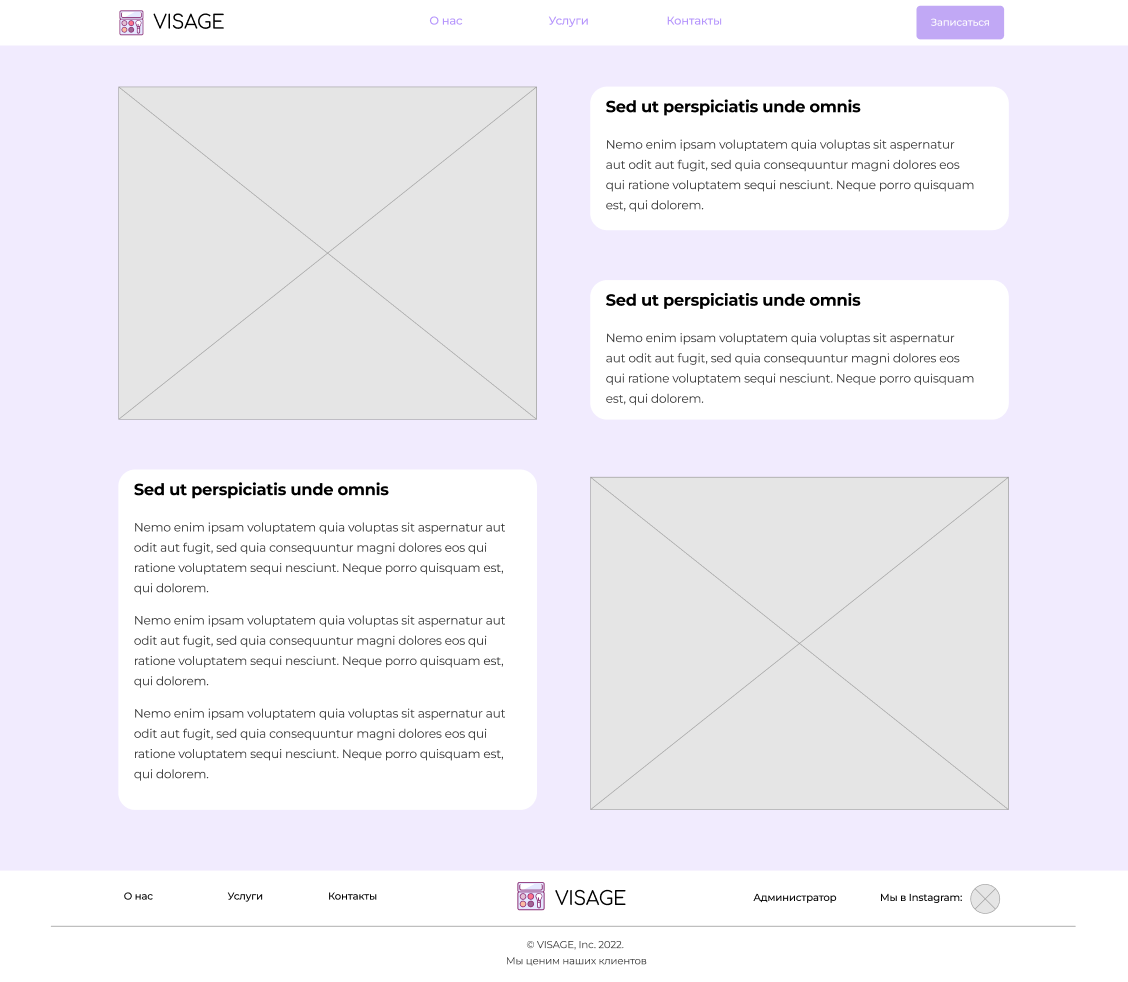


Рисунок 12 – Прототип страницы «О нас»

