**ТЕХНИЧЕСКИЙ ПРОЕКТ**

на разработку

\_\_\_\_\_\_\_\_ информационной системы для автошколы \_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО |  | СОГЛАСОВАНО |
| Колледж ВятГУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Суслов П.А.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  М.П. |  | Руководитель УП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженкова М. Л.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |

2024

**Содержание**

[1. Уточнение структуры данных 3](#_Toc182776149)

[1.1. Связи между сущностями 8](#_Toc182776150)

[1.2. Целостность ссылок 9](#_Toc182776151)

[1.3. Целостность сущностей 9](#_Toc182776152)

[2. Формы представления данных 13](#_Toc182776153)

[3. Разработка алгоритма решения задачи 29](#_Toc182776154)

[4. Определение языка, структуры программы и требования к техническим средствам 39](#_Toc182776155)

[5. Требования к техническим средствам 40](#_Toc182776156)

# Уточнение структуры данных

Необходимо выделить набор сущностей с их свойствами для выбранной предметной области. Результаты занести в таблицу

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сущность | Свойство | Тип | Уникальность | Обязательность заполнения | Ограничения |
| Студенты | - ID\_студента  - СдачаВнутреннегоЭкзамена  - СдачаВнешнегоЭкзамена | - Числовой  - Булевой  - Булевой | - Да  - Нет  - Нет | - Да  - Нет  - Нет | - Первичный ключ  -  - |
| Контракты | - ID\_контракта  - ДатаНачалаКонтракта  - ДниКонтракта  - СтатусОплаты  - ID\_группы  - ID\_инструктор  - ID\_Студента  - ID\_плана | - Числовой  - Дата  - Числовой  - Булевой  - Числовой  - Числовой  - Числовой  - Числовой | - Да  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет | - Да  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет | - Первичный ключ  - Внешний ключ  - Внешний ключ  - Внешний ключ  - Внешний ключ  - |
| Планы обучения | - ID\_плана  - ФИО  - Теор. время  - Практ. время  - ID\_категории | - Числовой  - Текст  - Дата  - Дата  - Числовой | - Да  - Да  - Нет  - Нет  -Нет | - Да  - Да  - Нет  - Нет  - Да | - Первичный ключ  - Внешний ключ |
| Группы | - ID\_группы  - ID\_плана  - ID\_лектора | - Числовой  - Числовой  - Числовой | - Да  - Нет  - Нет | - Да  - Да  - Да | - Первичный ключ  - Внешний ключ  Внешний ключ |
| Лекторы | - ID\_лектора | - Числовой | - Да | - Да | - Первичный ключ |
| Категории управления | - ИД\_категории  - Имя  - Стоимость | - Числовой  - Текст  - Двойственный | - Да  - Нет  - Нет | - Да  - Нет  - Нет | - Первичный ключ |
| Машины | - ID\_машины  - Модель  - Цвет\_авто  - РегистрационныйНомер | - Числовой  - Текст  - Текст  - Числовой | - Да  - Нет  - Нет  - Нет | - Да  - Нет  - Нет  - Нет | -Первичный ключ  - |
| Инструктора | - ID\_Инструктор  - Рейтинг  - ИД\_категории  - ИД\_машины | - Числовой  - Числовой  - Числовой  - Числовой | - Да  - Нет  - Нет  - Нет | - Да  - Нет  - Нет  - Нет | - Первичный ключ  - Внешний ключ  - Внешний ключ |
| Пользователи | - ID\_юзера  - Фамилия  - Имя  - Отчество  - Адрес  - Телефон  - ДеньРождения  - Паспорт  - МедСертификат  - Почта | - Числовой  - Текст  - Текст  - Текст  - Текст  - Текст  - Дата  - Текст  - Текст  - Текст | - Да  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет | - Да  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет  - Нет | - Первичный ключ |
| УрокиЛектора | - ID\_урока  - ID\_лектора  - Дата урока | - Числовой  - Числовой  - Дата | - Да  - Нет  - Нет | - Да  - Нет  - Нет | - Первичный ключ  - Внешний ключ  - |
| УрокиИнструктора | - ID\_урока  - ID\_инструктора  - ID\_студента  - Дата урока | - Числовой  - Числовой  - Числовой  - Дата | - Да  - Нет  - Нет  - Нет | - Да  - Нет  - Нет  - Нет | - Первичный ключ  - Внешний ключ  - Внешний ключ  - |

На рисунке 1 изображена функциональная модель автошколы.

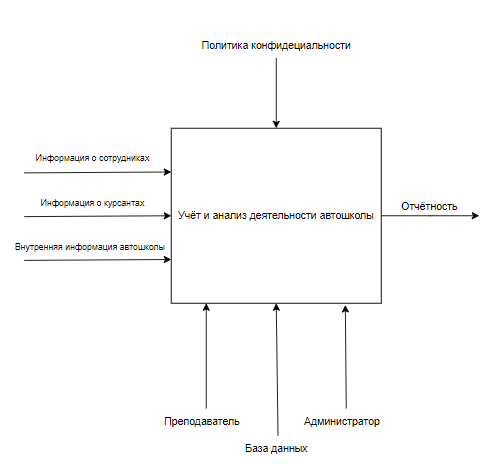


Рисунок 1 – Контекстная диаграмма

Внешней входящей информацией являются:

* Информация о сотрудниках
* Информация о курсантах
* Внутренняя информация автошколы

Управляющим воздействием будут являться:

* Политика конфиденциальности

Механизмом управления будут являться:

* Преподаватель
* База данных
* Администратор
* Приложение

Внешней исходящей информацией будут являться:

* Отчётность

Диаграмма декомпозиции, представляет контекстную функцию, разложенную на 4 составляющих более простых функций.

На рисунке 2 показан результат декомпозиции процесса деятельности автошколы.

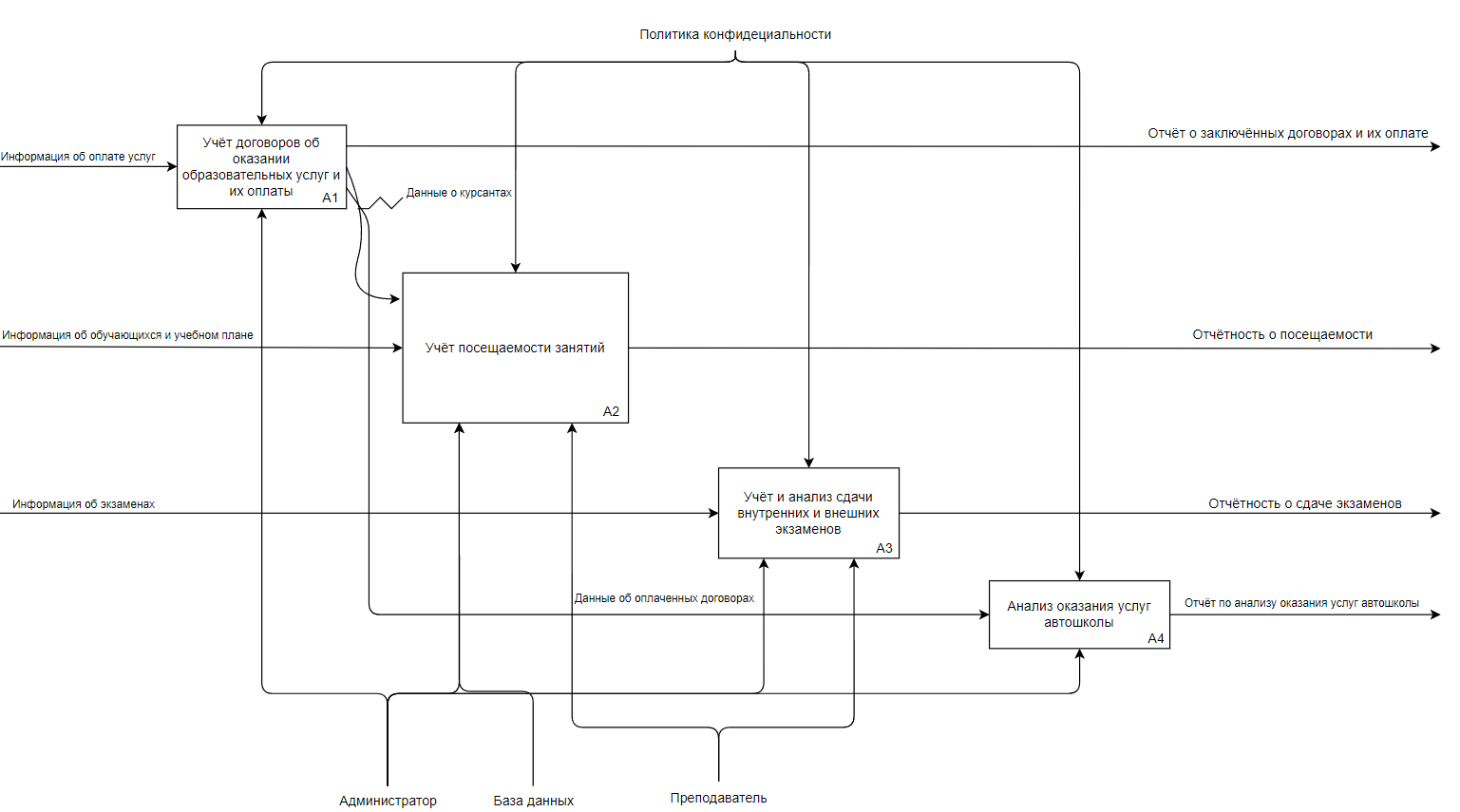


Рисунок 2 – Результат процесса декомпозиции

Первая функция выполняет проверку оплаченных договоров, на вход подается информация об оплате услуг, после чего получается отчет о заключенных договорах и оплате. Далее происходит учёт посещаемости занятий, на вход подается информация об обучающихся и учебном плане, на выходе получаем отчет о посещаемости. Затем анализ сдачи экзаменов, здесь получаем информацию об экзаменах и на выходе получаем отчетность о сдаче экзаменов. Последняя функция выполняет анализ оказания услуг автошколы, на вход получаем данные об оплаченных договорах, после чего получается отчет по анализу оказанных услуг автошколы.

