МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономики и управления на предприятии природопользования»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 «Прикладная информатика»
(квалификация – бакалавр)

На тему «Проектирование информационной системы «Разработка концепции ИС»»

Исполнитель Родионов Даниил Андреевич

Руководитель к.т.н.Сафонова Татьяна Владимировна

«К защите допускаю»
Руководитель кафедрой_____
кандидат экономических наук
Майборода Евгений Викторович
«17» 01 2026 год

Филиал Российского государственного гидрометеорологического университета в г. Туапсе
НОРМОКОНТРОЛЬ ПРОЙДЕН
«12» 01 2026
Мартынова Т.В.

Туапсе
2026

ОГЛАВЛЕНИЕ

Введение

В связи с активным ростом популярности владения личным транспортным средством, увеличивается и популярность автошкол, вследствие чего среди которых возрастает конкуренция.Для того что бы автошкола могла быть конкурентоспособной, необходимо постоянно повышать эффективность труда и получать наилучшие результаты деятельности при минимальных затратах труда и средств. К исследованиям эффективности труда нужно походить ни только с учетом оценки выполнения планов, но и пытаться находить пути для экономического и социального роста с учетом оказания помощи в успешном принятии обоснованных с точки зрения теории менеджмента решений.

Разработка самой концепции информационной системы однозначно позволит:

— учитывать договоры на образовательные услуги и отслеживать оплату за обучение;

— контролировать процесс посещаемости занятийкурсантами;

— систематически вести учет и анализ данных по внутренним и внешним экзаменам;

— анализировать качество услуг автошколы.

Актуальность настоящего исследования обусловлена потребностью в разработке концепции информационной системы (ИС) посредством проектирования Telegram–бота. Данная автоматизация несомненно позволит автоматизировать рутинные производственные задачи, такие как, редактирование личных данных, составление и ведение расписания занятий, прием и обработку платежей, а также позволит облегчить работу преподавателям, что безусловно повысит уровень и качество обучения с помощью тестовых упражнений по ПДД.

Объектом исследования выпускной квалификационной работы является процессучета и анализа деятельности АНО ДПО Автошкола «Навигатор+».

Предметом исследования выпускной квалификационной работы являетсяпроцесс проектирования Telegram-ботас учетом определенной концепции автошколы.

Целью выпускной квалификационной работы является разработка информационной системы «Автошкола» на основе Telegram-бота, интегрирующую необходимые функции и задачи, связанные с обучением курсантов. Для достижения поставленной были определены следующие задачи:

— изучить уже существующие программный продукты, имеющие схожий функционал, и сделать вывод о необходимости создания нового специализированного программного продукта;

— исследовать возможности применения Telegram-бота в автошколах для автоматизации задач и улучшения процесса обучения;

— определить требования к функциональности и интерфейсу системы, учитывая потребности автошколы и пользователей;

— реализовать прототип системы и провести тестирование его функциональности, надежности и безопасности

— провести оценку экономической обоснованности разработки данной системы.

В процессе написания выпускной квалификационной работы применялись различные методические подходы, благодаря которым был исследован экспериментальный метод и выделен системный анализ.

Теоретической основой послужили работы как российских, так и зарубежных ученых, которые активно изучали информационные платформы, проектировали площадки и создавали учебные веб-сайты.

1Аналитическая часть

1.1Анализ предметной области и требований

Официальная статистика на сегодняшний день свидетельствует о том, что в России функционирует свыше 14000 автошкол. Анализ данных, проиллюстрированных на рисунке 1.1 свидетельствует по региональному сравнению учебных заведений. Исходя из представленных данных лидируют, несомненно, Краснодарский край и Московская область. На диаграмме так же представлены сведения о соотношении негосударственных автошкол к общему количеству учебных заведений дополнительного образования в регионе. Невооруженным взглядом можно отметить, что число частных автошкол большим перевесом превиолируют на государственными[2].

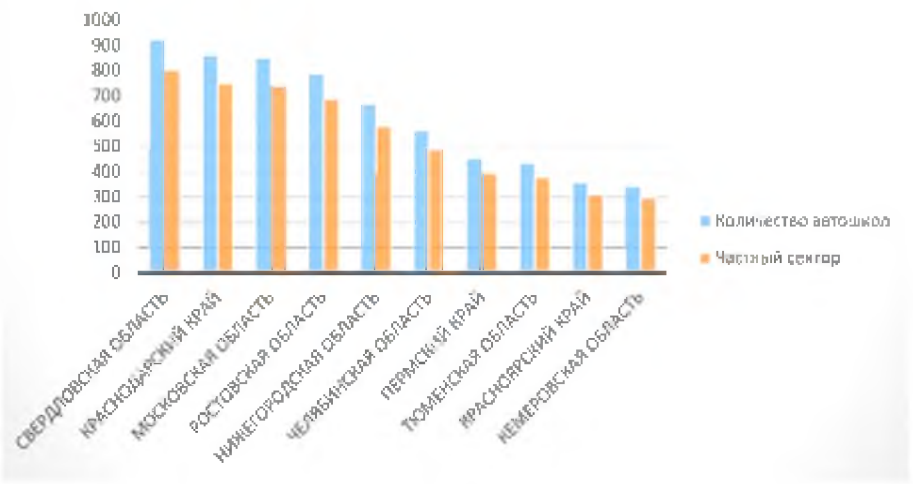


Рисунок 1.1 – Анализ количества автошкол в регионах России

Автошкола «Навигатор +» работает с октября 2010 года. Расположено по адресу: г.Туапсе, ул. К.Маркса дом 1 (рисунок 1.2). Учреждено одним лицом, которое на момент создания учреждения является единственным участником для ведения любых не запрещенных законом видов хозяйственной деятельности, в том числе:

— образовательная деятельность;

— деятельность образовательных учреждений широкого профиля и специализированных;

— иная не запрещенная законом деятельность.

Рисунок 1.2 – АНО ДПО Автошкола «Навигатор+»

Автошкола имеет следующие производственные задачи:

— реализация основных программ профессионального обучения по подготовке и переподготовке по профессии водитель;

— разработка учебных планов и образовательных программ.

В автошколе реализуются программы обучения по подготовке и переподготовке водителей транспортных средств на категорию «В». В автошколе работают опытные и квалифицированные преподаватели и мастера производственного обучения. Занятия проводятся в специально оборудованных классах и на учебном автодроме.

Законодательно предусмотрено лицензирование или получение определенного вида разрешений, на основании которых определяются виды деятельности автошколы в установленном порядке.

Автошкола несёт полную имущественную ответственность по своим обязательствам, детально и последовательно ведет самостоятельно бухгалтерский учет и осуществляет статистические данные, а так же производит предусмотренные законодательством бюджетные платежи.

Обучение в автошколе ведется по профессиональной программе обучения категории «В», соответствующей современным требованиям и оснащенной новейшим оборудованием.

Курсы вождения проводятся под руководством опытных мастеров производственного обучения в течение всего календарного года. Определим преимущества обучения:

Во-первых, рассрочка платежа.

Во-вторых, техническая оснащенность автошколы включает:

— технические средства обучения (компьютеры, проекторы с соответствующим программным обеспечением);

— тренажеры первоначального обучения навыкам вождения, оборудованные учебные транспортные средства;

— площадка первоначального обучения вождению (рисунок 1.3);

Рисунок 1.3 – Оборудованный автодром в с.Цыпка

— новый автопарк, укомплектованный автомобилями как с АКПП, так и с МКПП.

В-третьих, нелья не сделать акцент на высоком профессионализме педагогического состава, а именно преподаватели по теории ПДД и мастера проивзодственного обучения) благодаря которым, курсанты осваивают полный курс образовательной подгототвки, начиная с вождения на автодроме в п.Цыпка, а получив в последствие водительское удостоверение, продолжают уверенно чувствовать себя рулем собственного автомобиля.

Помимо основной профессиональной программы подготовки водителей категории «В», автошкола проводит обучение по утвержденным курсам:

— курсы квалифицированной подготовки по организации первозок автомобильным транспортом в пределах РФ. Подготовка несомненно затрагивает повышение уровня знаний в обалсти ПДД руководителей различного уровня управления, индивидуальных предпринимателей,

непосредственно осуществляющих организацию транспортной деятельности и перевозки различного масштаба в городе и за его пределами;

— курсы подготовки водителей автотранспортных средств, непосредственно оборудованных специальными световыми и звуковыми сигналами;

— курсы по безопасности правил дорожного движения, которые предусматривают полный курс изучения дорожно – транспортной аварийности, изучение и анализ ситуаций, связанных с повышенной опасностью, организация первой медицинской помощи пострадавшим при ДТП, изучение нормативно-правовой базы, регулирующей дорожное движение, а так же анализ и оценку опасных участков и условий перевозки пассажиров.

Для анализа производства и оказания образовательных услуг дополнительного образования используются данные бухгалтерской отчетности, как помесячной, так и годовой, обрабатываются сведения статистического и оперативного учета в сравнении с предыдущим периодом для каждого вида деятельности или группы услуг в индивидуальном порядке. Анализ динамики объемов производства и реализации услуг осуществляется путем распределения расхождений между текущими показателями по каждому направлению деятельности, согласно категории и данными предыдущего (базового) периода. Клячевой услугой, которую оказывает АНО ДПО Автошкола «Навигатор+» остается согласно лицензии на бессрочной основе обучение профессии водителя категории «В».

Анализ динамики показателей автошколы, влияющих на объем образовательных услуг представлен в таблице 1.1, из которой видно, что объем реализуемых услуг по сравнению с 2022 годом увеличился в 2024 году на 1973,6 тыс.руб., что составило 22,2%.

Среднесписочная численность работающих увеличилась на меньшее значение - на 7,7% или на 2 человека. За счет роста выручки от реализации работ и услуг увеличилась производительность труда на 46,0 тыс. рублей на одного человека.

Таблица 1.1 - Анализ динамики показателей, влияющих на объем оказания услуг АНО ДПО Автошкола «Навигатор+»

Показатели	Ед.изм.	2023 год	2024 год	Отклонения	
				+/- (руб.чел.)	%
Выручка от реализации работ и услуг (без НДС)	тыс. руб.	8895,9	10869,5	1973,6	122,2
Среднесписочная численность работающих	чел.	26	28	2	107,7
Выработка на одного работающего	тыс. руб.	342,2	388,2	46,0	113,5

Финансовый анализ автошколы, как можно четко проследить по данным динамики показателей за 2024 год выявил существенный рост, что, несомненно, указывает на успешное расширение деятельности. Это проявляется в повышении оборотов, росте числа обучающихся, укреплении рыночных позиций и увеличение доли рынка в сфере оказания услуг по обучению водителей легкового транспорта.

От качества программ профессионального обучения, качества методического и технического оснащения самого процесса обучения, качества преподавательской деятельности и освоения практических уроков полностью зависит качество всего обучения в автошколе, что становится принципиально наиболее главным аспектом в деятельности учебного заведения.