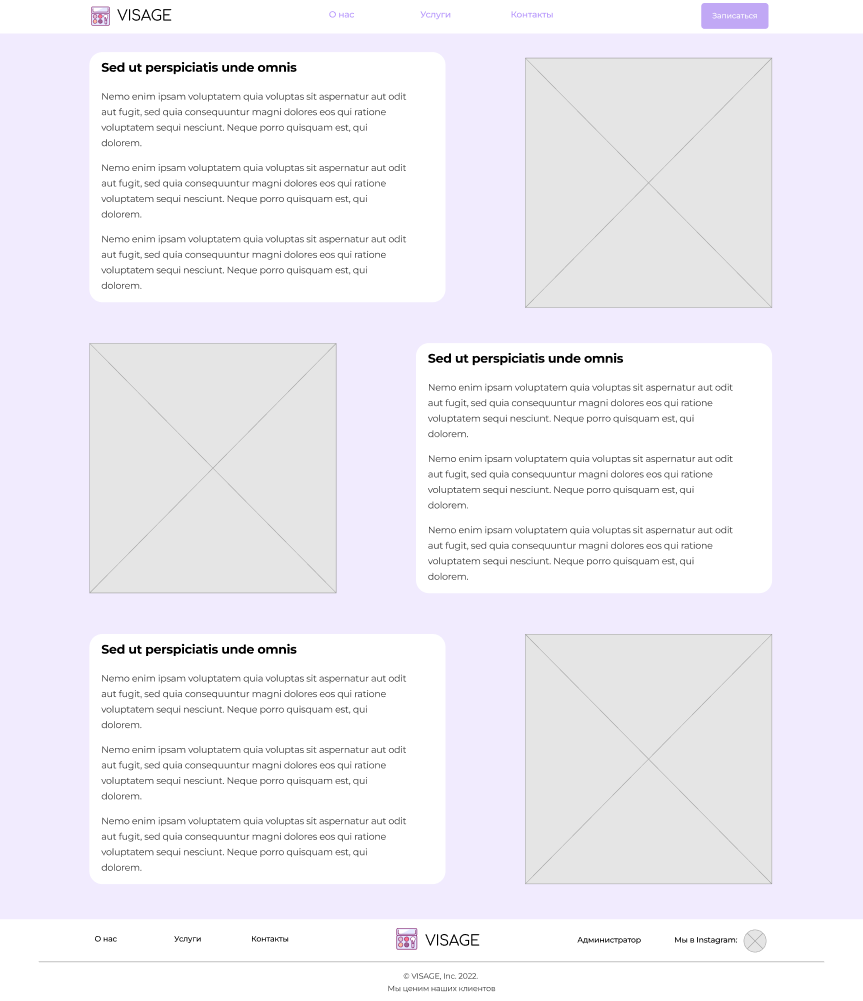


Рисунок 13 – Прототип страницы «Услуги»