На рисунках 3 и 4 показана база данных в логической ER-диаграмме и физической ER-диаграмме.

Логическая модель данных.

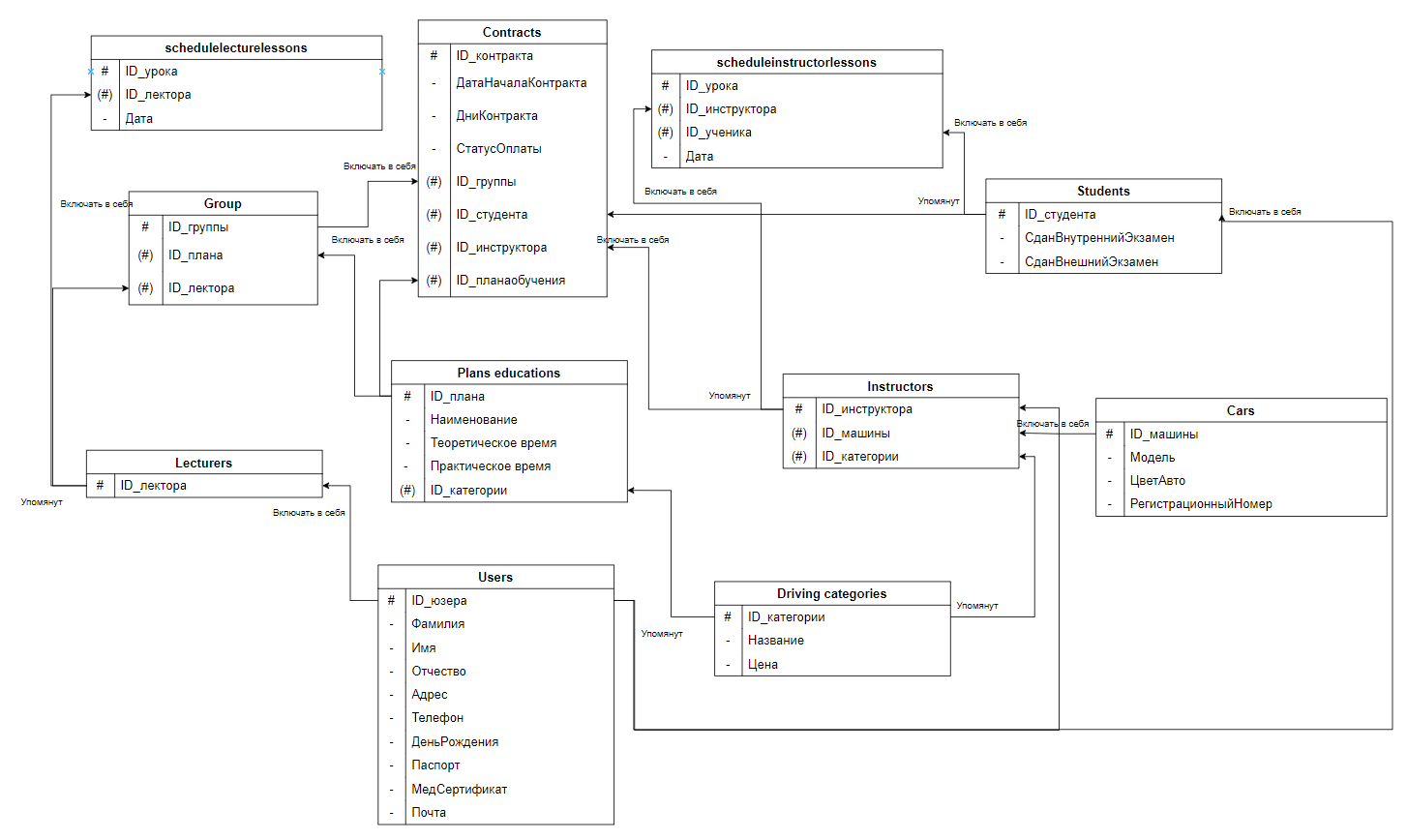


Рисунок 3 – Логическая модель данных.

Физическая модель данных.

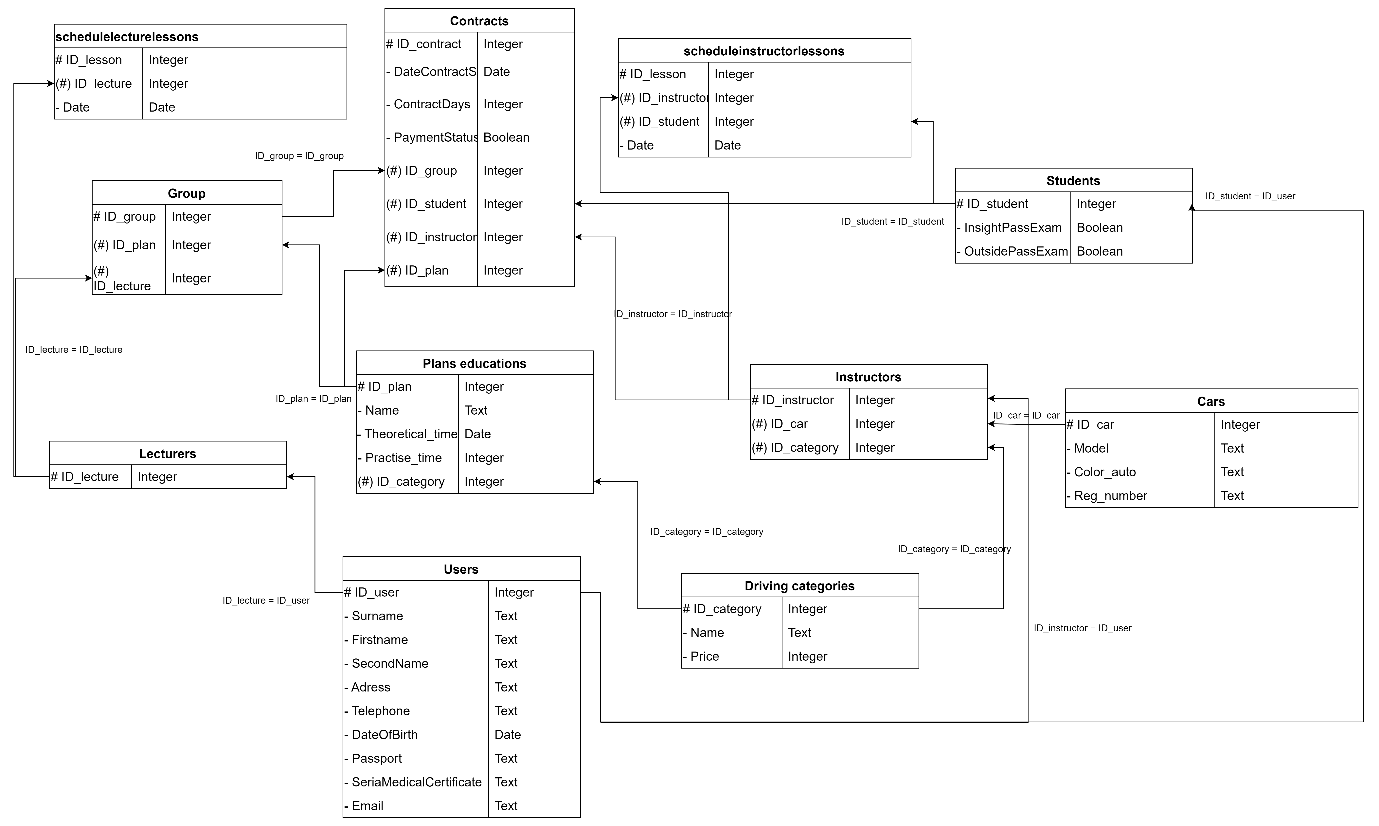


Рисунок 4 – Физическая модель данных.

Связи между сущностями

**Категории управления** – **Планы обучения**: Один ко многим, в одной категории управления может быть несколько планов обучения, но каждый план относится только к одной категории управления.

**Планы обучения** – **Группы**: Один ко многим, один план может применяться в нескольких группах, но каждая группа связана только с одним планом.