1.2 Анализрешения поставленнойзадачи

Создание эффективной и удобной системы для автошколы требует более глубокого изучения всех ее рабочих процессов и определения конкретных потребностей. В ходе проводимого анализа были изучены и рассмотрены более детально текущие операции и система управления, выявлены запросы пользователей и согласованы задачи, которые необходимо решить с помощью потребностей. В ходе анализа мы изучили текущие операции и систему управления, выявили запросы пользователей и определили задачи, которые

может решитьTelegram-бота. Требования, которые были сформулированы к проектируемой системе, в первую очередь затрагивают функции удобства использования, безопасности и эфективности. Анализ деятельности АНО ДПО Автошкола «Навигатор+» является важным, если не сказать больше, этапом в разработке информационной системы на базе Telegram-бота.

Исследования несомненно помогли понячть специфику работы автошкол, выявить как слабые, так и сильные стороны и определить степень помощи бота относительно улучшения наиболее слабых позиций. Первым шагом стало детальная характеристика важных критериев, таких как первоначальная подача заявки на обучение, получение знаний, экзамены как теоретические, так и практические и весь процесс документооборота. Выделим, что раньше для формирования записи на курсы, кандидатам в курсанты приходилось лично посещать автошколу, получать необходимую информацию и оставлять свои персональные и контактные данные для непосредственной связи.

Предлагаемый процесс обучения при внедрении информационной системы будет полностью автоматизирован и предоставит доступ он-лайн. Пользователю достаточно несколько кликов в боте, после чего он получает звонок от специалиста по УР с необходимой информацией для начала работы. Управление расписанием, ранее доступное только по телефону, после разработки будет осуществляться через интуитивно понятный цифровой интерфейс, удобный в пользовании, как самим ученикам, так и преподавателям. Оптимизацию затронет и сама система записи на экзаены. Кандидаты получают возможность записи на практический или теоретический экзамен толкько после освоения требуемого количества часов, согласно программе обучения. Данный факт ни только упрощает и ускоряет сам процесс, но и значительно сокращает очереди, ранее возникающие из-за многократных записей одних и тех же лиц. Кроме того, появляется возможность хранения основного пакета документов: паспорта, СНИЛС и медицинской справки в электронном формате прямо непосредственно в блте. Достаточно для этого отправить номер документа и сотрудник автошколы проводит его верификацию на соответствии

подлинности.После внедрения программного решения получение всей информации для обучения станет значительно проще, благодаря Telegram-боту, который приобретает роль централизованного источника данных. Для разработки эффективной системы были проанализированы и рассмотрены потребности всех пользователей автошколы без исключения, начиная учеников и заканчивая администрацией автошколы. Приоритетными критериями оказались удобство использования, безопасность в плане конфидициальности персональных данных, скорость обработки информации, доступность и надежность.

Оценка и детальное изучение текущей системы управления автошколой выявили ряд существующих на протяжении длительного времени проблем, включая трудности в планировании теоретических занятий, обработка в ручном режиме учета платежей и организацию процесса тестирования по ПДД. Эти и многие другие недостатки могут быть, безусловно, устранены путем внедрения информационной системы на основе новой концепции, а именно примениителеграм-бота.

Нельзя еще раз не затронуть основные требования относительно проектирования новой системы и сделать акцент на удобство, доступность, прозрачность, безопасность и надежность в использовании. Так же было утсановлено, что система должна быть полностью автоматизированной, чтобы минимизировать ручной труд и придать ускоение выполнению повседневных рабочих задач.Telegram-бот представляет собой приложение, разработанное для платформы Telegram и позволяющее в полном объеме пользователям взаимодействовать посредством сообщений и функционала чата. Боты могу выполнять широкий спектр задач, начиная от управления информацией и обработки заказов, до проведения и анализа платежей[15,с.108].

В настоящее время телеграм-боты широкое применение нашли в бизнес-процессах многих учреждений, охватив такие сферы деятельности, как банковское дело, общественное питание, здавоохранение, авиаперевозки и электронная коммерция. Компании активно внедряют боты для оптимизации

клиентского сервиса и повышения эффективности внутренних операций. Рассмотрим наиболее важные преимущества использования Telegram-ботов при разработке концепции информацинной системы [3,с.59]:

— доступность. Телеграм-бот доступен для всех пользователей мессенджера Telegram, что делает его очень удобным для использования;

— простота использования. Бот позволяет пользователю взаимодействовать с системой без необходимости вводить команды вручную;

— быстрота работы. Телеграм-бот обрабатывает запросы мгновенно, что позволяет пользователям быстро получать необходимую информацию;

— автоматизация процессов. Телеграм-бот позволяет автоматизировать процессы, такие как запись на занятия, управление расписанием, прием и обработка платежей, тестирование и т.д;

— безопасность. Все данные, передаваемые через телеграм-бот, защищены с помощью протокола шифрования TLS;

— надежность. Телеграм-боты работают на серверах Telegram, что обеспечивает их высокую надежность и стабильность;

— возможность многократного использования. Телеграм-бот может использоваться не только для создания Информационной системы «Автошкола», но и для других систем;

— низкие затраты. Создание телеграм-бота требует намного меньше времени и денег, чем создание приложения;

— гибкость. Телеграм-бот может быть настроен на различные задачи и требования пользователей;

— интеграция. Телеграм-бот может быть интегрирован с другими системами и приложениями;

— универсальность. Телеграм-бот может использоваться для различных целей, включая образование, бизнес, развлечения и т.д.

— широкий функционал. Телеграм-бот может предоставлять пользователю широкий спектр функций, таких как оплата, отправка файлов,

рассылка сообщений и т.д;

— возможность персонализации. Телеграм-бот может быть настроен на индивидуальные потребности пользователя;

— улучшение клиентского сервиса. Телеграм-боты могут значительно улучшить клиентский сервис, предоставляя быстрый и эффективный способ связи с компанией. Клиенты могут легко получать ответы на свои вопросы, заказывать услуги и получать поддержку в любое удобное для них время.

Проведенный анализ деятельности автошколы выявил потенциал для оптимизации ряда обозначенных требований и пожеланий посредством внедрения информационной системы на базе телеграм-бота обусловлена комплексом современных факторов.

Создание и внедрение информационной системы на базе Телеграм-бота для автошколы открывает больше возможностей для принципиального улучшения качества обучения и оптимизации управления всем учебным процессом. Актуальность разработки такой системы обусловлена несколькими причинами [21,с.207]:

В первую очередь, автошколы играют ключевую роль в обеспечении безопасности дорожеого движения, но их эффективное управление часто затруднительно по причине именно отсутствия автоматизации процесса.

Во-втрых, именно современные технологии и в частности, Телеграм-бота, способны значительно упростить и ускорить бизнес-процессы, включая и запись на занятия, и процесс планирования графика обучения, и обработку платежей, и, соответственно итоговое тестирование по ПДД.Третий аргумент в пользу проекта заключается в повсеместном распространении телеграм–ботов, которые уже активно применяются в бизнесе и производстве.

Их ценность определяется непосредственно простотой и оперативностью при получении сведений или информации, а также возможностью ведения диалога с потенциальными клиентами и контрагентами. Принимая во внимание тенденцию к бесконтактному взаимодействию, разработка программного решенияна базе телеграм-бота для автошколынесомненно позволит

обучающимся удобный и наиблее безопасный способ управления своими знаниями и всем процессом получения учебного материала.

Таким образом, проект обладает значительной актуальностью и несомненной пользой как для пользователей, так и для совершенствования образовательных учреждений в целом [7,с.107].

1.3Оценкааналоговинформационныхсистем

Обзор аналогов играет ключевую роль в процессе создания информационной системы. Именно он открывает возможность детального изучения аналогичных продуктов, выявления как сильных, так и слабых сторон.В ходе проводимых исследований был исследован и тщательно изучен рынок информационных систем, которые применяют автошколы для того, чтобы подобрать принципиально подходящее по требования автошколы «Навигатор+» информационное программное решение.

После тщательного изучения было выявлено, что полноценная информациронная система на базе телеграм-бота отсутствует, что нацеливает на целесообразность проведения сравнительного анализа существующих мобильных приложений и веб-ресурсов.

На данный момент существует несколько похожих приложений. Наиболее популярны из них: «Автошкола-контроль», «Автошкола Онлайн», «ИСО Профтех» являются наиболее распространенным сервисом для автоматизации работы автошкол (рисунок 1.4).

«ИСО Профтех» - это интерактивная система обучения, предназначенная для работы учеников в автошколах. Также система предоставляет сервисы для администраторов автошкол, такие как система для внутренних сообщений ученикам, просмотр статистики тестирования, редактирование учебного плана. Наиболее интересным в курсе является наполнение материала за счет видеоуроков, конспектам расширенного ракурса по всем темам [12].