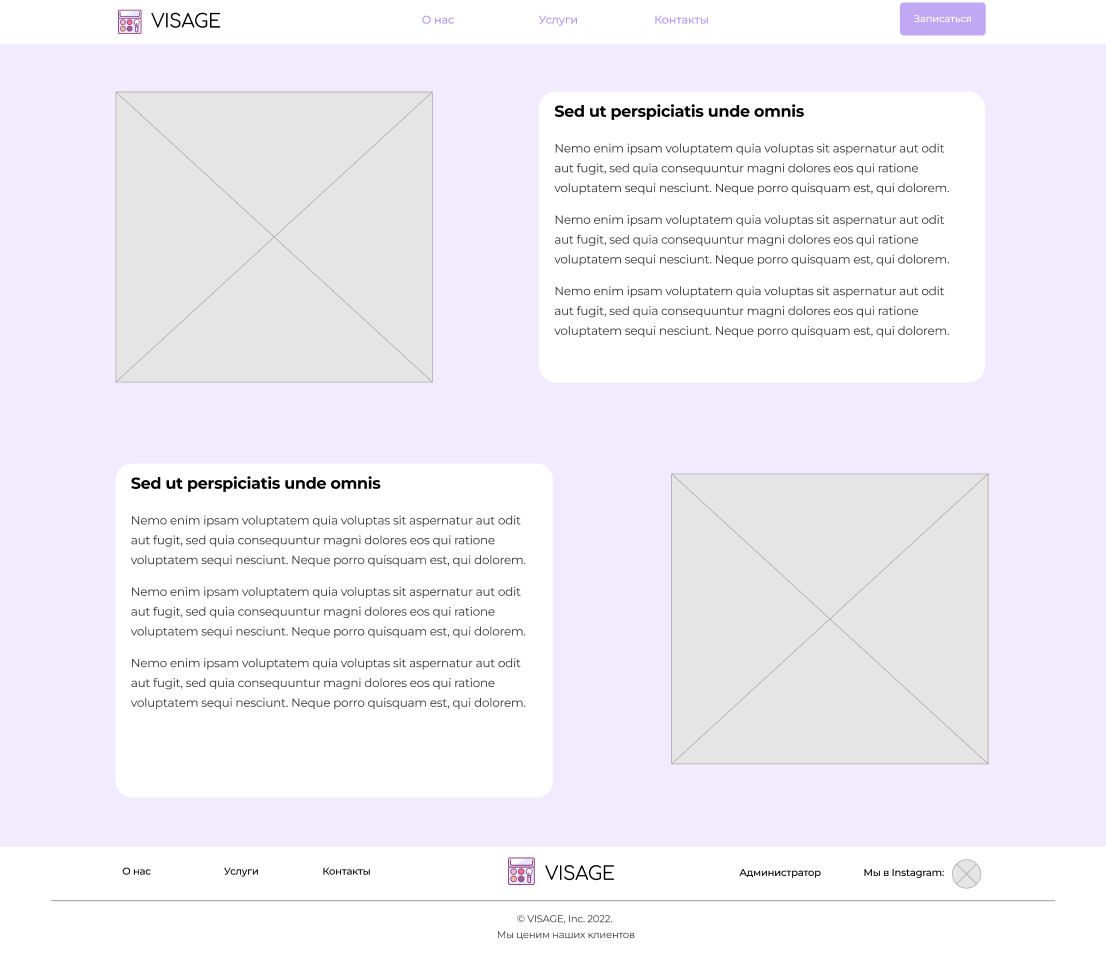


Рисунок 14 – Прототип страницы «Контакты»

Интерфейсы являются основой взаимодействия всех современных информационных систем. Благодаря созданию прототипов, разработка программного продукта будет значительно облегчена, за счёт наглядных примеров будущих страниц информационной системы.

Во время разработки прототипа интерфейса был учтен проведенный анализ аналогов, что помогло реализовать представленный вид прототипа информационной системы «Визаж».

1. **Разработка ИС**
   1. **Разработка интерфейса ИС**

Разработка удобного и простого пользовательского интерфейса – это один из важнейших этапов в процессе создания информационной системы «Визаж».

Для разработки пользовательского интерфейса были использованы такие инструменты как: HTML, CSS, Bootstrap. В разрабатываемой информационной системе файлы, содержащие описание внешнего вида страниц, представлены в формате \*.html и \*.сss.

Разработка ИС началась с создания основного шаблона, код которого используется для создания всех следующих html-страниц (рисунки 15). В состав входит верхняя навигационная панель и footer.



Рисунок 15 – Часть HTML кода основного шаблона

С помощью фреймворка Bootstrap были созданы формы, таблицы и стили кнопок, благодаря которому информационная система выглядит красочно, но, в то же время, приятна глазу.

На рисунке 16 изображена страница «Главная», на которой представлена общая информация о деятельности информационной системы «Визаж»: информация о главном визажисте и сертификате, подтверждающего его квалификацию.