**Группы** – **Студенты**: Многие ко многим, один студент может быть в нескольких группах, и в одной группе может быть много студентов.

**Инструктора** – **Машины**: Один ко многим, у инструктора может быть несколько машин, но каждая машина принадлежит одному инструктору.

**Инструктора** – **УрокиИнструктора**: Один ко многим, инструктор может проводить несколько уроков, но каждый урок привязан к одному инструктору.

**Студенты** – **УрокиИнструктора**: Один ко многим, студент может проходить несколько уроков, но каждый урок может быть привязан только к одному студенту.

**УрокиИнструктора** – **УрокиЛектора**: Один ко многим, каждый урок инструктора связан с одним уроком лектора.

Целостность ссылок

**Студенты**:

**ID студента**: Первичный ключ, уникальный для каждого студента, используется для обеспечения целостности ссылок при связи с другими таблицами, например, в таблице **УрокиИнструктора**.

**Категории управления**:

**ID категории**: Первичный ключ, уникальный для каждой категории, используется для связи с другими таблицами, например, в таблице **Планы обучения**.

**Планы обучения**:

**ID плана**: Первичный ключ, уникальный для каждого плана обучения, используется как внешний ключ в таблице **Группы**.

**Группы**:

**ID группы**: Первичный ключ, уникальный для каждой группы, используется для связи с другими таблицами, например, в **Контракты**.

**ID плана**: Внешний ключ, связывающий группу с конкретным планом обучения.

**Инструктора**:

**ID инструктора**: Первичный ключ, уникальный для каждого инструктора, используется для связи с таблицами **УрокиИнструктора** и **Машины**.

**Машины**:

**ID машины**: Первичный ключ, уникальный для каждой машины, используется для связи с таблицей **Инструктора**.

**УрокиИнструктора**:

**ID урока**: Первичный ключ, уникальный для каждого урока инструктора, используется для связи с таблицами **Студенты** и **Инструктора**.

**УрокиЛектора**:

**ID урока**: Первичный ключ, уникальный для каждого урока лектора, используется для связи с таблицей **УрокиИнструктора**.

Целостность сущностей

**Студенты**:

**ID студента**: Уникальный идентификатор студента, обязателен для заполнения, чтобы обеспечить целостность сущности.

**СдачаВнутреннегоЭкзамена** и **СдачаВнешнегоЭкзамена**: Булевые значения, указывающие на статус экзаменов, необязательные для заполнения.

**Категории управления**:

**Имя**: Обязательное для заполнения текстовое поле, определяющее имя категории.

**Стоимость**: Необязательное числовое поле, содержащее стоимость, если она применима.

**Планы обучения**:

**ФИО**: Обязательное текстовое поле, определяющее название курса или ФИО лектора.

**Теоретическое и практическое время**: Дата или числовые значения, указывающие продолжительность теоретических и практических занятий.

**Группы**:

**ID группы**: Уникальный идентификатор группы, обязателен для заполнения.

**ID плана**: Внешний ключ, обязательный для указания, связывает группу с учебным планом.

**ID лектора**: Внешний ключ, связанный с таблицей **Лекторы**, обязательный для указания.

**Инструктора**:

**Рейтинг**: Числовое поле, не обязательно для заполнения, указывает на рейтинг инструктора, если он есть.

**Машины**:

**Модель** и **Цвет\_авто**: Обязательные текстовые поля, определяющие модель и цвет машины.

**РегистрационныйНомер**: Обязательное числовое поле, уникально для каждой машины.

**УрокиЛектора**:

**Дата урока**: Обязательное поле даты, указывающее дату проведения урока лектора.

**УрокиИнструктора**:

**Дата урока**: Обязательное поле даты, указывающее дату проведения урока.

Эта структура с целостностью ссылок и сущностей позволяет поддерживать согласованность данных в базе и предотвращать дублирование информации и ошибки в связях между сущностями.

Таблица описания полей базы данных.

|  |  |  |  |
| --- | --- | --- | --- |
| * Таблица | * Атрибут | * Тип | * Размер |
| * Users(Пользователи) | * ID\_user * Surname * Name * Lastname * Telephone * Date\_of\_Birth * Passport * SeriaMedical * Email | * INT * Varchar * Varchar * Varchar * Varchar * Date * Number * Number * Varchar | * - * 50 * 50 * 50 * 12 * 50 |
| * Contracts(Контракты) | * ID\_contract * DateContStart * ContractDays * PaymentStatus * ID\_group * ID\_instructor * ID\_student * ID\_plan | * INT * Date * INT * Bool * INT * INT * INT * INT | * 0, 1 |
| * Plans Educations (Планы обучения) | * ID\_plan * Name * Theor\_time * Pract\_time * ID\_category | * INT * Varchar * Date * Date * INT | * 50 |
| * Group (Группа) | * ID\_group * ID\_plan * ID\_lecture | * INT * INT * INT |  |
| * Lectures (Лектор) | * ID\_lecture | * INT | - |
| * DrivingCategories (Категории управления) | * ID\_category * Name * Price | * INT * Varchar * Real | * 50 |
| * Cars (Машины) | * ID\_car * ID\_model * Color\_auto * Req\_number\_auto | * INT * Varchar * Varchar * INT | * 50 * 50 |
| * Instructors (Инструктора) | * ID\_instructor * Rating * ID\_user * ID\_car * ID\_category | * INT * INT * INT * INT * INT |  |
| * Students(Студенты) | * ID\_student * PassInsightExam * PassOutsideExam | * INT * Bool * Bool |  |

# Формы представления данных

На рисунке 5 представлена диаграмма прецедентов системы автоматизации процесса.

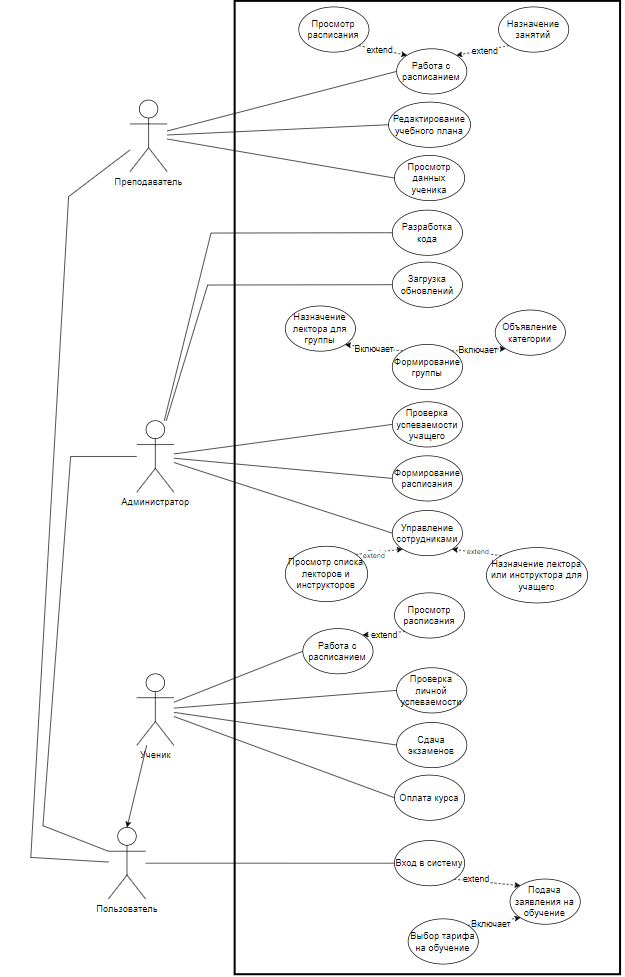


Рисунок 5 – Диаграмма прецендентов

Акторы:

Администратор – это ключевая роль в системе, ответственная за организацию и контроль всего учебного процесса. Этот актор имеет самые широкие права и доступ ко всем функциям системы

Ученик – это зарегистрированный пользователь системы, который проходит обучение. Учащиеся имеют ограниченные права, ориентированные на выполнение учебных задач и контроль за своим прогрессом.

Пользователь - это потенциальный ученик, который ещё не зарегистрирован в системе, но заинтересован в обучении. Гости могут выполнять ограниченные действия, связанные с подачей заявки и выбором курса.

Преподаватель - это общий термин, который охватывает как лекторов, так и инструкторов. Это человек, который обучает студентов, разрабатывает учебные программы и следит за успеваемостью.

Прецеденты:

Администратор

Просмотр расписания: Администратор просматривает текущее расписание занятий.

Назначение занятий: Администратор назначает занятия в расписание для различных групп.

Работа с расписаниями: Включает создание и редактирование расписания для учебных групп.

Редактирование учебного плана: Администратор изменяет и корректирует учебные планы для курсов.

Просмотр данных ученика: Доступ к профилю учащегося для просмотра информации о его прогрессе и успеваемости.

Разработка кода: Внесение изменений в программное обеспечение или платформу.

Загрузка обновлений: Внедрение обновленных версий программного обеспечения или исправлений.

Назначение лектора для группы: Назначение преподавателя для конкретной учебной группы.

Объявление категории: Администратор может объявить или создать новую категорию для курсов.