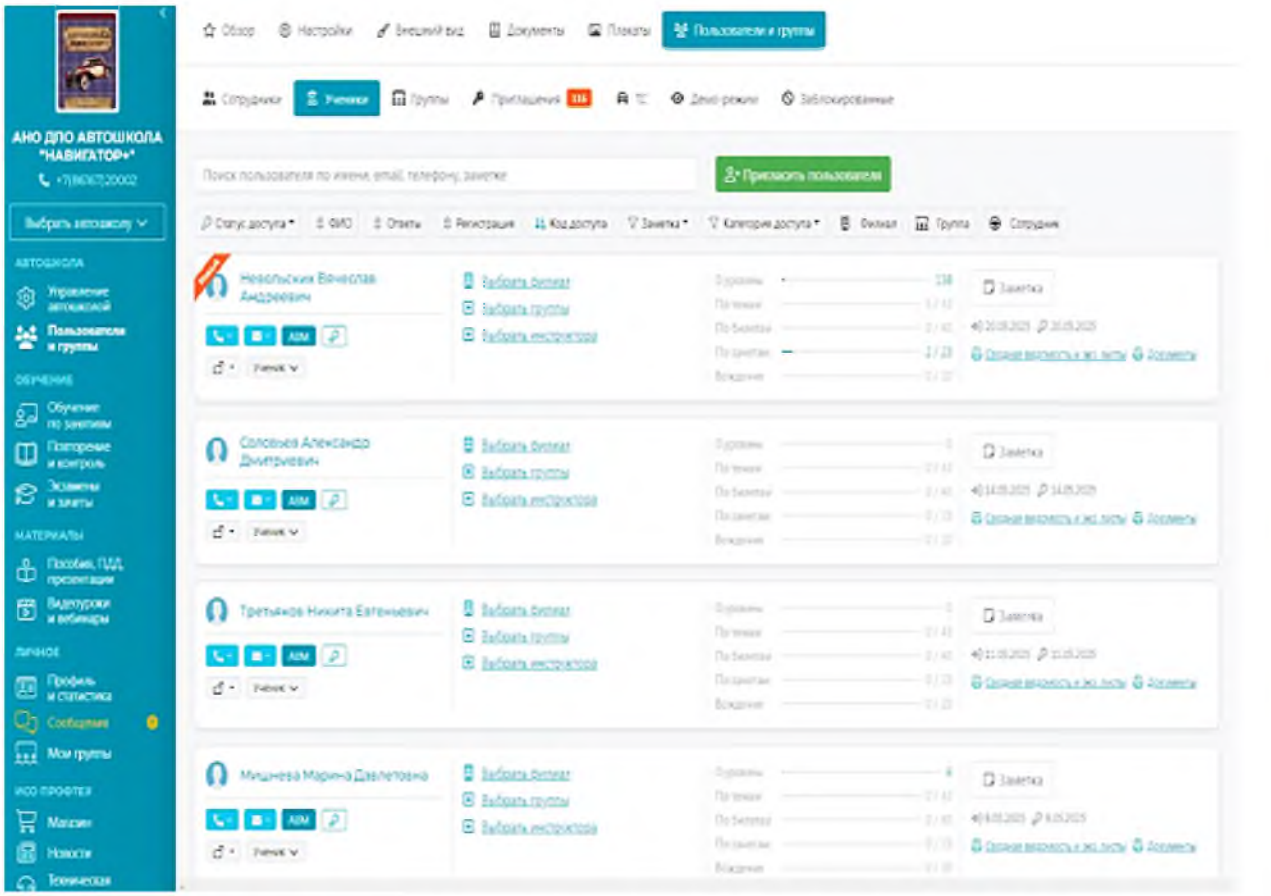


Рисунок 1.4 – Портал электронного обучения АНО ДПО Автошкола «Навигатор+»

Второй пример представлен мобильным приложением «Автоконтроль», которое пользователям возможность записи на занятия, предоставляет систему управвления расписанием в своей учебной группе и тем самым, получать все необходимые сведения о ходе обучения. Тем не менее, данное приложение не имеет в своем функционале файла приема и обработки платежей, а также не охватывает итоговый результат тестовых заданий по правилам дорожного движения.

Саму работу приложения «Автоконтроль» можно увидеть на рисунке 1.5. Среди возможностей для курсантов можно выделить следующие позиции: расписание на полный курс обучения, возможность регистрации на теоретические и практические уроки, курсы он-лайн, подготовка по наиболее актуальным билетам и темам ПДД, учет времени и числа проведенных занятий

по практическому обучения, а так же доступ к уже проведенным теоретическим занятиям, маршрутам и полной нормативно-правовой информации об автошколе. Что касается инструкторов то для них представлена возможность составления индивидуального расписания занятий, виден профиль каждого курсанта, закрепленного за мастером ПО, обозначены маршруты обучения, запись и корректировка занятий, отправка уведомлений, общение с коллегами и учениками, а также открыт доступ к билетным формам и материалам по ПДД.

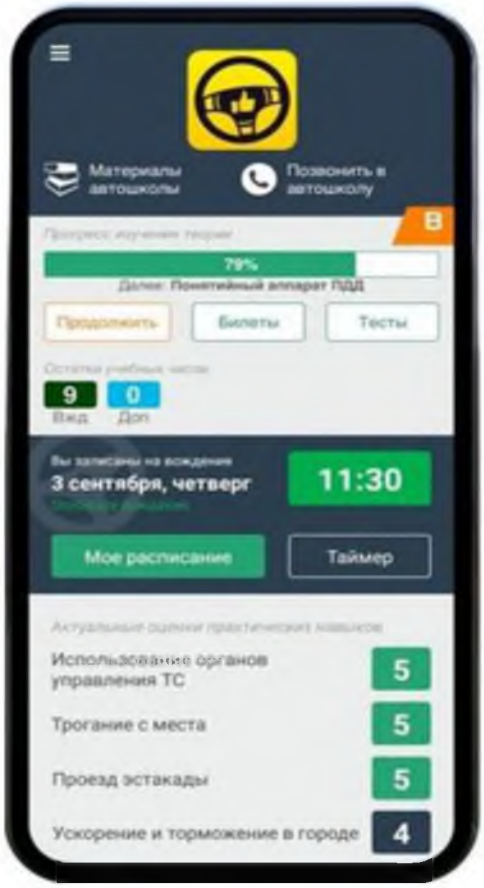


Рисунок 1.5 - Обзор аналога «Автоконтроль»

Далее перейдем к изучению следующей аналогичной информационной системы«AVTO-ONLINE PRO» (рисунок 1.6), которая позволяет автоматизировать сам процесс вождения автотранспортного средства.

Приложение функционирует в тесной интеграции с мобильным софтом. Он-лайн портал предлагает комплекс возможностей по изучению правил дорожного движения в удобном для пользователя формате. Благодаря простому и интуитивно понятному интерфейсу портал намного упрощает процесс

прохождения тестовых упражнений и заданий. Для успешной реализации проектаважное значение имеют полнота и качество предъявляемых требований, которые должны быть реализуемыми, тестируемыми и содержать исчерпывающую информацию.

К достоинствам можно отнести следующее:

— наличие личного кабинета;

— возможность изучать теоретический материал;

— наличие обратной связи с автошколой

Из недостатков выделяется следующее:

— изучение теоретической части;

— просмотр статистики прохождения билетов;

— отсутствие возможности прохождения билетов по тематическим блокам.

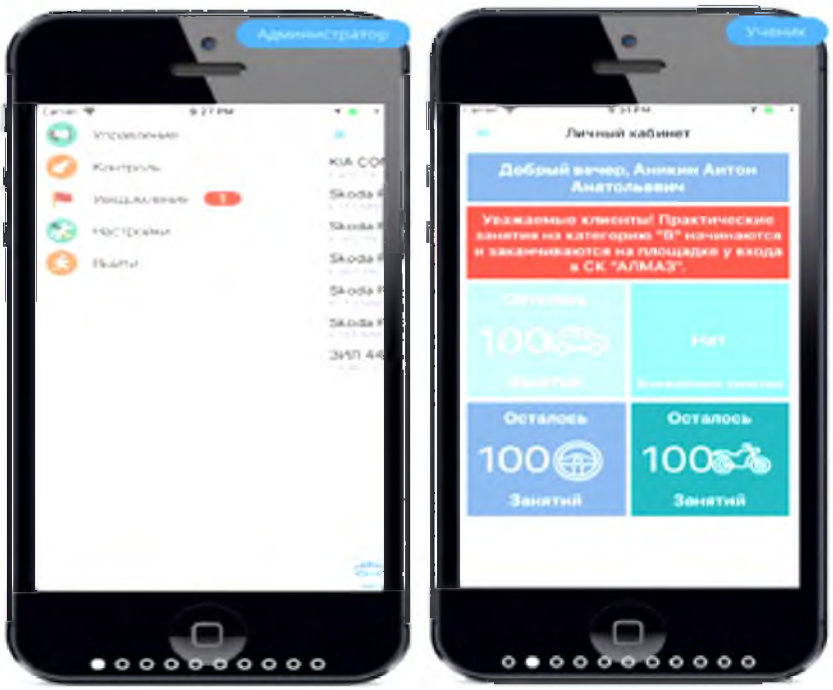


Рисунок 1.6 – Обзор аналога «AVTO-ONLINE PRO»

Выделяют функциональные и нефункциональные требования [7, с.209]:

— функциональные требования – список задач, которые должно решать само программное обеспечение. Они определяют действия системы и какова реакция системы на различные входные данные;

— нефункциональные требования определяют свойства и характеристики системы, не относящиеся к ее поведению. Они включают производительность, удобство сопровождения, надежность, безопасность и другие критерии.

Изучение характеристик существующих в настоящее время аналогов информационных приложений необходимо для более детального определения самого объема работ и как следствие, выявления сильных и слабых сторон существующих решений. Такой глубокий анализ позволяет получить заключение об уникальности разрабатываемой системы в сравнении с существующим.

Кроме того, именно данный продукт оказывает помощь в принятии более обоснованного решения о целесообразности разработки собственной системы, которая будет полностью соответствовать правилам и требованиям организации, устраняя недостатки аналогичных программных продуктов.

Сравнение аналогов и проектируемой информационной системы приведено в таблице 1.2.

Таблица 1.2 – Анализ характеристик аналогов и разрабатываемой системы

Название параметра	«Автоконтроль»	«AVTO-ONLINE PRO»	ИСО Профтех»	Проектируемая ИС автошколы на основе телеграм-бота
Личный кабинет	Да	Да	Да	Да
Прохождение тестов	Да	Да	Да	Да
Запись на практические занятия	Да	Нет	Нет	Да
Просмотр пройденных тестов	Нет	Нет	Да	Да
Изучение теоретического материала	Нет	Нет	Да	Да
Прохождение тестов по блокам	Нет	Нет	Да	Да

Создание полноценной информационной системы на базе телеграм-бота конкретно для использования в автошколе имеет несомненно ряд значительных преимуществ в сравнении с уже существующими вариантами. Прежде всего, бот отличается как удобством и оперативность обработки данных, так и преимуществом работы без скачивания и установки отдельного приложения, что предполагает еще введение логина и пароля на сайте. Сам доступ к системе осуществляется непосредственно через мессенджер, что буквально повышает его практичность, однозначно делая наиболее удобным инструментом для пользователей.

Основной целью проектирования является упрощение процесса обучения и повышения его качества за счет автоматизированного процесса наиболее привычных повседневных задач, а также усиление процесса коммуникативного процесса между участнрками образовательного процесса. Сама информационная система должна быть легкой и удобной в использовании, обладать качествами надежности и безопасности данных.

Кроме того, она должна быть в обязательном порядке масштабируемой, чтобы ее легко можно было расширять в будущем по функциональным возможностям. В итоге, информационный продукт должен быть простым в обслуживании и экономически эффективным с минимальными затратами на разработку и техническую поддержку.

Применение информационной системы должно увеличить количество обучающихся и поднять уровень конкурентоспособности, сделав ее узнаваемой и успешной на рынке. Разрабатываемое программное обеспечение актуально для частных автошкол, которые стремяться получить необходимый функционал, что послечет за собой повышение спроса данного учебного заведения с минимальными затратами.

2 Проектная часть

2.1 Специфика разработки информационной системы

Специфика разработки информационной системы (ИС) включает в себяразличные аспекты: этапы, методику, инструмены и технологии. При этом несомненно следует принимать во внимание специфику сферы применения продукта.В частности, при проектировании ИС для автошколы необходимо обеспечить ее конкурентоспособный уровень на рынке образовательных услуг дополнительного образования и всячески способствовать достижению успеха. Современные программные приложения и информационные программы настолько развиты, что терминология «архитектура» стала неотъемлемой частью их описания. Построение корректной, эффективной и надежно функционирующей информационной системы является сложной задачей.