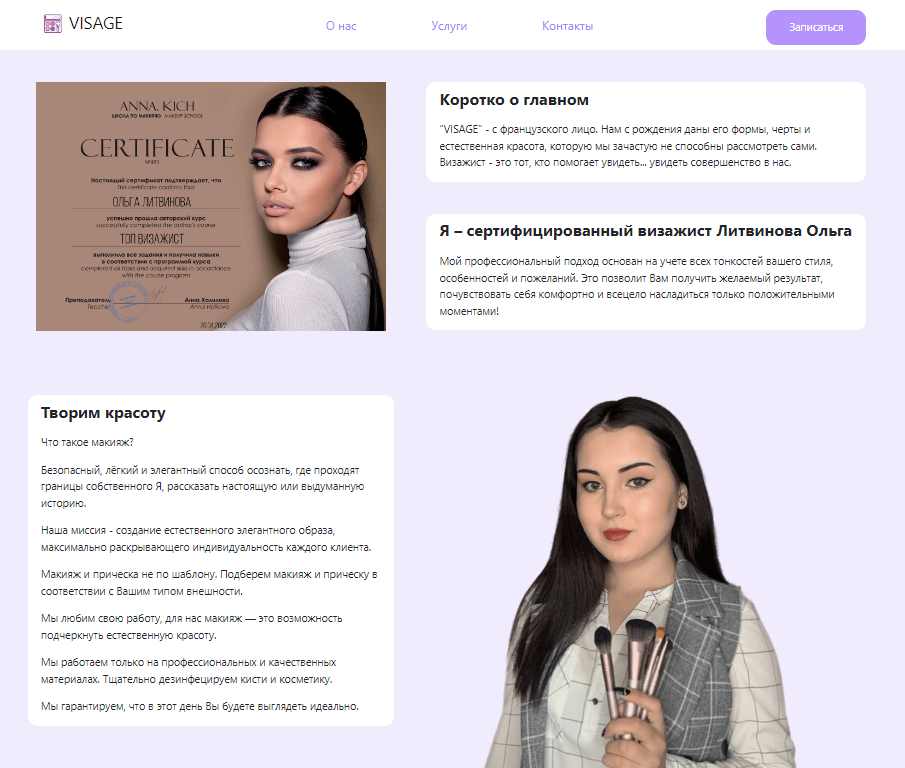


Рисунок 16 – Страница «Главная»

На странице «Услуги», показанной на рисунке 18, предоставлена информация о видах макияжа и причесок. Для каждого вида макияжа и прически предусмотрена отдельная форма, на которой расположен текст с названием, описанием и стоимостью услуги. Для каждой квалификации визажистов предусмотрена своя стоимость на каждую услугу. Также, показан пример выполнения услуги, что позволяет наглядно увидеть, как будет смотреться тот или иной вид макияжа или прически.

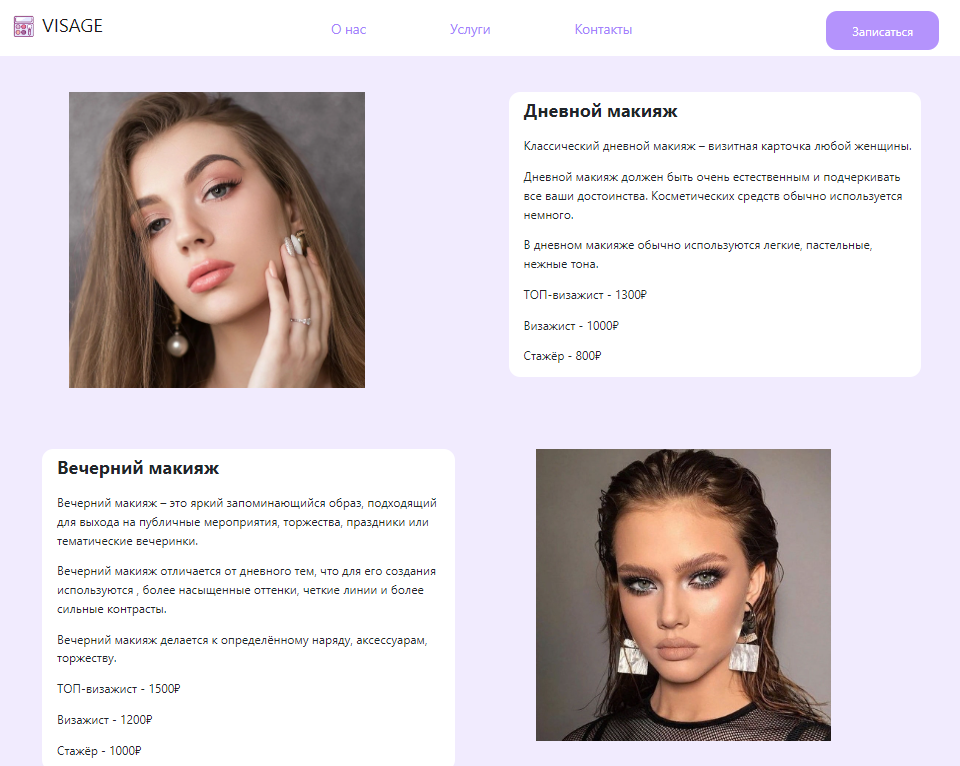


Рисунок 18 – Страница «Услуги»

На рисунке 17 представлена страница «О нас» с информацией о визажистах и их квалификациях, а также их фотографии.