Формирование группы: Создание или редактирование учебных групп.

Проверка успеваемости учащегося: Просмотр успеваемости студентов или учащихся.

Формирование расписания: Создание расписания для учебных занятий.

Управление сотрудниками: Назначение и распределение задач для сотрудников, таких как лекторы или инструкторы.

Менеджер

Просмотр списка лекторов и инструкторов: Менеджер имеет доступ к информации обо всех преподавателях и инструкторах.

Просмотр расписания: Возможность просматривать расписание, чтобы контролировать учебный процесс.

Работа с расписаниями: Менеджер может редактировать расписание, если необходимо внести изменения.

Назначение лектора или инструктора для учащегося: Назначение преподавателя или инструктора индивидуально для учащегося.

Учащийся

Проверка личной успеваемости: Учащийся может просматривать свои оценки и успехи в учебе.

Сдача экзаменов: Возможность сдавать экзамены онлайн или через систему.

Оплата курса: Процесс оплаты курса или обучения через систему.

Вход в систему: Учащийся должен авторизоваться в системе для доступа ко всем остальным возможностям.

Гость

Подача заявления на обучение: Потенциальные студенты могут подать заявление на поступление на курс.

Выбор тарифа для обучения: Гость выбирает подходящий тариф или пакет обучения перед регистрацией.

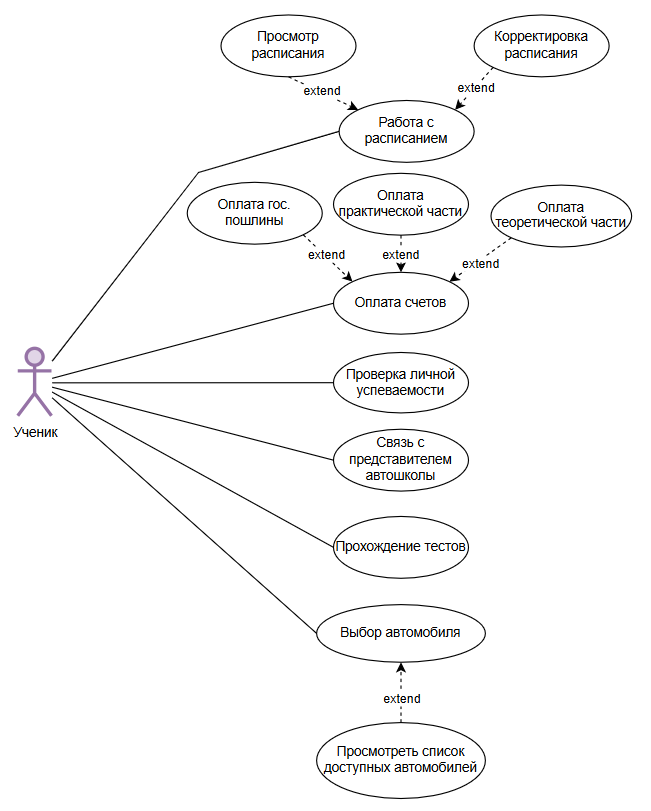


Рисунок 6 – диаграмма прецендента Ученик

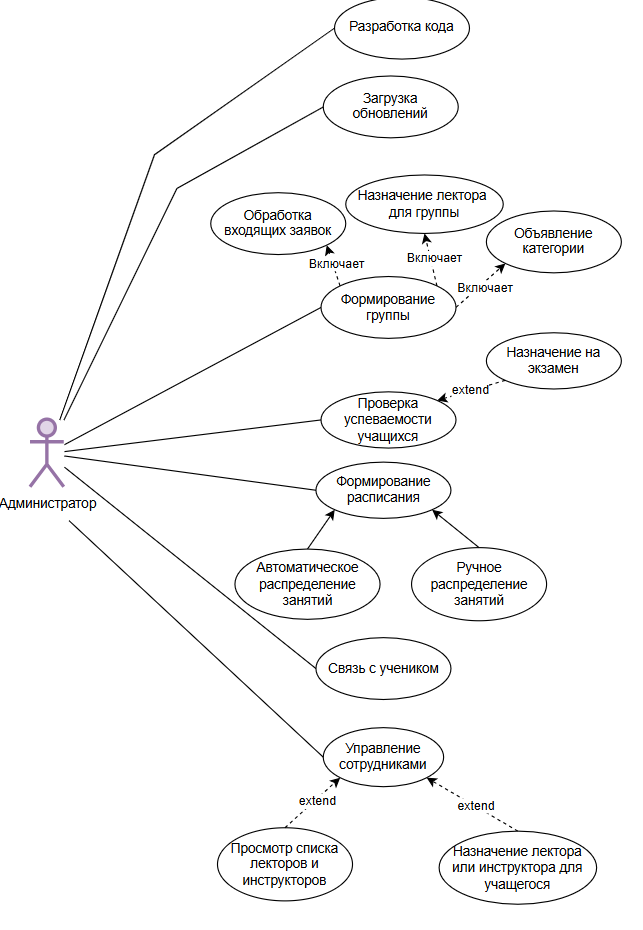


Рисунок 7 – диаграмма прецендента Администратор

Пользовательский интерфейс для авторизации/регистрации должен обеспечить выполнение следующий функций:

* Содержать окно взаимодействия для авторизации пользователей;
* Содержать два поля для ввода логина и пароля;
* Содержать кнопку «Войти».
* Содержать кнопку «Зарегистрироваться».

Прототип интерфейса авторизации представлен на рисунке 8.



Рисунок 8 – Прототип интерфейса окна авторизации

Пользовательский интерфейс личного кабинета должен удовлетворять следующим требованиям:

* Предоставлять окно взаимодействия для просмотра и управления личными данными.
* Содержать разделы и поля для просмотра и редактирования информации о пользователе, такие как имя, фамилия, адрес электронной почты и другие персональные данные.
* Предоставлять кнопки или ссылки для выполнения различных действий, таких как изменение пароля, обновление контактных данных и выход из аккаунта.
* Содержать элементы управления, позволяющие загрузить или изменить аватар пользователя, если такая функциональность поддерживается.
* Обеспечивать возможность просмотра истории активности в личном кабинете, если это применимо.
* Должен быть интуитивно понятным и удобным для использования, обеспечивая удовлетворение потребностей пользователя в управлении своим аккаунтом.

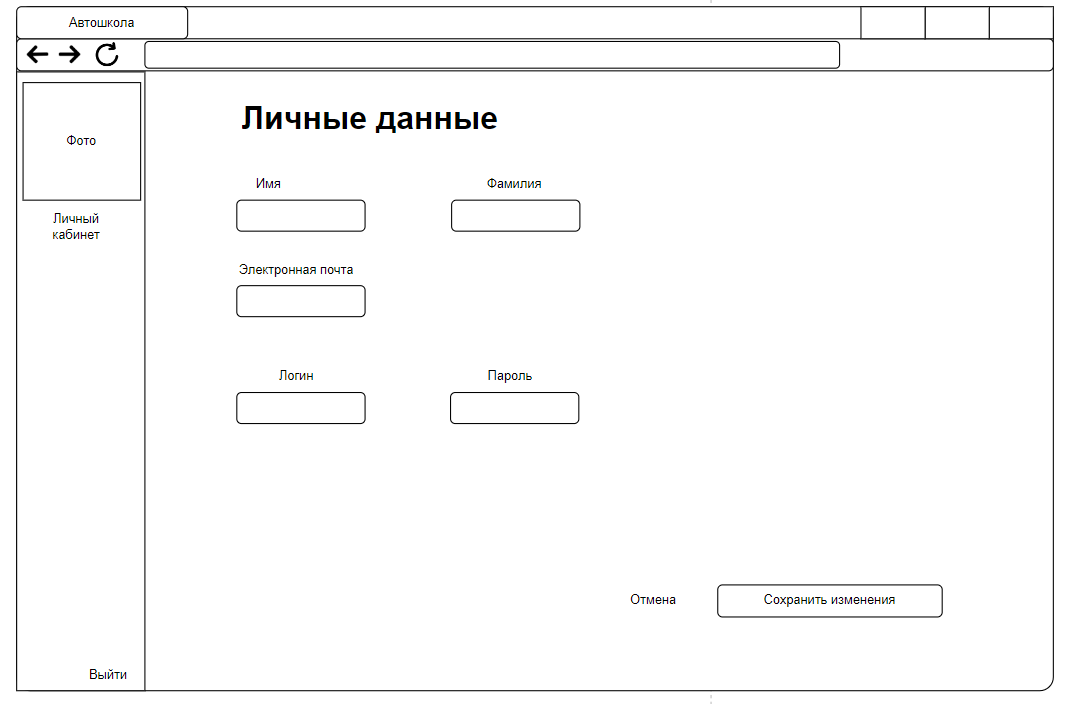


Рисунок 9 – Прототип вкладки Личный кабинет

Пользовательский интерфейс сайта информационной системы автошколы для администратора должен обеспечивать удобный и эффективный доступ к различным функциям и данным, необходимым для управления автошколой. Вот некоторые основные элементы, которые могут содержаться в таком интерфейсе:

* Панель навигации: В верхней части страницы может располагаться панель навигации с основными разделами системы, такими как "Ученики", "Инструкторы", "Расписание", "Финансы", "Отчеты" и т. д. Администратор может выбирать нужный раздел, чтобы управлять соответствующей информацией.
* Управление учениками: Создание, редактирование и удаление ученических профилей; Поиск учеников по различным критериям; Назначение инструкторов ученикам.
* Управление инструкторами: Добавление, редактирование и удаление профилей инструкторов; Поиск инструкторов и просмотр их графиков занятий.
* Расписание занятий: Просмотр и управление расписанием занятий; Назначение инструкторов и автомобилей на занятия.
* Финансовые операции: Просмотр и учет платежей от учеников; Генерация счетов и квитанций. Просмотр финансовых отчетов.
* Отчеты и аналитика: Генерация различных отчетов о работе автошколы, таких как успеваемость учеников, доходы и расходы, активность инструкторов и другие.
* Уведомления: Система уведомлений о важных событиях и изменениях в расписании.
* Настройки аккаунта администратора: Возможность изменения пароля и персональных настроек.
* Выход из системы: Кнопка для безопасного выхода из аккаунта администратора.
* Поддержка и контакты: Информация о контактах службы поддержки и справочные материалы.