Под архитектурой систем положено понимать совокупность решений относительно [24,с.98]:

— организации программной системы;

— выбора структурных элементов, составляющих систему и их интерфейсов;

— объединение этих элементов в подсистемы;

— поведения этих элементов во взаимодействии с другими элементами данной системы;

— архитектурного стиля, который определяет логическую и физическую организацию системы.

Архитектура программной системы охватывает не только структурные аспекты ее построения, но также определяет правила использования и интеграфию с другими системами. Кроме того, она вклбючает в себя характеристики функциональности, производительности, гибкости и надежности в использовании. В сферу архитекутры входят также полнота функциональных возможностей, технологические ограничения и дизайн

пользовательского интерфейса.

Принято выделять следующую классификацию программных систем по их архитектуре [18,с.73]:

— централизованная архитектура;

— архитектура «файл-сервер»;

— двухзвенная архитектура «клиент-сервер»;

— многозвенная архитектура «клиент-сервер»;

— архитектура распределенных систем.

В целях успешной реализации проекта в рамках выпускной квалификационной работы принято решение использовать архитектуру «клиент-сервер». Данный выбор обусловлен широким применением этой архитектуры непосредственно в web-приложениях. Необходимо отметить, что именно серверная часть подобных приложений часто реализуется при непосредственном использовании различной конфигурации архитектурных решений и обладает специфическим набором определенных возможностей, таких как, напрмер:

— отсутствует необходимость устанавливать и использовать дополнительное ПО на стороне клиента;

— используя единое место хранения данных обеспечиваются минимальные требования для поддержания целостности данных;

— работоспособность сервера и каналов связи определяют доступность и недоступность;

— архитектура web-систем не имеет существенных ограничений и другие.

Всё это дает возможность довольно быстро и просто изменять приложение. Далее необходимо перейти к выбору паттерна проектирования. Рассмотрим паттерны, предназначенные для проектирования web - приложения. К ним относятся [23, с.167]:

- модель-представление-контроллер(ModelViewController);

— контроллер страниц (Page Controller);

— преобразователь (Transform View);

— представление по шаблону (Template View);

— контроллер запросов (Front Controller);

— контроллер приложения (Application Controller);

— двухшаговая шаблонизация (Two Step View).

Таблица 2.1 содержит описание основных шаблонов проектирования веб-приложений. Следует принимать во внимание, что не существует единого шаблона, способного решить все проблемы проектирования. При создании системы необходимо отбирать шаблон, наиболее соответствующий конкретной области проектирования [9,с.71].

Таблица 2.1 – Web-паттерны представления данных

Название	Описание
Модель-представление-контроллер (Model View Controller)	Представление отображает содержимое модели средствами графического интерфейса. Контроллер получает входные данные от пользователя, выполняет операции над моделью и указывает представлению на необходимость соответствующего обновления. В этом плане графический интерфейс можно рассматривать как совокупность представления и контроллера (рисунок 2.1).
Контроллер страниц (Page Controller)	В основе контроллера страниц лежит идея создания компонентов, которые будут выполнять роль контроллеров для каждой страницы веб-сайта. На практике количество контроллеров не всегда в точности соответствует количеству страниц, поскольку иногда при щелчке на ссылке открываются страницы с разным динамическим содержимым (рисунок 2.2).
Преобразователь (Transform View)	Преобразует записи в HTML по одной. Когда выполняются запросы к БД, то можно получить данные, но этого недостаточно, чтобы отобразить нормальную web-страницу. Использование Transform View подразумевает преобразование, когда на входе есть модель, а на выходе HTML.

Продолжение таблицы 2.1

Контроллер запросов (Front Controller)	Контроллер запросов обрабатывает все запросы, поступающие к Веб-сайту, состоит из двух частей: Веб- обработчика и нерархии команд. Веб-обработчик – это объект, выполняющий фактическое получение POST- или GET- запросов, поступивших на Веб-сервер. Веб-обработчик обычно реализуется в виде класса, поскольку он не генерирует никаких откликов. Команды также являются классами; более того, им не нужно знать о наличии Веб-окружения, несмотря на то, что им часто передается информация из HTTP-запросов(рисунок 2.3).
Представление по шаблону (Template View)	Основная идея, лежашая в основе типового решения представление по шаблону, – вставка маркеров в текст готовой статической HTML-страницы. При вызове странишы для обслуживания запроса эти маркеры будут заменены результатами некоторых вычислений. Подобная схема позволяет создавать статическую часть страницы с помощью обычных средств и не требует знания языков программирования. Для получения динамической информации маркеры обращаются к отдельным программам (рисунок 2.4).

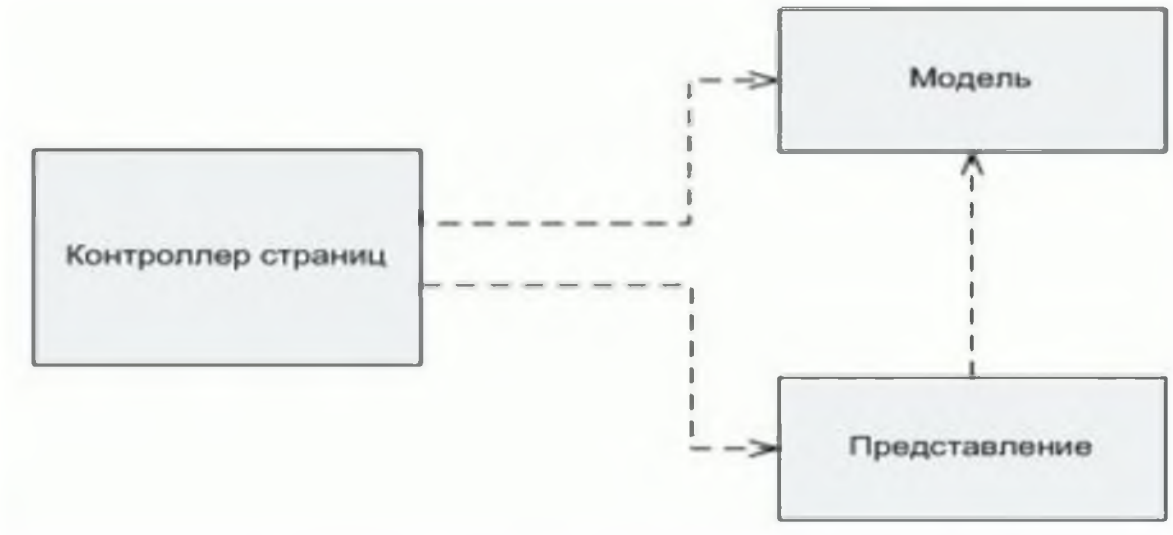


Рисунок 2.1 - MVC-паттерн

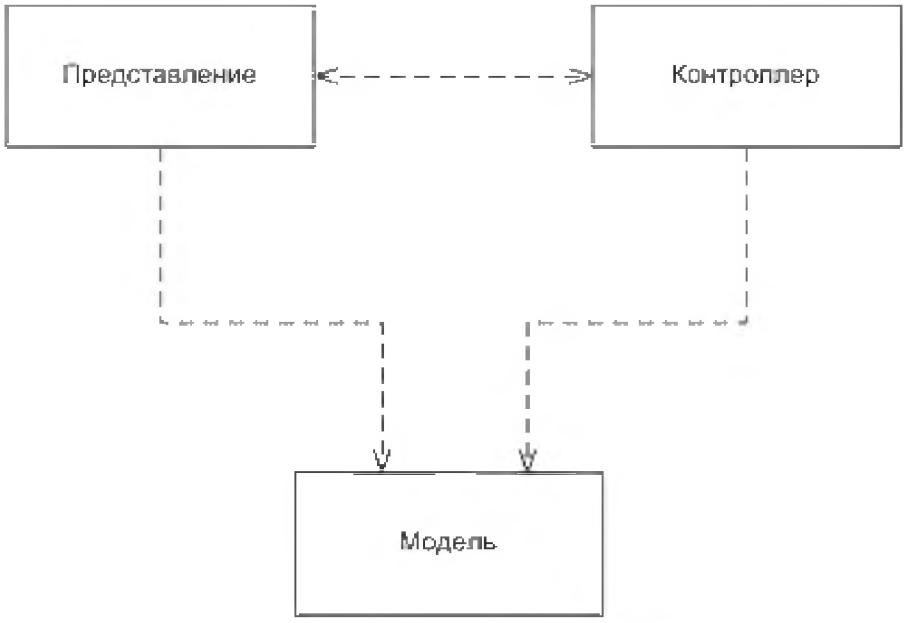


Рисунок 2.2 - PC-паттерн

Рисунок 2.3 – FC-паттерн

Рисунок 2.4 - TV-паттерн

Разработка телеграм-ботов в первую очередь описается в своем проекте

на определенные щаранее согласованные шаблоны, которые структуризируют обработку поступающей информации и сообщений, команд и определенного плана событий. Данные шаблоны оптимизированы для взаимодействия с APITelegram и охватывают как частичную, так и групповую переписку.

При создании телеграм-ботов применяют специальные модели проектирования, направленные в первую очередь несомненно на упорядочивание обработки самой информации. Эти модели ведут учет специфики взаимодействия с APITelegram и предусматривают возможность работы бота в индивидуальных и групповых чатах[5,с.155].

Для того чтобы реализовать на практике проект был выбран MVC паттерн. MVC (Model-View-Controller) – это паттерн, возможностями которого является разделение логики приложения на три составные части:

— Model - представление данных и бизнес-логики (пользовательские профили, состояния диалогов);

— View - форматирование ответов пользователю (текст, кнопки, карточки);

— Controller - обработчики событий, которые принимают входящие сообщения.

Основная идея MVC: контроллер перехватывает событие извне и, в соответствии с логикой, изменяет модель. После этого модель отправляет событие об изменении, и все подписанные на него представления обращаются к модели за обновлёнными данными и отображают их [11].

Применение данного шаблона позволяет создавать изолированные модули кода, что означает процесс происхождения изменений в одном блоке. Данная операция ни в коем случае не повлияет на работоспособность других частей программы. Такое разделение компонентов соответствует принципу единственной ответственности, что, несомненно, упрощает действия отдельных элементов.

Выбор этого паттерна обусловлен его интуитивной понятностью,

простотой внедрения и ориентацией на создание гибких, масштабируемых и легко поддерживаемых приложений с возможностью дальнейшего расширения функциональных свойств.

Далее рассмотрим логическую модель, которая представляет собой абстрактное описание структурных данных, служащее основой для проектирования БД.

Логическая модель определяет организацию данных и их взаимосвязи, оставаясь при этом независимой от конкретной системы управления базами данных (СУБД) на протяжении всей работы.

Логическа модель служит для отображения внутренней структуры предметной области, относящейся к определенной задаче. Ее разработка базируется на концептуальной модели и предполагает выбор определенной модели данных, при этом игнорируются специфические характеристики целевой системы управления базами данных (СУБД).