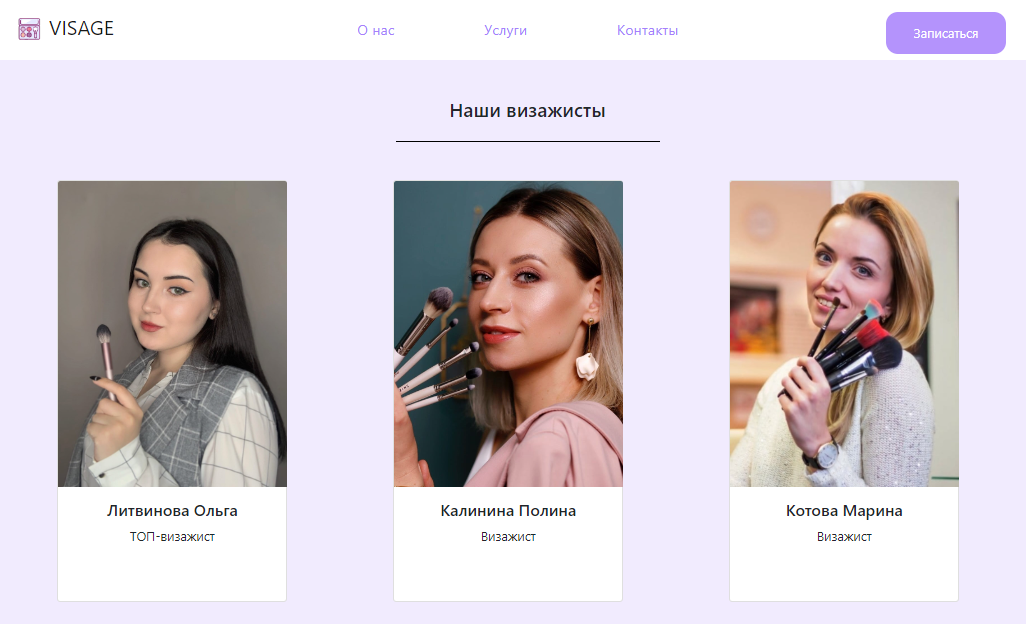


Рисунок 17 – Страница «О нас»

На странице «Контакты» представлена информация с контактными данными для связи (рисунок 19).

Для удобства клиента предусмотрена карта с графическим отображением местонахождением проведения услуг. С левой стороны от карты располагается контактная информация: номер телефона для связи, точный адрес. А также указаны ссылки на группы в социальных сетях и мессенджерах, таких как: Instagram, Вконтакте, Telegram. Вместе с тем, в контактной информации указаны часы работы.