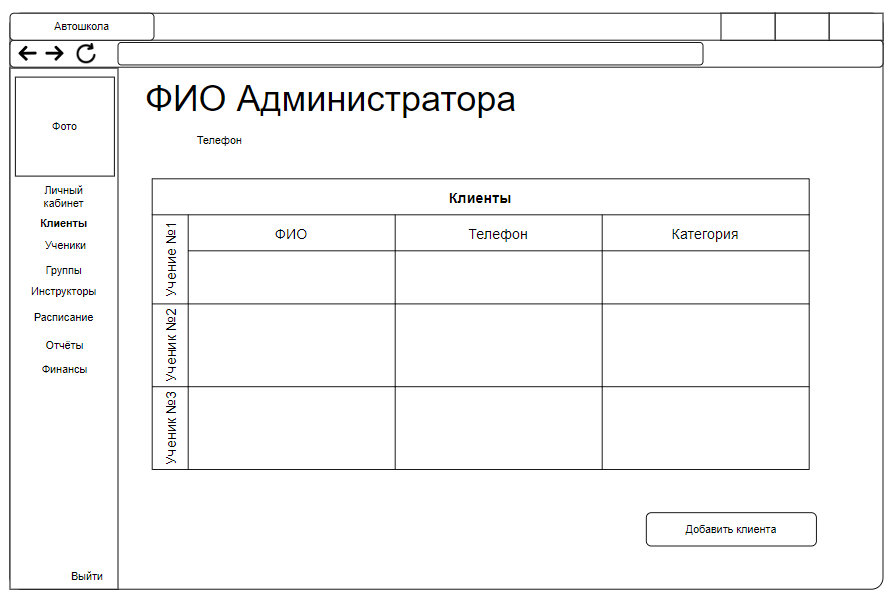


Рисунок 9 – Раздел “Клиенты” для администратора

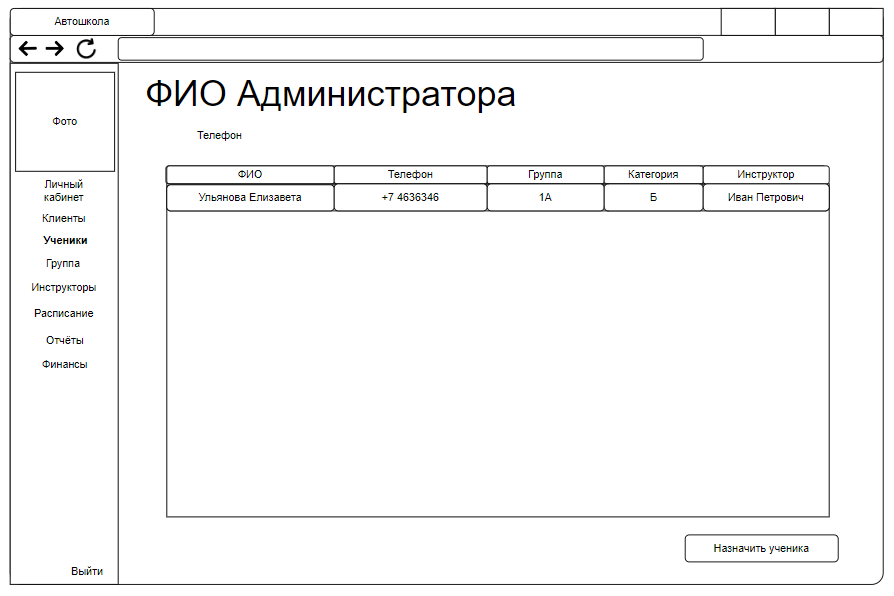


Рисунок 10 – Раздел “Ученики” для администратора

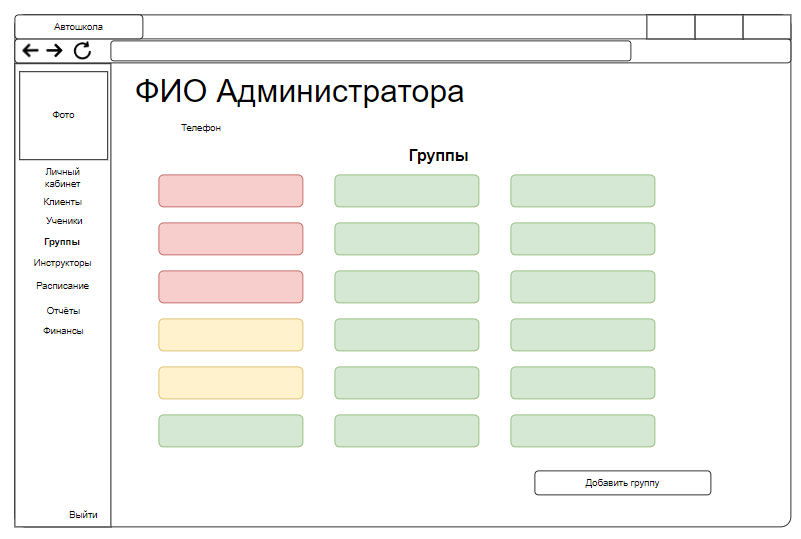


Рисунок 11 – Раздел “Группы” для администратора

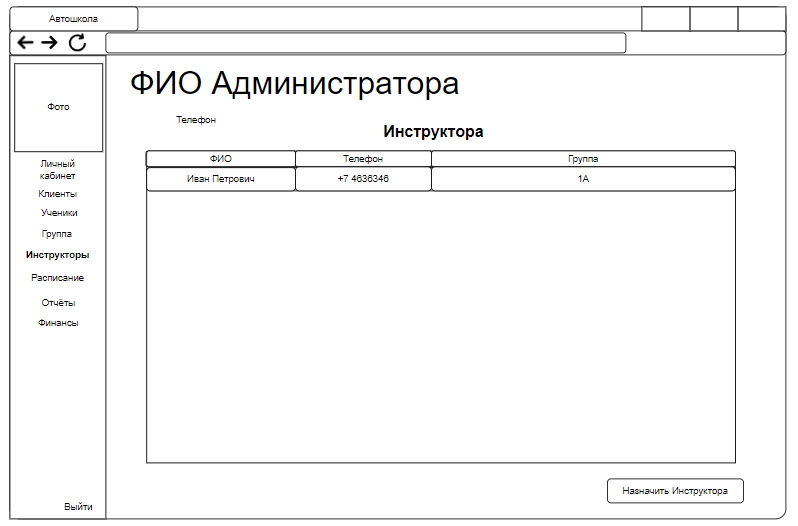


Рисунок 12 – Раздел “Инструкторы” для администратора

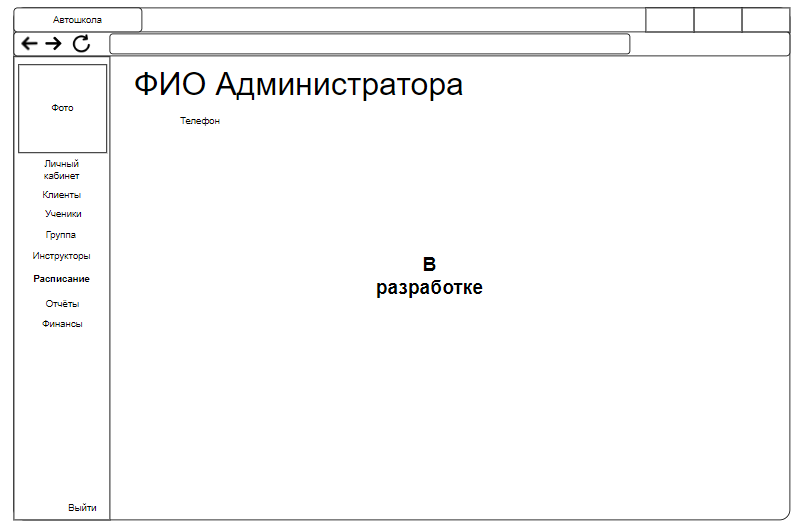


Рисунок 13 – Раздел “Расписание” для администратора

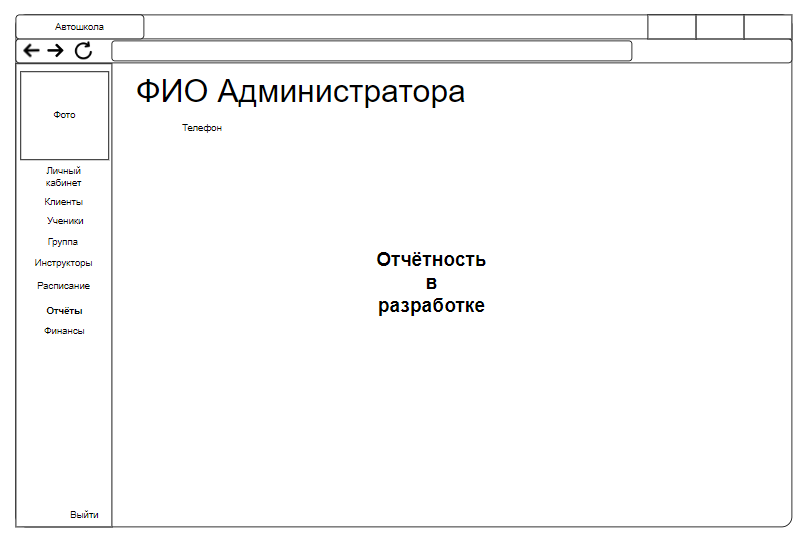


Рисунок 14 – Раздел “Отчёты” для администратора

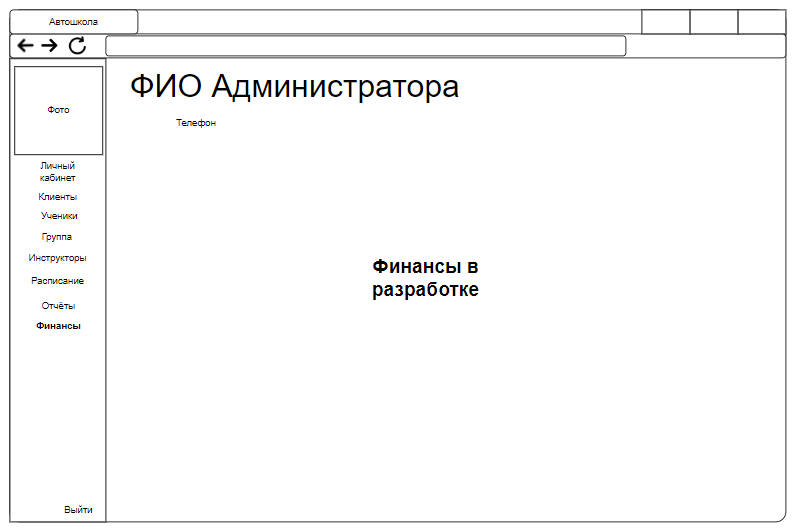


Рисунок 15– Раздел “Финансы” для администратора

Интерфейс должен быть интуитивно понятным и легко навигировать, чтобы администратор мог эффективно управлять всеми аспектами работы автошколы через систему. Кроме того, важно обеспечить безопасность данных и доступа, особенно при работе с личной информацией учеников и инструкторов.

Пользовательский интерфейс сайта информационной системы автошколы для преподавателя должен быть удобным и функциональным, чтобы облегчить процесс управления учебным процессом и взаимодействия с учениками. Вот некоторые основные элементы, которые могут содержаться в таком интерфейсе:

* Панель навигации: В верхней части страницы может располагаться панель навигации с основными разделами системы, такими как "Главная", "Расписание занятий", "Ученики", "Задания и оценки" и другие. Преподаватель может выбирать нужный раздел для выполнения своих задач.
* Расписание занятий: Просмотр собственного расписания занятий. Возможность добавления и изменения расписания. Просмотр информации о месте проведения занятий и времени.