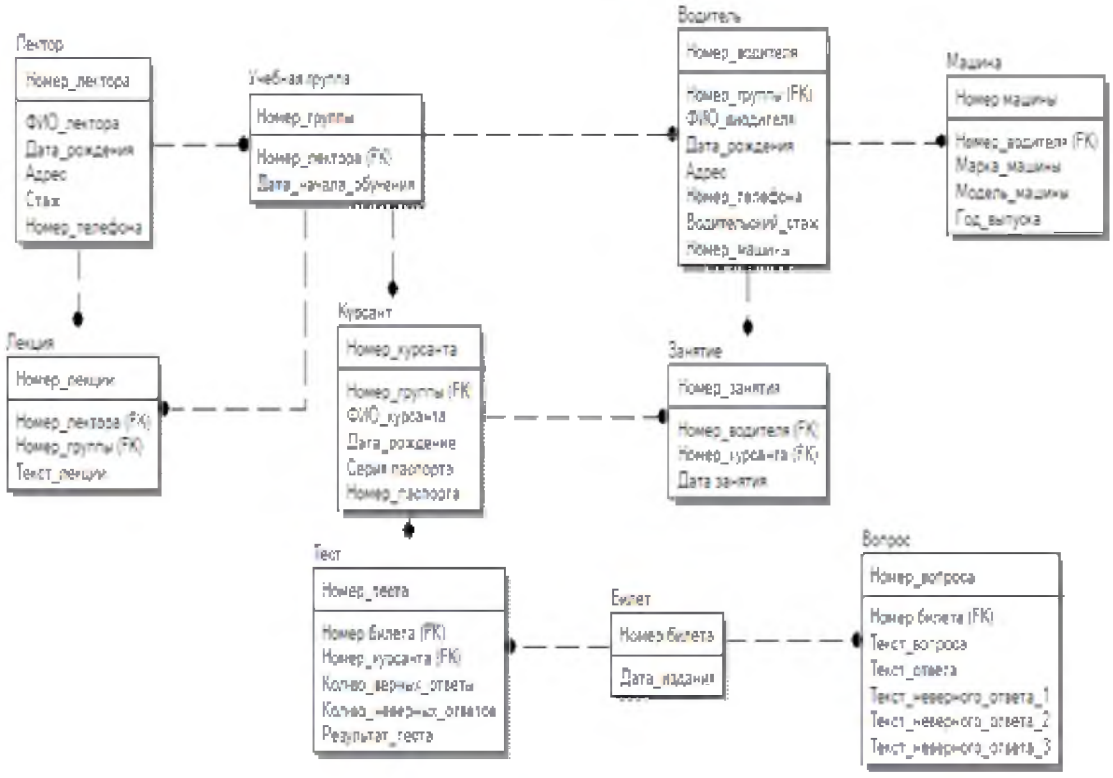


Рисунок 2.5 - Логическая модель базы данных

Построим логическую модель базы данных в нотации IDEF1X при помощи case-средства ERwin Data Modeler (рисунок 2.5).

Полученная логическая модель состоит из 10 сущностей: «Лектор», «Лекция», «Учебная группа», «Курсанты», «Водитель», «Машина», «Занятие», «Тест», «Билет», «Вопрос». Каждая сущность связана с другой неидентифицирующими связями.

Сущность«Лектор» имеетполя,содержащие ФИО лектора, дату рождения, адрес проживания, стаж работы, номер телефона. Сущность «Лекция» содержит идентификатор лектора, идентификатор группы, которой предназначена лекция и сам текст лекции.

Сущность «Учебная группа» содержит поле идентификатора лектора, преподающего теоретические знания ПДД и дату начала обучения. Сущность «Машина», содержит поле идентификатора водителя, который преподает на этой машине, марку машины, её модель и год выпуска. Сущность «Водитель» содержит информацию о водителе, его ФИО, адрес, дату рождения, номер телефона, водительский стаж, номер машины, номер группы.

Сущность «Курсант» состоит из полей, содержащих идентификатор группы, в которой состоит курсант, его ФИО, дату рождения и паспортные данные. Сущность «Билет» содержит номер билета и дату издательства билета. Сущность «Вопрос» содержит поле с номером билета, текст вопроса, текст ответа и 3 неверных ответа. Сущность «Тест» содержит поле с номером билета, поле с номером курсанта, который проходил тест, количество верных ответов, количество неверных ответов и результат теста.

Графический интерфейс пользователя, отличающийся простотой и интуитивностью, был спроектирован с учетом диаграммы вариантов использования и результатов анализа требований.

Для его создания применялись HTML с использованием таблиц стилей CSS. Особое внимание было уделено проектированию интерфейса, как одного из важных показателей системы, ориентированного на конечного пользователя.

2.2 Программные и системные требования при разработке Telegram-бота

Четкое определение архитектуры и стратегии интеграции является ключом к подбору оптимальных инструментов и обеспечивает бесперебойное функционирование бота в заданной среде. В контексте информационной системы для автошколы, построенной на базе Telegram, архитектура включает три основных элемента: клиентское приложение, т.е. сам телеграм-бот, серверное приложение и базу данных.Подробную диаграмму прецедентов можно изучить на рисунке 2.6.

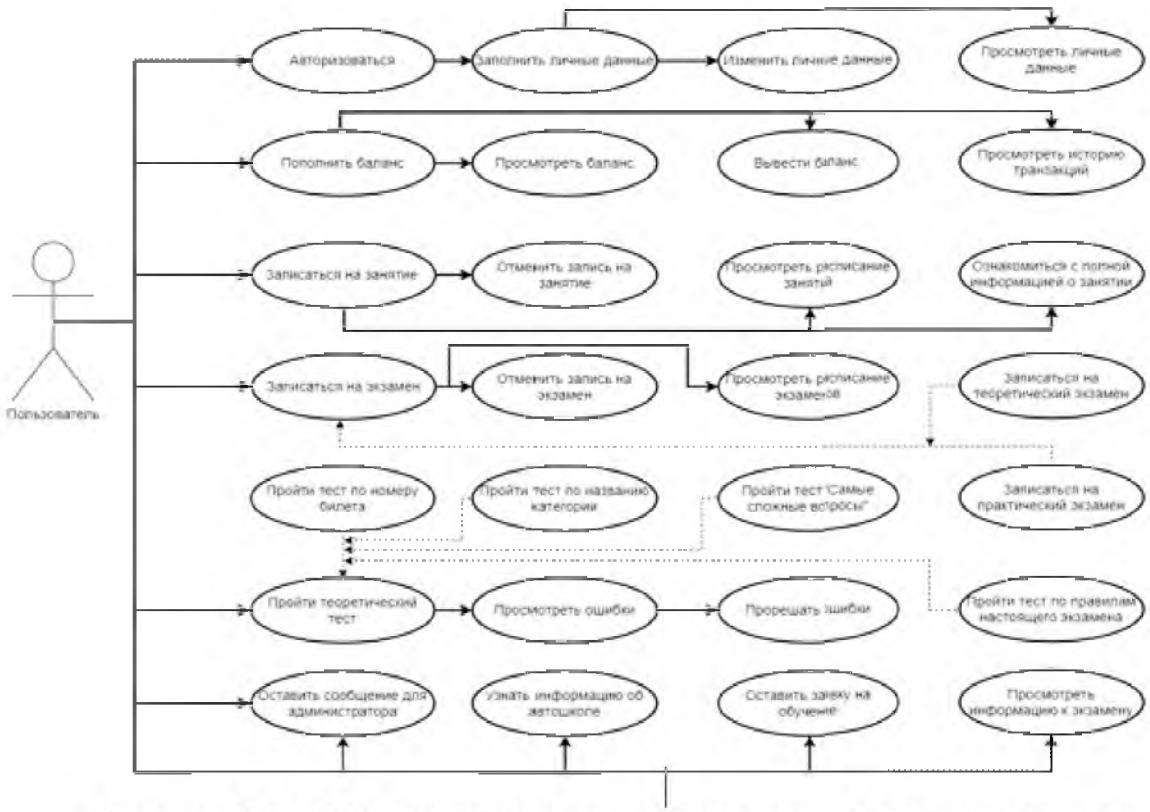


Рисунок 2.6 – UML диаграмма прецедентов интерфейса пользователя

Пользовательский интерфейс для обучающихся реализован в формателеTelegram-бота. Данный бот служит центральным звеном для взаимодействия учеников с платформой, предоставляя им функционал для записи на курсы, осуществление платежей, доступа к материалам экзаменов, прохождения тестовых заданий и отслеживание результатов своего обучения в автошколе. Помимо клиентского модуля система включает в себя отдельные интерфейсы для инструкторов и административного персонала.

Инструменты инструктора предоставляют функциональные возможности для управления непосредственно расписанием, отслеживание успеваемости учеников и коммуникации с ними. Что касается администратора, то его раздел дает возможность полного контроля над всеми пользовательскими данными, охватывая управление расписанием, доступ к учебным материалам, мониторинг финансовых операций и общее функционаирование учебного заведения дополнительного образования. Подробнее диаграммы прецедентов этих интерфейсов можно изучить на рисунке 2.7.

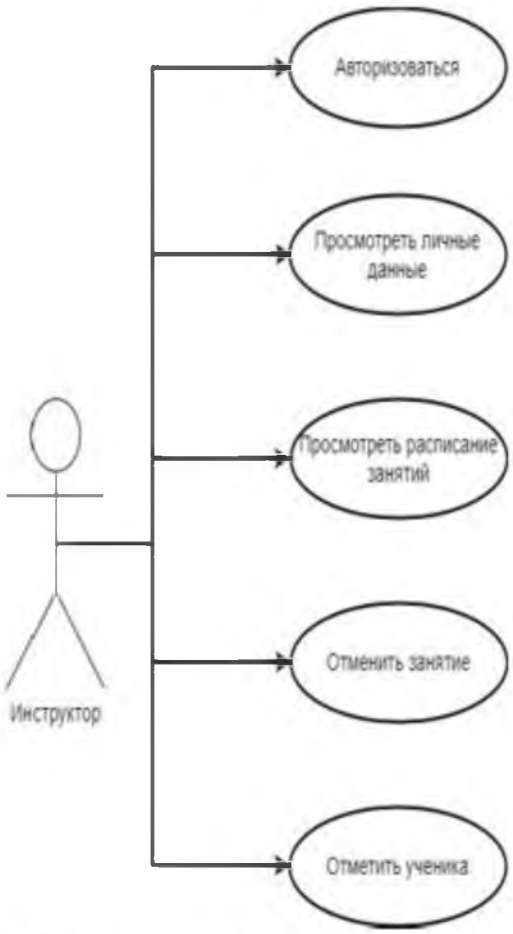


Рисунок 2.7 - UML диаграмма прецедентов интерфейса инструктора

С целью полного понимания взаимодействия компонентов бота, была создана UML- диаграмма последовательности (рисунок 2.8). Диаграмма служит визуальным инструментом, отображающим, как и в какой хронологической последовательности происходит обмен сообщениями между элементами системы[16,с.88].