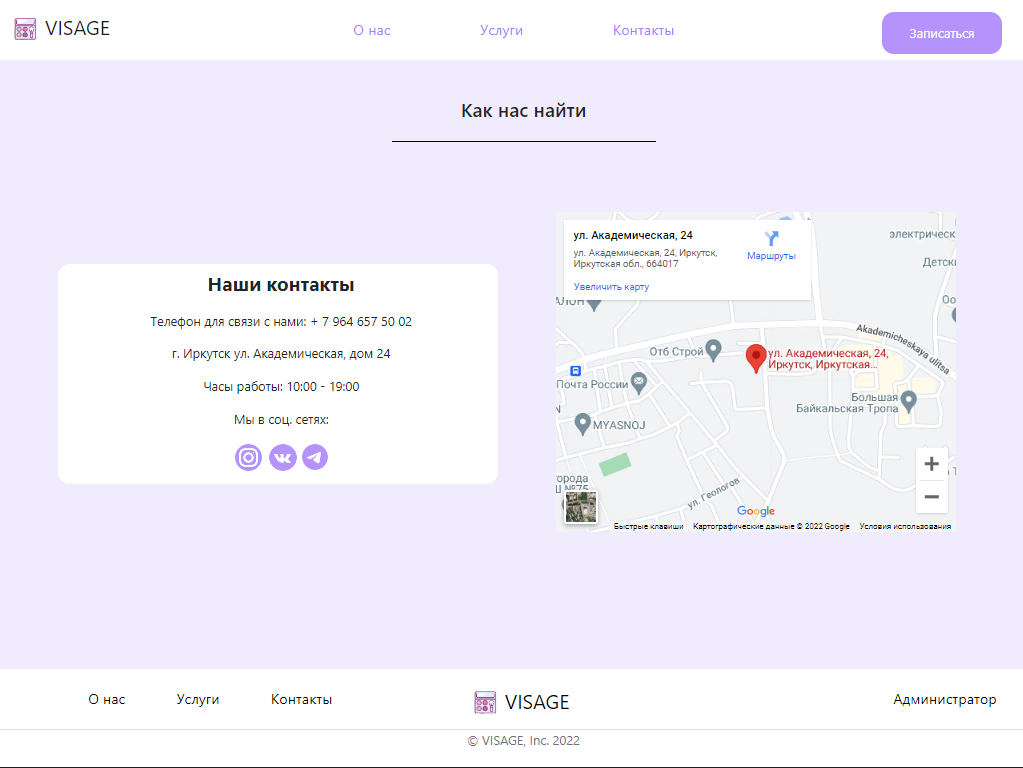


Рисунок 19 – Страница «Контакты»

На рисунке 20 изображена форма для записи, в которой клиент может записаться на сеанс к визажисту. При записи на занятие клиент вводит своё имя, номер телефона, выбирает макияж, прическу и визажиста, а также нужную ему дату и время. Клиент может выбрать либо оба вида услуг, либо один из представленных. Если клиент не выбрал ни одной услуги, то появляется сообщение об ошибке.

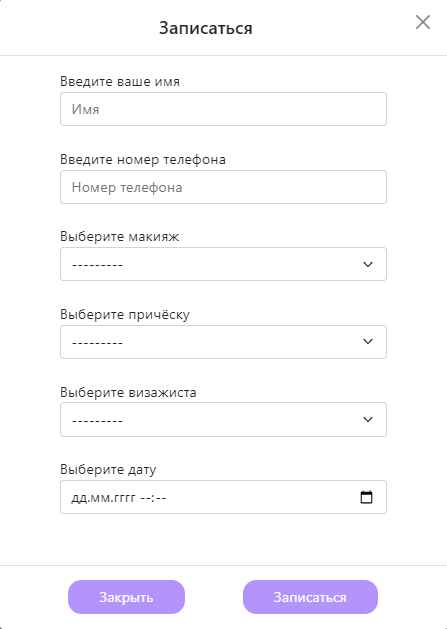


Рисунок 20 – Форма для записи

С помощью формы авторизации, изображенной на рисунке 21, предоставлена возможность авторизации для входа на панель управления.

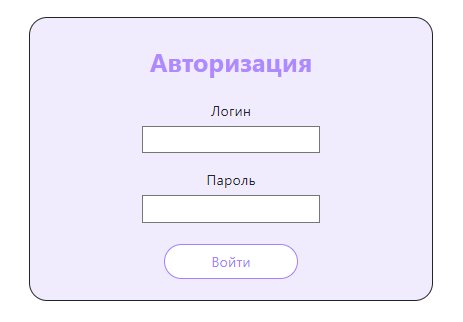


Рисунок 21 – Форма авторизации

* 1. **Разработка базы данных ИС**

База данных – эта неотъемлемая часть информационной системы, без которой она не может существовать. Поэтому разработка БД оказывает влияние на работу самой информационной системы.

Разработка базы данных ИС «Визаж» реализовывалась в СУБД SQLite. БД состоит из 5 таблиц. А именно: таблицы визажистов, квалификации визажистов, заявок, видов макияжей и причесок.

На рисунке 22 представлена таблица «vizajisti». Таблица имеет поля, такие как: идентификатор (первичный ключ), фамилия, имя, фото, идентификатор квалификации (внешний ключ).



Рисунок 22 – Таблица «vizajisti»

На рисунке 23 показана таблица «kvalifikaciya». Таблица имеет поля, такие как: идентификатор (первичный ключ), опыт работы и надбавка.



Рисунок 23 – Таблица «kvalifikaciya»

Таблица «zayavki» представлена на рисунке 24. Таблица имеет поля, такие как: идентификатор (первичный ключ), имя клиента, номер телефона, дата создания заявки, дата выполнения услуги, идентификатор макияжа, прически и визажиста (внешние ключи).