* Управление учениками: Просмотр списка учеников, назначенных на занятия. Доступ к профилям учеников для просмотра и редактирования информации. Возможность проставления оценок и ведения журнала успеваемости.
* Задания и материалы: Загрузка учебных материалов и заданий для учеников. Просмотр сданных заданий и работ учеников. Выставление оценок и обратной связи по выполненным заданиям.
* Сообщения и уведомления: Система уведомлений о важных событиях и изменениях в расписании или заданиях. Возможность общения с учениками через встроенную систему сообщений.
* Настройки аккаунта преподавателя: Возможность изменения пароля и персональных настроек.
* Выход из системы: Кнопка для безопасного выхода из аккаунта преподавателя.
* Поддержка и контакты: Информация о контактах службы поддержки и справочные материалы.

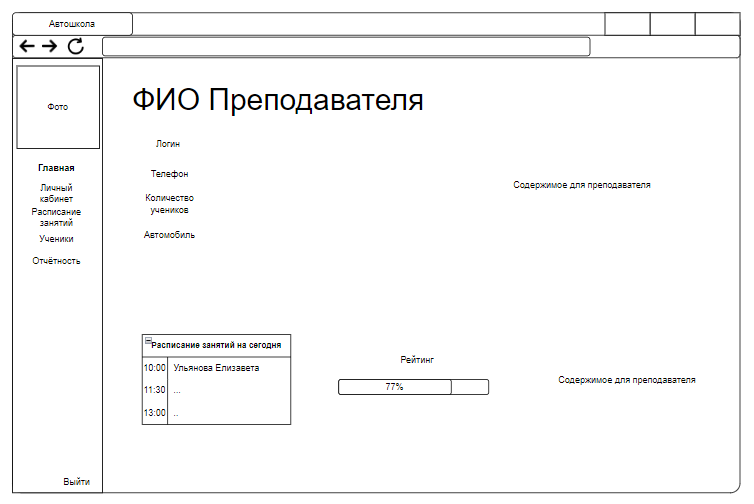


Рисунок 16 – Раздел “Главная” для преподавателя

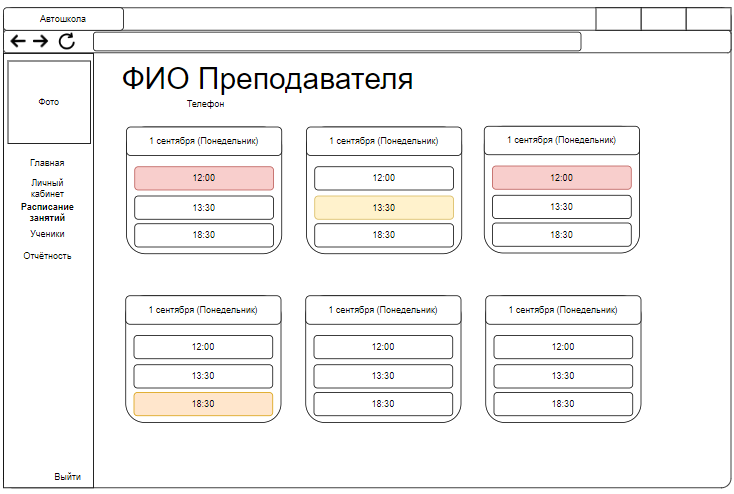


Рисунок 17 – Раздел “Расписание занятий” для преподавателя

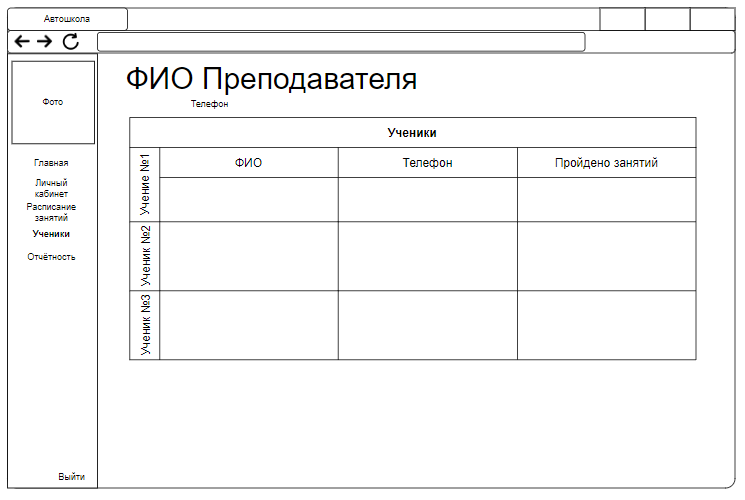


Рисунок 18 – Раздел “Ученики” для преподавателя

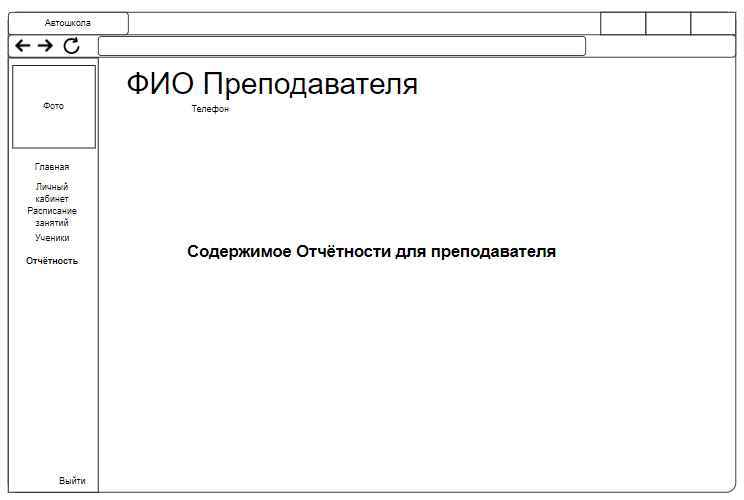


Рисунок 19 – Раздел “Отчётность” для преподавателя

Интерфейс должен быть интуитивно понятным и легко навигировать, чтобы преподаватель мог эффективно управлять учебным процессом, отслеживать прогресс учеников и взаимодействовать с ними. Кроме того, важно обеспечить безопасность данных и доступа, особенно при работе с учебными материалами и оценками.