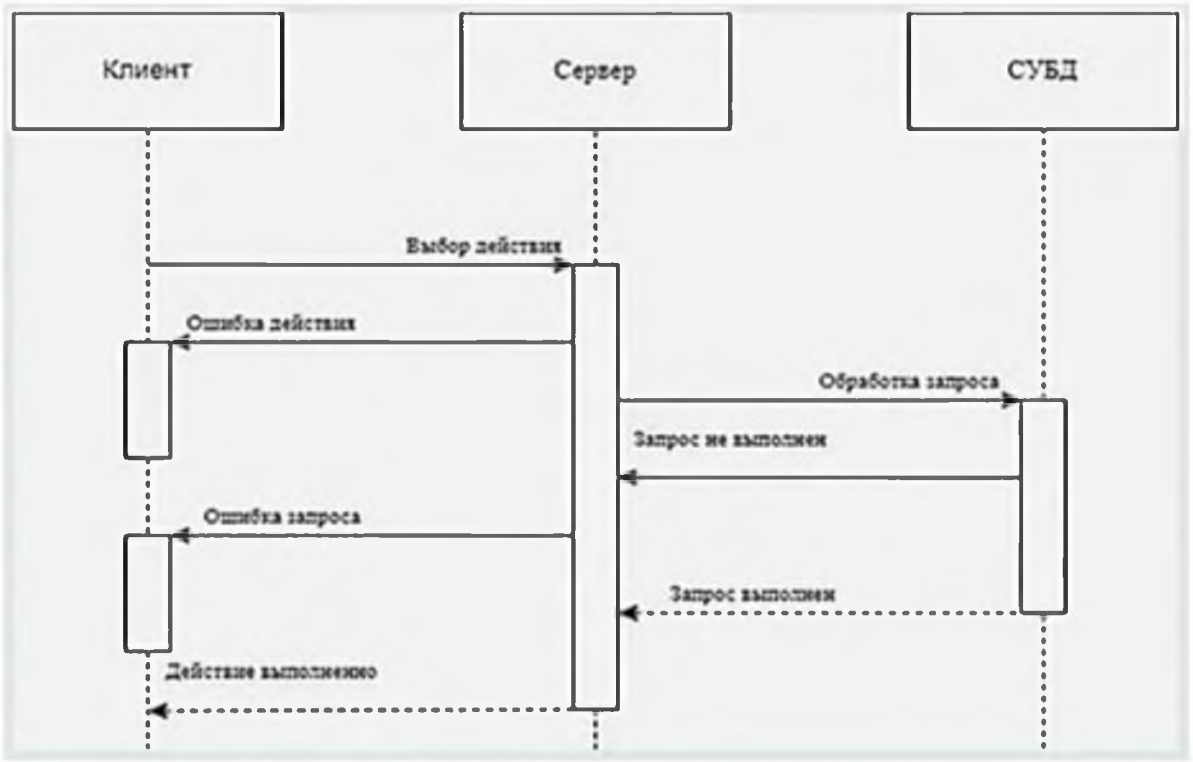


Рисунок 2.8 - UML-диаграмма последовательности

При создании концепции автошколы «Навигатор+» была поставлена основная задача, которая заключается в разработке по своему наполнению удобную и эффективную информационную систему на базе именно Telegram-бота. Система будет призвана сделать процесс обучения легко доступным, интересным, познавательным, и несомненно результативным, сокращая при этом затраты времени и средств. Пользователи данной системы смогут легко регистрироваться на занятия, видеть свои достижения в процессе обучения, общаться с мастерами ПО и получать автоматические чеки о своих платежах, что обеспечит в полном объеме полную прозрачность и надежность всех финансовых операций. При выборе инструментария для проектированияпрограммного продукта первостепенное значение имеет его соответствие специфике самого проектного задания и способность гарантий продуктивности процесса создания.

Корректный подбор определенных средств разработки выступает одним из фундаментальных аспектов при построении и проектировании

информационной системы. Именно такой подход способствует оптимизации полного цикла разработки, минимизации временных и финансовых издержек на создание самого продукта, а так же обеспечивает высокий уровень качества и безопасности на всех этапах.

Разработка телеграм-бота для автошколы открывает широкие возможности благодаря популярности самой платформы и ее технических достоинств. Телеграм гарантирует охват большой аудитории и простоту взаимодействия для пользователей. Кроме того, боты на данной платформе отличаются высокой производительностью, содержат спектр надежных элементов, оснащены компонентами безопасноти и гибкости в интеграции с другими системами, что делает их оптимальными для выбора проекта информационной системы автошколы [4,с.55].

При разработке и масштабировании интернет-проектов критически важным является выбор именно технологического стека, включающего хостинг, систему управления БД и язык программирования. Основными ориентирами при этом должны выступать соответствие функциональным и нефункцитнальнымтребованиям к проекту, а так же наличие у команды разработчиков необходимого набора компетенций и профессионального опыта работы для возможности дальнейшей поддержки выбранных решений. Важно помнить, что выбор хостингатакжеподразумеывает определение операционной системы для серверной инфраструктуры.

Например, Windows-хостинг подходит для веб-проектов, разработанных на базе технологий Microsoft, а Linux-подобные системы предоставляют поддержку для многих языков программирования и других опенсорс-решений [14,с.37].

Перенос информационной системы (ИС) на сервер требует тщательного выбора хостинга, т.е. иными словами определенной услуги, предоставляющей вычислительные ресурсы и пространство в сети Интернет. Основными критериями выбора становятся масштабируемость, доступность, качественная техническая поддержка и, конечно же, экономическая эффективность.

2.3 Технические средства создания концепции ИС

Входе определения системных требований был определен сервис TimeWeb.Cloud, поскольку он предлагает ряд преимуществ, соответствующих моим требованиям. Во-первых, TimeWeb.Cloud обладает высокой надежностью и обеспечивает стабильную доступность моей информационной системы, что критически важно для ее бесперебойной работы.

Во-вторых, этот хостинг предлагает масштабируемое решение, позволяющее легко адаптировать мою систему к возрастающим потребностям и нагрузке. Кроме того, TimeWeb.Cloud обеспечивает высокий уровень безопасности, что гарантирует защиту данных и обеспечивает конфиденциальность информации.

Важным фактором является также качественная техническая поддержка, которую предлагает TimeWeb.Cloud, готовая оказать помощь и решить любые вопросы, связанные с хостингом.

Наконец, главным из преимуществ TimeWeb. Cloud стала доступная цена, что сделало его окончательным выбором с точки зрения соотношения цены и качества предоставляемых услуг [17,с.88].

Выбор системы управления базами данных (СУБД) напрямую зависит от того, какое программное обеспечение будет в перспективе применимо для структурирования, систематизации и хранения взаимосвязанной информации в цифровом формате, а так же для обеспечения удобного доступа к ней.Для Linux-дистрибутивов чаще всего выбирают СУБД MySQL, но популярным решением также является PostgreSQL.

Windows-хостинг поддерживает MySQL, но, как правило, в качестве СУБД в этой системе применяется Microsoft SQL Server.

Один из главных факторов при выборе СУБД является ее совместимость с конкретными требованиями приложения [25,с.166].

Среди наиболее востребовательных и наиболее актуальных СУБД выделяется Oracle, которая нашла широкое применение в корпоративном

секторе и разработке бизнесприложений. Ее сильными сторонами считаются высокая производительность, масштабируемость и надежность, дополненные богатым функционалом, включая управление транзакциями, безопасность и резервное копирование.

Однако, высокая стоимость лицензирования и поддержки может стать существенным препятсвием для наибольших проектов и стартапов, делаа данную СУБД менее привлекательнм выборов в таких случаях.

Среди восстребованных систем управления базами данных (СУБД) выделяется MySQL – свободная и открытая система, которая активно применяется в веб-разработках. MySQL демонстрирует неплохую производительность, однако ее возможности предельно ограничены в плане поддержки транзакций и масштабирования.

В отличие от нее MongoDB также выделяется высокой доступностью и отказоустойчивостью, что делает ее оптимальным решением для приложений, требующих оперативного доступа к информации.

В качестве системы управления базами данных для информационной системы телеграм-бота «Автошкола» была выбрана PostgreSQL.

Данная СУБД с открытым исходным кодом и бесплатным распространением обладает всеми необходимыми характеристиками для работы с крупными и сложными проектными заданиями.

Среди ключевых особенностей PostgreSQL можно выделить грамотную поддержку транзакций, функционал полнотекстового поиска, встроенную репликацию и возможности масштабирования.

Высокая проивзодительность и надежность PostgreSQL позволяют эффективно обрабатывать значительные объемы данных без малейших потерь производительности.

Помимо данных характеристик отметим, что PostgreSQ включает в себя широкую линейку инструментов для работы с данными и признательность сообщества, что значительно упрощает процесс разработки и поддержки создаваемой информационной системы.

Определение языка программирования связано с выбором инструмента разработки. Например, Windows Server больше подходит для разработки приложений на C#, VB.NET, ASP / ASP.NET и использования библиотек экосистемы Microsoft, но она также поддерживает PHP. Linux хостинг предоставляет широкий выбор языков, помимо PHP: Python, Node.js, Perl, Ruby on Rails. Каждый язык программирования обладает своими уникальными особенностями, которые существенно влияют на процесс разработки и конечный результат [29, с.55].

Для создания телеграм-ботов отлично подходит Python – высокоуровневый язык с простым и интуитивно понятным синтаксисом. Его динамическая типизация позволяет автоматически определять тип переменной в процессе выполнения кода. Помимо Python, для разработки ботов также могут использоваться и други языки программирования, например, такие как[30]:

— JavaScript – популярный язык для создания веб-сайтов и приложений, который также отлично подходит для разработки телеграм-ботов. Он относится к языкам с динамической типизацией и обладает синтаксисом, напоминающим С.В отличие от Python, JavaScript не имеет встроенной поддержки матриц и научных вычислений, но зато предлагает обширный набор инструментов для работы с веб-интерфейсами и серверным взаимодействием;

— PHP – это широкий в применении язык для проектирования веб-сайтов и приложений, который так же может быть использован для разработки телеграм-ботов. Как и JavaScript его можно отнести к языкам с динамической типизацией по причине того, что он имеет схожий сС синтаксис. Однако, в отличие от Python, PHP не может выделиться в своей структуре таким же количеством научных библиотек, так как в основном ориентирован на веб-разработку;

— Java относится к популярным языкам программирования, который наиболее чаще остальных выбирают для создания приложений под Android и другие мобильные платформы. Он относится к тем языкам, которые представляют строгую типизацию и внешне напоминают C++. Главным

отличием от Python скорей всего можно считать первоначальное выполнение компиляции, а уже после этого следует само выполнение, что в принципе ускоряет саму работу. Но в наличие можно выделить и главный минус – затруднение процесса разработки телеграм-ботов в сравнении с Python в написании больше кодов.