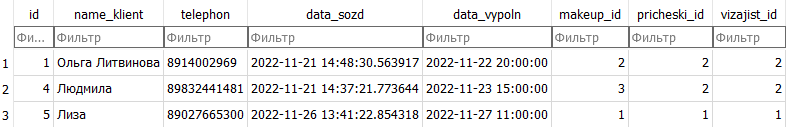


Рисунок 24 – Таблица «zayavki»

Также были разработаны таблицы видов макияжей и причесок, которые изображены на рисунках 25, 26. Таблицы имеют поля: идентификатор (первичный ключ), наименование, описание, стоимость и фото услуги.

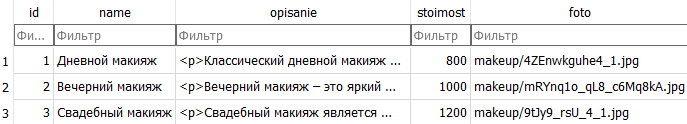


Рисунок 25 – Таблица «makeup»

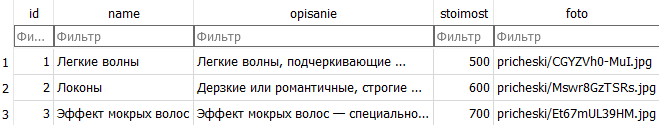


Рисунок 26 – Таблица «pricheski»

* 1. **Разработка ИС**

Подключение к базе данных SQLite осуществляется в конфигурационном файле settings.py, показанном на рисунке 27.

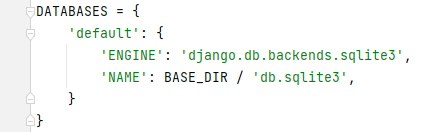


Рисунок 27 – Подключение к БД

В администрировании для добавления, редактирования и удаления используются функции, хранящиеся в файле views.py.

Для вывода всех визажистов из БД используется функция «admin\_vizajist», изображенная на рисунке 28.

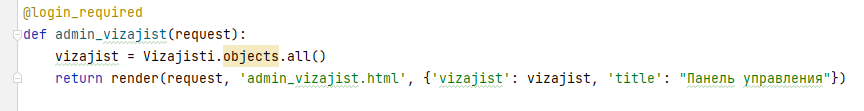


Рисунок 28 – Функция «admin\_vizajist»

Функция «add\_vizajist», представленная на рисунке 29, добавляет нового визажиста.



Рисунок 29 – Функция «add\_vizajist»

Функция «update\_vizajist» предназначена для редактирования информации о визажисте (рисунок 30).

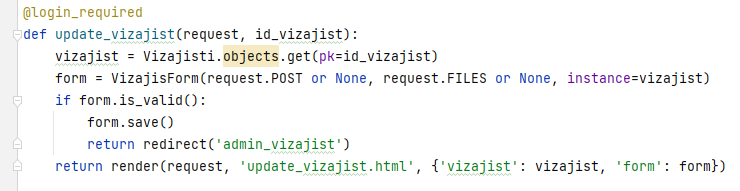


Рисунок 30 – Функция «update\_vizajist»

Функция «delete\_vizajist», показанная на рисунке 31, предназначена для удаления визажиста.

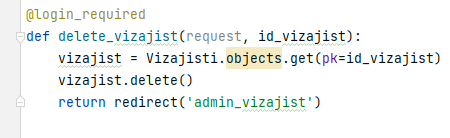


Рисунок 31 – Функция «delete\_vizajist»

1. **Документирование программного продукта**
   1. **Руководство пользователя ИС**

Информационная система «Визаж» имеет простой и интуитивно понятный интерфейс, что позволяет легко понять пользователю, как с ней работать.