Пользовательский интерфейс сайта информационной системы автошколы для студента должен быть простым и удобным для использования, чтобы облегчить доступ к необходимой информации и функциональности. Вот некоторые основные элементы, которые могут содержаться в таком интерфейсе:

* Личный профиль студента: Информация о студенте, такая как имя, фамилия, контактные данные; Возможность редактирования личных данных.
* Расписание занятий: Просмотр расписания занятий, включая даты, время и место проведения; Уведомления о предстоящих занятиях и изменениях в расписании.
* Учебные материалы: Доступ к учебным материалам, включая учебники, видеоуроки, инструкции и другие обучающие ресурсы.
* Задания и домашние задания: Просмотр заданий, которые необходимо выполнить; Загрузка и отправка выполненных заданий; Просмотр оценок и обратной связи по заданиям.
* Финансовая информация: Просмотр информации о статусе оплаты и возможность оплаты обучения. История платежей и квитанции.
* Уведомления: Система уведомлений о важных событиях, например, об изменениях в расписании или новых материалах.
* Поддержка и контакты: Информация о контактах службы поддержки и справочные материалы.
* Выход из системы: Кнопка для безопасного выхода из аккаунта студента.



Рисунок 20 – Раздел “Главная” для студента



Рисунок 21 – Раздел “Учебные материалы” для студента

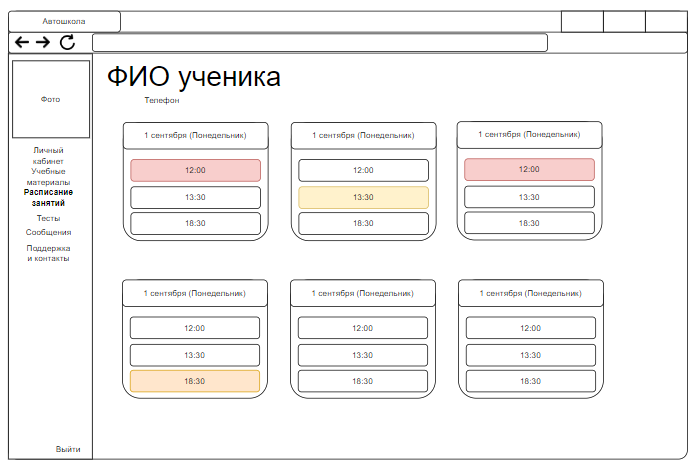


Рисунок 22 – Раздел “Расписание занятий” для студента

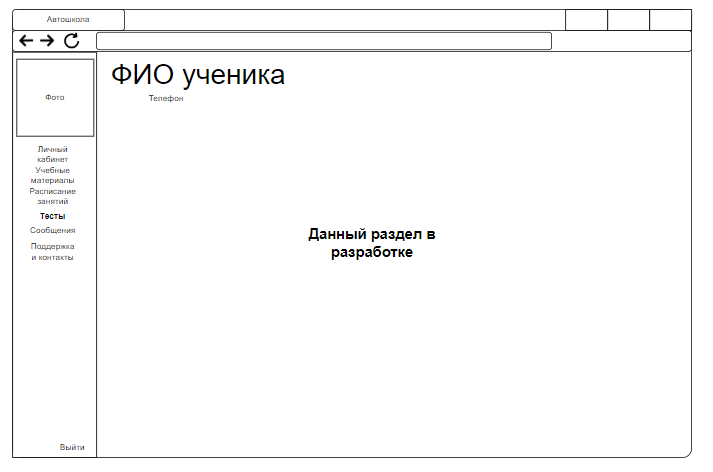


Рисунок 23 – Раздел “Тесты” для студента

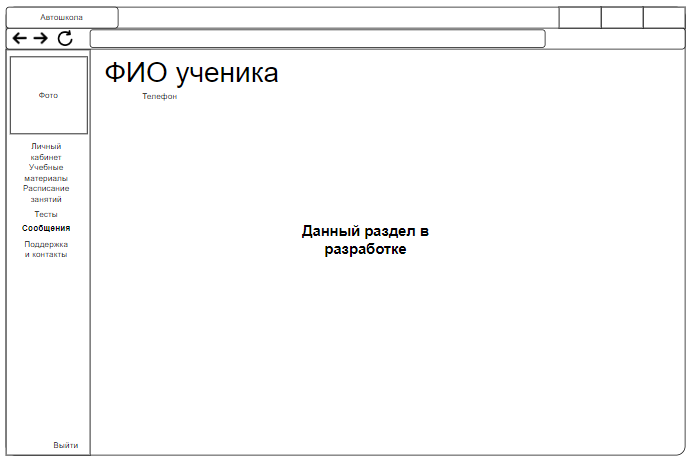


Рисунок 24 – Раздел “Сообщения” для студента

Интерфейс должен быть дружелюбным и интуитивно понятным, чтобы студент мог легко управлять своими учебными данными, просматривать расписание и получать доступ к необходимой информации для успешного обучения. Также важно обеспечить безопасность личных данных и конфиденциальность коммуникации.

1. **Разработка алгоритма решения задачи**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Название сценария** | **Цель** | **Предусловия** | **Основной поток** | **Результат** | **Исключения** |
| Авторизация пользователя | Обеспечить вход зарегистрированного пользователя в систему | Пользователь зарегистрирован в системе и имеет логин и пароль. | 1. Пользователь открывает окно авторизации. 2. Вводит логин и пароль. 3. Нажимает кнопку «Войти». 4. Система проверяет корректность данных. 5. При успешной проверке пользователь входит в личный кабинет. | Пользователь успешно авторизован и получает доступ к своей учетной записи. | 1. Некорректный логин или пароль — система отображает сообщение об ошибке. 2. Поля не заполнены — система требует ввести данные. 3. Системная ошибка — система предлагает повторить попытку позже. |
| Регистрация нового пользователя | Создать учетную запись нового пользователя в системе. | Пользователь не зарегистрирован в системе. | 1. Пользователь открывает окно регистрации. 2. Вводит личные данные (ФИО, email, телефон, пароль). 3. Подтверждает пароль. 4. Нажимает кнопку «Зарегистрироваться». 5. Система проверяет корректность введённых данных. 6. При успешной проверке создаётся учетная запись, и пользователь получает уведомление об успешной регистрации. | Учетная запись успешно создана, пользователь может авторизоваться. | 1. Email или телефон уже зарегистрированы — система уведомляет об этом. 2. Поля заполнены некорректно — система отображает ошибки. 3. Ошибка сохранения данных — система предлагает повторить попытку. |
| Обновление личных данных | Обеспечить пользователю возможность изменить свои персональные данные. | Пользователь авторизован в системе. | 1. Пользователь открывает вкладку «Личные данные». 2. Нажимает кнопку «Редактировать». 3. Вносит изменения в необходимые поля (например, email, номер телефона). 4. Нажимает кнопку «Сохранить». 5. Система проверяет корректность введённых данных. 6. Изменения успешно сохраняются. | Данные пользователя обновлены. | 1. Поля заполнены некорректно — система уведомляет об ошибках. 2. Ошибка сохранения данных — система предлагает повторить попытку. |
| Просмотр расписания занятий | Предоставить пользователю возможность просмотреть актуальное расписание занятий. | Пользователь авторизован в системе. | 1. Пользователь открывает раздел «Расписание занятий». 2. Система отображает актуальное расписание с указанием дат, времени и места проведения занятий. 3. Пользователь просматривает интересующую информацию. | Пользователь получает доступ к расписанию занятий. | 1. Расписание недоступно — система уведомляет пользователя. 2. Нет данных для отображения — выводится сообщение о пустом расписании. |
| Отмена записи ученика | Отменить запись ученика | Ученик записан на занятие | 1. Администратор открывает запись ученика. 2. Нажимает «Отменить». 3. Подтверждает отмену. 4. Система уведомляет ученика. | Запись отменена, ученик уведомлен. | 1. Запись уже прошла — система уведомляет администратора. 2. Ошибка связи — предложить повторить отправку уведомления. |
| Добавление учеников | Зарегистрировать нового ученика | Данные ученика известны | 1. Администратор открывает раздел «Ученики». 2. Нажимает «Добавить ученика». 3. Вводит данные (ФИО, контакты, паспорт и т. д.). 4. Сохраняет изменения. | Новый ученик добавлен в систему. | 1. Данные не сохранены из-за ошибки — предложить повторить. 2. Дублирование — уведомить администратора. |
| Учет автомобилей | Учитывать занятость и техническое состояние | Автомобили добавлены в систему | 1. Администратор открывает раздел «Автомобили». 2. Отмечает занятость или состояние автомобиля. 3. Сохраняет изменения. | Данные о состоянии или занятости автомобиля обновлены. | Невозможно обновить из-за ошибки — предложить повторить действие. |
| Управление базой учеников | Обновлять данные учеников | Ученик зарегистрирован в системе | 1. Администратор открывает раздел «Ученики». 2. Находит нужного ученика. 3. Вносит изменения. 4. Сохраняет данные. | Данные ученика обновлены. | Ошибка сохранения данных — предложить повторить действие.. |
| Оповещение учеников | Уведомить ученика о занятиях или изменениях | Ученик зарегистрирован в системе | 1. Администратор открывает запись ученика. 2. Создаёт уведомление. 3. Система отправляет сообщение через выбранный канал (SMS, email). | Ученик уведомлен | Ошибка отправки — предложить повторить или проверить контактные данные ученика. |
| Просмотр списка свободных автомобилей | Проверить доступные автомобили | База данных автомобилей актуальна | 1. Администратор заходит в систему. 2. Выбирает раздел "Список свободных автомобилей". 3. Система отображает список доступных автомобилей. | Отображен список свободных автомобилей | База данных недоступна — система уведомляет об ошибке. |
| Размещение информации о местоположении автомобиля | Обновить данные о местоположении автомобиля | Список автомобилей доступен | 1. Администратор выбирает автомобиль из списка. 2. Вводит или обновляет данные о местоположении. 3. Сохраняет изменения. | Информация о местоположении автомобиля успешно обновлена | Ошибка при сохранении — система предлагает повторить действие. |