Выбор Python в качестве языка программирования для разработки телеграм-бота был продиктован его неоспоримыми преимуществами. Во-первых, его простой и понятный синтаксис значительно повышает скорость написания кода и облегчает тем самым дальнейшее сопровождение.

Кроме того, богатый набор библиотек и фреймворков, в частности, python – telegram – bot, который значительно упрощает взаимодействие с TelegramAPI, существенно ускоряет процесс разработки и дает возможность эффективного решения даже нетривиальных задач.

Высокаяпроивзодительность языка программирования является важным фактором, обеспечивая быструю обработку данных и выполнение вычислений.

Благодаря широкой популярности и активному сообществу, Python постоянно развивается и усовершенствуется, предлагая все новые и новые инструменты и решения. Таким образом, Python является идиальным выбором в качестве языка программирования с целью создания чатботов, предоставляя тем самым полный спектр возможностей для работы cAPI, обработки информации и формирования пользовательского опыта.

2.4 Проектирование базовой архитектуры ИС

Архитектура информационных систем (ИС) определяет фундаментальное устройство: как компоненты системы взаимосвязаны между собой и внешним миром, а так же определяют правила, которыми руководствуются при создании, внедрении,развитии и поддержки ИС. В контексте Telegram-бота, его архитектура представляет собой комплексную структуру, которая охватывает выбор технологий, способов взаимодействия и инфраструктуры. От этой

структуры напрямую зависит скорость работы, способьность к росту и надежность всего бота. Регистрация бота – это перначальный этап его создания и подключения к информационной системе. Для этого требуется создание нового аккаунта через официальное приложение Telegram на мобильном устройстве или персональном компьютере.

Главное меню показано на рисунке 2.9.

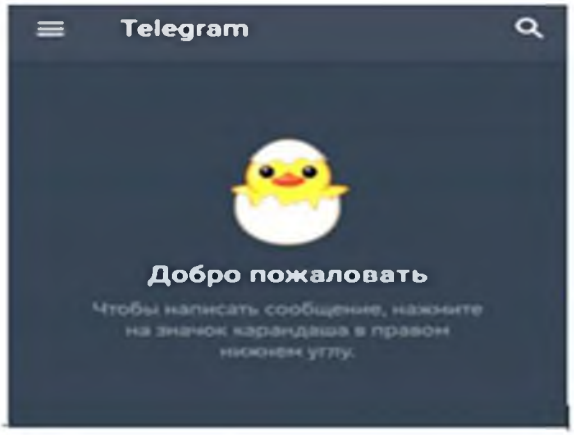


Рисунок 2.9 – Главное меню

Далее нужно найти в поиске Telegram - бота с именем «BotFather»- это официальный бот Telegram для создания новых ботов (рисунок 2.10).

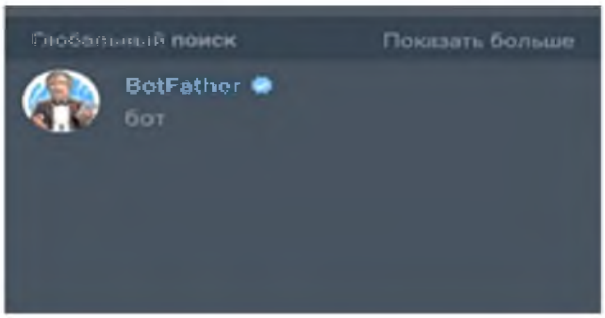


Рисунок 2.10 – Поиск BotFather

Процесс создания бота начинается и использования компанды /newbot, после чего необходимо выбрать имя бота, которое будет использоваться для его идентификации в поиске и задать логин, который определит его уникальную веб-ссылку (рисунок 2.11).

Рисунок 2.11 – Выбор имени бота

После успешного создания бота путем применения BotFather, получаем уникальный токен, который будет служить идентификатором для созданного бота и необходим при установке связи с TelegramAPI, что обеспечивает его аутенфикацию. Визуальное представление данных, полученных от бота, представлено на рисунке 2.12.

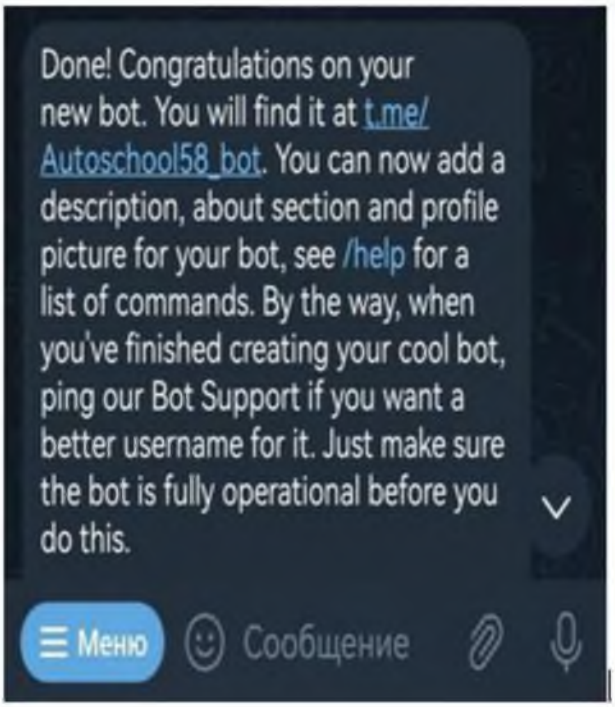


Рисунок 2.12 – Получение ссылки на бот

После успешной регистрации телеграм-бота, его токен становится доступен для настройки и интеграции с информационной системой АНО ДПО

Автошкола «Навигатор+». Проектирование бота в контексте работы с базой данных включает следующее: выбор СУБД, определение методов доступа к данным, демонстрацию кода и обеспечение безопасности предоставления сведений.

Для хранения информации необходимо реализовать базу данных со следующими таблицами, которые будут служить для хранения всех имеющихся данных, а также структуры базы (поля, их типы и свойства) [26,с.173]:

— applications - таблица, содержащая информацию о заявках студентов;

— exam_pdd - таблица, содержащая информацию о всех вопросах для экзаменационного теста;

— students - таблица, содержащая информацию об активных студентах;

— payment - таблица, содержащая информацию о проведенных транзакциях;

— cars - таблица, содержащая информацию о машинах, находящихся на учете автошколы;

— groups - таблица, содержащая информацию о группах обучения;

— students_of_exam - информация об экзамене;

— teachers- информация об инструкторах автошколы;

— lesson - информация о типах занятий;

— class - информация о местах проведения занятий;

— category- информация о категориях обучения;

— timetable_teory - расписание теоретических занятий;

— timetable_practic - расписание практических занятий;

— program - информация о программах обучения;

— gibdd_exam - информацию о датах проведения экзамена, сформированная МРЭО № 2 ГИБДД;

— result_quiz - информацию о прохождении тестирования в боте.

Для наглядного понимания работы БД, на рисунке 2.13 представлена схема физической модели базы данных.

Информационная система предполагает реализацию таких функций, как:

— сбор заявок;

— авторизация;

— управление и просмотр данных;

— управление и просмотр расписания;

— подготовка к экзамену;

— просмотр материалов нужных для сдачи теоретического и практического экзамена;

— управление и учет транзакций;

— информирование по mail.

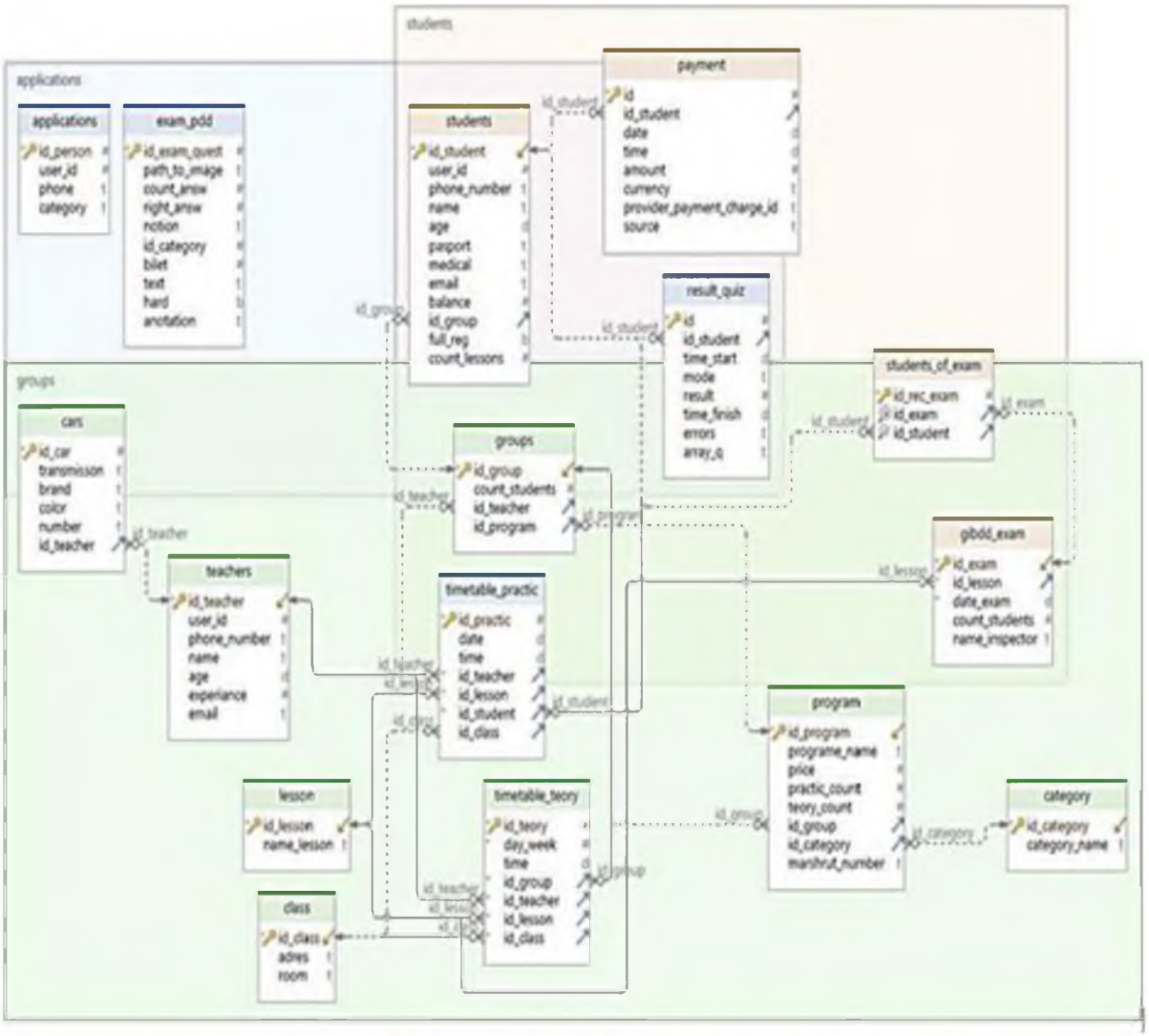


Рисунок 2.13 - Физическая модель базы данных

Чтобы начать работу с ботом, первым шагом будет его поиск. Варианта два: либо перейти по ссылке, которая будет предоставлена, либо воспользоваться поиском, как показано на рисунке 2.14.