# Заключение

# Список используемых источников

1. bestprogrammer.ru: MySQL Workbench, учебное пособие: полное руководство по инструменту СУБД. – URL: https://bestprogrammer.ru/baza-dannyh/uchebnoe-posobie-po-mysql-workbench-polnoe-rukovodstvo-po-instrumentu-subd (дата обращения: 21.09.2022). – Текст: электронный.
2. blog.skillfactory.ru: Bootstrap. – URL: https://clck.ru/32g7yu (дата обращения: 22.09.2022). – Текст: электронный.
3. blog.skillfactory.ru: MySQL. – URL: https://clck.ru/gfzqt (дата обращения: 22.09.2022). – Текст: электронный.
4. blog.skillfactory.ru: PostgreSQL. – URL: https://clck.ru/gw9i5 (дата обращения: 22.09.2022). – Текст: электронный.
5. habr.com: DFD (Data Flow Diagram) Диаграммы — зачем они нужны и какие бывают. – URL: https://habr.com/ru/post/668684/ (дата обращения: 15.09.2022). – Текст: электронный.
6. habr.com: Использование диаграммы классов UML при проектировании и документировании программного обеспечения. – URL: https://habr.com/ru/post/572234/ (дата обращения: 15.10.2022). – Текст: электронный.
7. hightech.in.ua: NetBeans IDE - универсальная интегрированная среда разработки приложений. – URL: https://hightech.in.ua/content/art-netbeans-ide (дата обращения: 24.09.2022). – Текст: электронный.
8. highload.today: Знакомство с PyCharm. – URL: https://highload.today/pycharm-python/ (дата обращения: 24.09.2022). – Текст: электронный.
9. learn.microsoft.com: Добро пожаловать в интегрированную среду разработки Visual Studio. – URL: https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2019 (дата обращения: 24.09.2022). – Текст: электронный.
10. mydocx.ru: Диаграмма декомпозиции. – URL: https://mydocx.ru/4-84615.html (дата обращения: 15.10.2022). – Текст: электронный.
11. nationalteam.worldskills.ru: Проектирование диаграммы деятельности UML (Activity Diadram). – URL: https://clck.ru/32g7zj (дата обращения: 15.10.2022). – 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

«ВИЗАЖ»

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (О.В. Литвинова) |

Иркутск 2022

**Общие сведения**

Наименование работы: информационная система «Визаж».

Исполнитель: студент ГБПОУИО «ИАТ», группы ПКС-19-1, Литвинова О. В.

Разработка информационной системы проходит в рамках курсового проекта по МДК.03.01 «Технология разработки программного обеспечения», на основании приказа №37-у от 12 сентября 2022 года.

Сроки разработки информационной системы с 12.09.2022 по 06.12.2022 года.

**Назначение и цели создания системы**

Целью курсового проекта является создание информационной системы «Визаж», которая предоставляет возможность клиентам просматривать, выбирать и бронировать время на сеанс.

В данной информационной системе реализуется следующий функционал:

− создание заявок;

* просмотр заявок;

− изменение и дополнение данных в ИС.

**Требования к системе в целом**

* 1. **Требования к структуре и функционированию системы**

Функции информационной системы:

1. Раздел «О нас»;
   1. создание новой заявки;
   2. отображение информации о визажистах;
   3. редактирование информации о визажистах;
2. Раздел «Услуги»;
   1. создание новой заявки;
   2. отображение информации об услугах;
   3. редактирование информации об услугах;
3. Раздел «Контакты»;
   1. отображение информации о способах связи.
   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.

* 1. **Требования к безопасности**

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

* 1. **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;

Минимальные системные требования для клиентского устройства:

1. Последние версии Google Chrome, Yandex Browser.
2. **Требования к документированию**

Основным документом, регламентирующими использование информационной системы является руководство пользователя.

Основным документом, регламентирующими разработку информационной системы является техническое задание.

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

В таблице 1 представлены плановые сроки начала и окончания работы по созданию информационной системы.

Таблица 1 – Плановые сроки по созданию информационной системы

|  |  |  |
| --- | --- | --- |
| **Наименование этапов курсового проекта** | **Срок** | **Объем** |
| Предпроектное исследование предметной области (выбор темы, постановка цели, задач, описание области применения, исследование предметной области) | 21.09.2022 | 10% |
| Разработка технического задания (выбор архитектуры программного обеспечения, выбор типа пользовательского интерфейса, выбор языка и среды программирования) | 26.09.2022 | 20% |
| Проектирование программного обеспечения  (разработка структурной и функциональной схемы ПО, проектирование базы данных (инфологическое, ER-модель, физическая модель) | 25.10.2022 | 50% |
| Разработка (программирование) и отладка программного продукта | 28.11.2022 | 90% |
| Составление программной документации (оформление ПЗ, руководство пользователя и презентации) | 06.12.2022 | 100% |
| Защита курсового проекта | 06.12 – 16.12.2022 | |

# Приложение Б – Листинг

Сделать комментарии\*