Для лучшего понимания алгоритма «Авторизации»» была создана схема алгоритма, которая изображена ниже (рис.25).

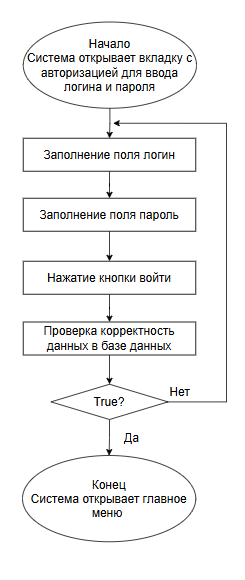


Рисунок 25 – Алгоритм «Авторизации»

Для лучшего понимания алгоритма «Регистрации» была создана схема алгоритма, которая изображена ниже (рис.26).

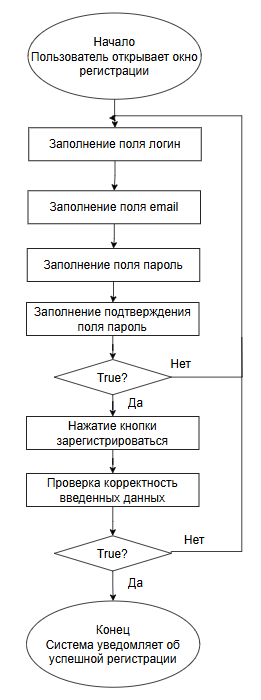


Рисунок 26 – Алгоритм «Регистрации»

Для лучшего понимания алгоритма «Обновление личных данных» была создана схема алгоритма, которая изображена ниже (рис.27).

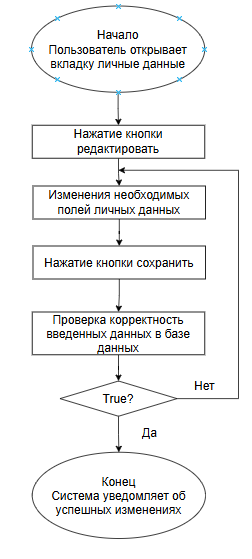


Рисунок 27 – Алгоритм «Обновление личных данных»

Для лучшего понимания алгоритма «Обновление личных данных» была создана схема алгоритма, которая изображена ниже (рис.28).

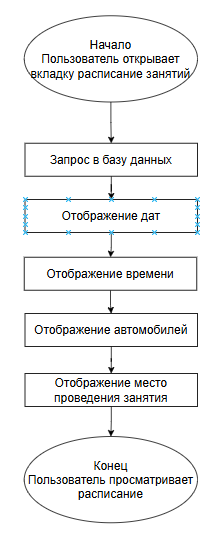


Рисунок 28 – Алгоритм «Просмотр расписания занятий»

Для лучшего понимания алгоритма «Отмена записи ученика» была создана схема алгоритма, которая изображена ниже (рис.29).

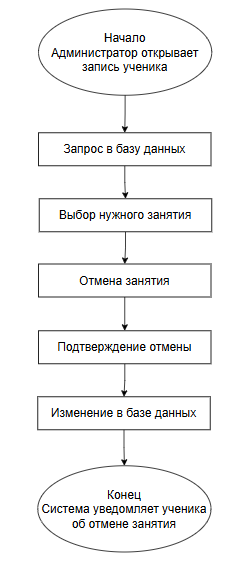


Рисунок 29 – Алгоритм «Отмена записи ученика»

Для лучшего понимания алгоритма «Управление базой учеников» была создана схема алгоритма, которая изображена ниже (рис.30).

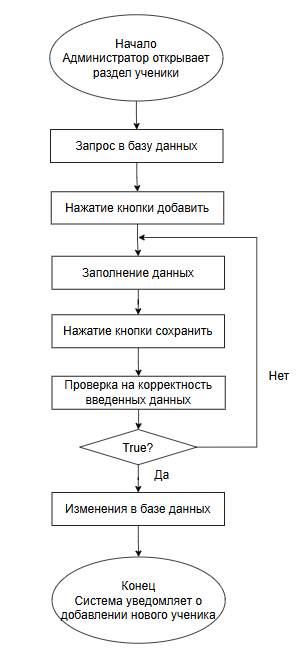


Рисунок 30 – Алгоритм «Управление базой учеников»

Для лучшего понимания алгоритма «Просмотр списка свободных автомобилей и Размещение информации о местоположении автомобиля» была создана схема алгоритма, которая изображена ниже (рис.31).

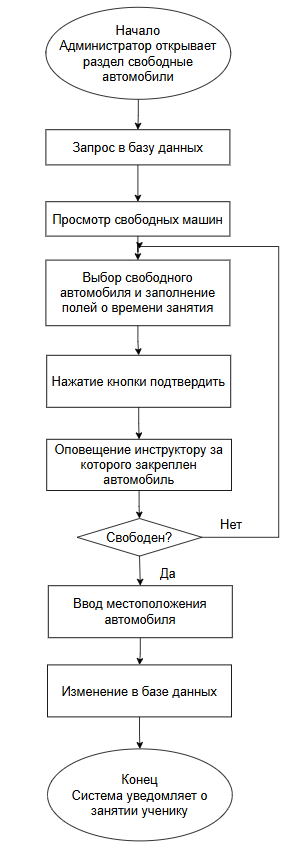


Рисунок 31 – Алгоритм «Просмотр списка свободных автомобилей и Размещение информации о местоположении автомобиля»

1. **Определение языка, структуры программы и требования к техническим средствам**

Язык программирования: был выбран Python из-за скорости работы с разными типами данных, а также скорости работы с ними.

Фреймворк: был выбран Django из-за возможности на нем создания графические приложения, огромное количество дополнений и встроенных функций в данном фреймворке.

Основные задачи Django:

Регистрация пользователей: Студенты и преподаватели могут регистрироваться в системе.

Управление данными: База данных хранит информацию о студентах, инструкторах, занятиях и экзаменах.

API для взаимодействия с фронтендом: Используется для передачи и получения данных в формате JSON.

Безопасность: Регистрация и авторизация пользователей, контроль доступа к данным в зависимости от ролей (студент, инструктор, администратор).

Логика работы с данными: Реализация бизнес-логики, например, расчет успеваемости, управление расписанием занятий.

Хранение данных: используется MySQL из-за небольшого количества хранимых данных, удобство и скорости в подключении к Python.

1. **Требования к техническим средствам**

Минмальные системные требования к окружению:

* Операционная система: Windows 7 или более новая версия;
* Оперативная память (ОЗУ): 4 ГБ;
* Жесткий диск: 10 ГБ;
* Процессор: Intel Core i3 или AMD FX-8300;
* Разрешение экрана: минимум 1024x768 пикселей для оптимального отображения интерфейса
* Сетевое подключение: стабильное соединение с высокой скоростью передачи данных для корректного подключения и работы базы данных.
* Веб-браузер: для доступа к веб-интерфейсу системы рекомендуется использовать актуальные версии браузеров, таких как Google Chrome, Mozilla Firefox или Microsoft Edge.
* **Внешние устройства:** для работы с печатью и сканированием документов, рекомендуется наличие подключенного принтера и сканера.