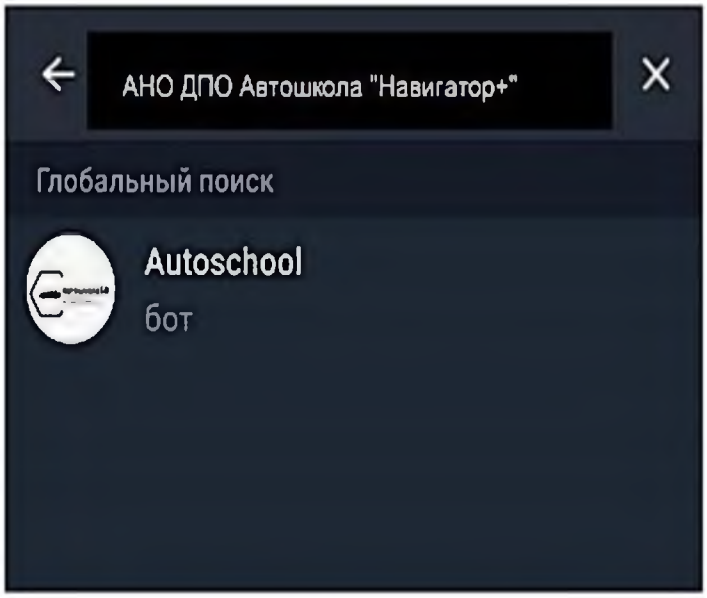


Рисунок 2.14 – Поиск бота

Для начала работы бот делает запрос на номер телефона для авторизации, что показано на рисунке 2.15. Данный процесс полностью обеспечивает контроль над регситарцией пользователей автошколы. Также, нажав на соответствующую кнопку, появляется возможность получения сведений об автошколе. Принцип применения и работы данной кнопки представлен далее.

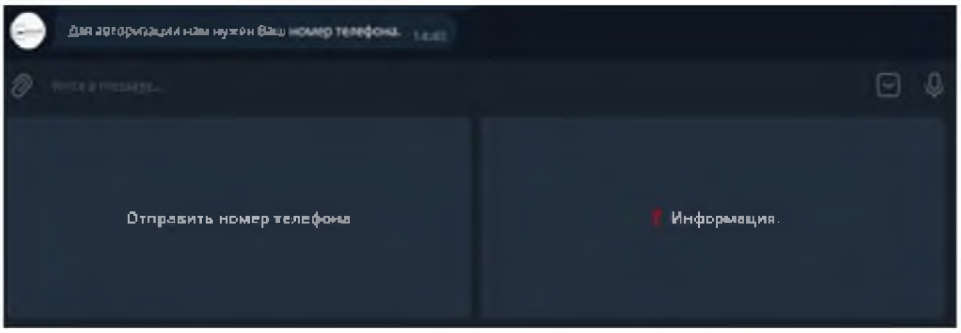


Рисунок 2.15 – Авторизация пользователя

В случае отсутствия пользователя в перечне, закрепленном за автошколой, система уведомит его об этом, что представлено на рисунке 2.16 и предложит оформить заявку на обучение. Для последующей автоматической

обработки заявки, пользователю необходимо осуществить соответствующий выбор категори обучения. После совершения данного выбора, администратор рассматривает заявку и пользователь автоматически включается в реестр обучающихся данного учебного заведения (рисунок 2.17).

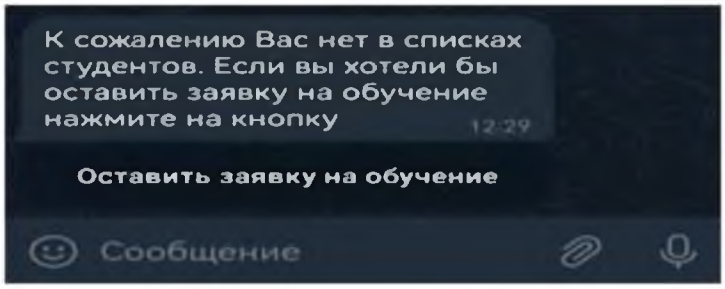


Рисунок 2.16 – Пример неудачной авторизации

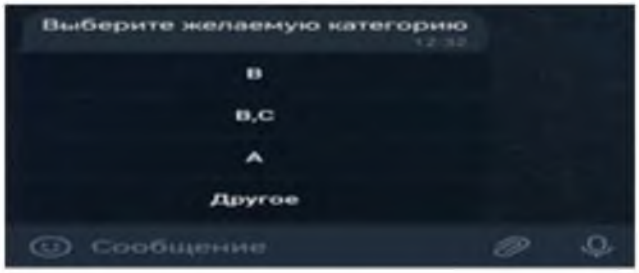


Рисунок 2.17 – Выбор категории

По завершении процедуры зачисления, с целью получения доступа к функциональу системы, требуется повторная авторизация. Процесс авторизации аналогичен ранее описанному алгоритму действий. Успешное прохождение авторизации подтверждается сообщением от бота, после которого курсанту открывается доступ к главному меню (рисунок 2.18).

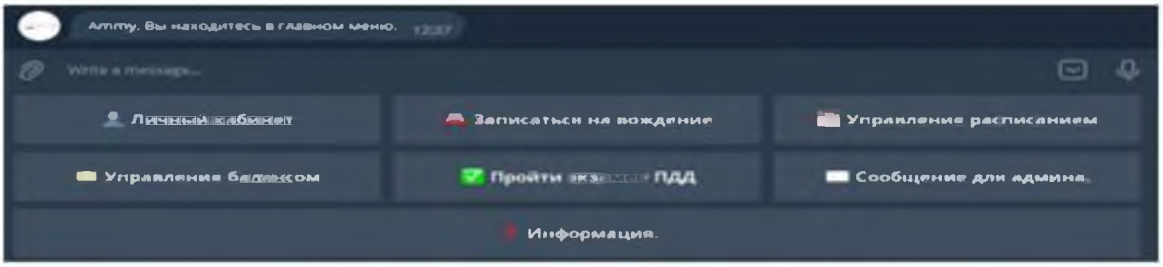


Рисунок 2.18 – Главное меню

Далее необходимо изучить алгоритм использования бота с целью эффективного обучения и заполнения личных данных, которые можно добавить

в категорию «Личный кабинет» (рисунок 2.19).

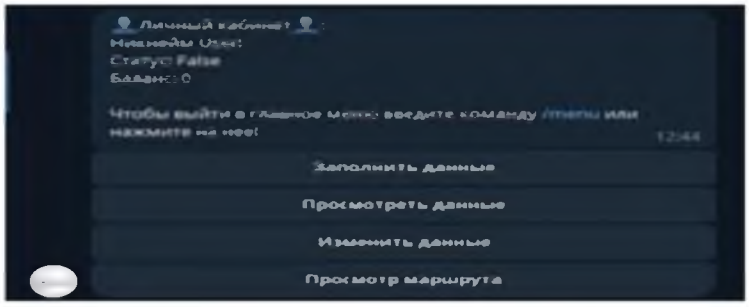


Рисунок 2.19 – Личный кабинет курсанта автошколы

При успешном исходе запонения всех обязательных полей и получения подтверждения корректности введенных данных, система осуществляет проверку достоверности информации. Далеепринажатии на кнопку «Просмотреть данные» происходит сопоставление введенных сведений с персональной информацией пользователя (рисунок 2.20)

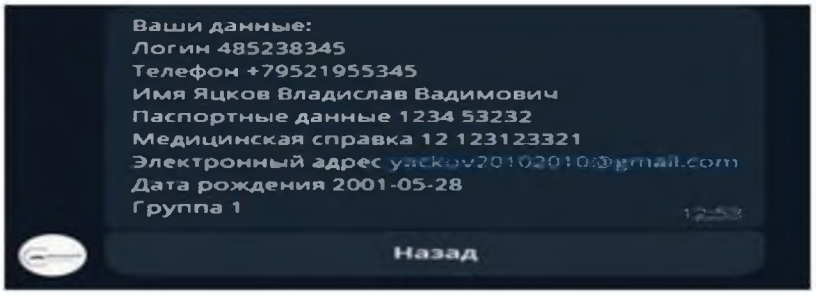


Рисунок 2.20 – Просмотр данных

Далее рассмотрим раздел «Управление балансом», который позволяет управлять балансом своего счета и контролировать нитолько поступление средств, но и необходимый остаток задолженности за обучение (рисунок 2.21).

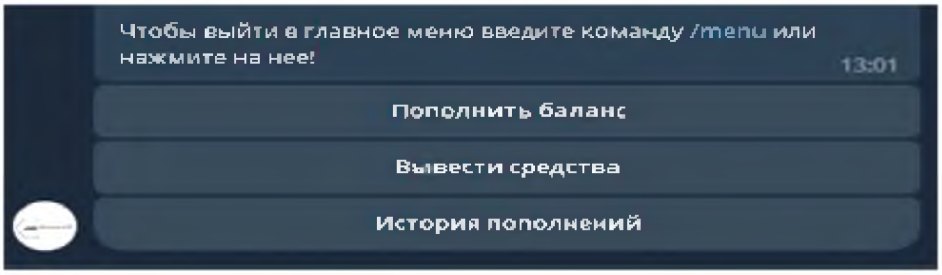


Рисунок 2.21 – Управление балансом

При пополнении баланса система отображает реквизиты автошколы для перевода денежных средств (рисунок 2.22). Обучающийся указывает желаемую

сумму платежа, делая выбор способа оплаты и нажимает кнопку «Оплатить». Затем требуется введение данных дебетовой карты. После заполнения данных на телефон пользователя приходит смс с кодом подтвержающего характера для завершения начатого платежа. Успешная транзакция сопровождается информационными сообщениями. В этом же разделе доступнафункцияч вывода денежных средств с помощью кнопки «Вывести средства». При этом бот непременно уточнит размер перевода и отобразит общий текущий баланс

Рисунок 2.22 – Пополнение баланса

Далее нужно вернуться в главное меню и перейти в раздел «Запись на занятия». В этом разделе есть возможность записи на 3 варианта мероприятий: теоретическое занятие, практический экзамен или теоретический экзамен, также бот дает информацию о записи. Это видно на рисунке 2.23.

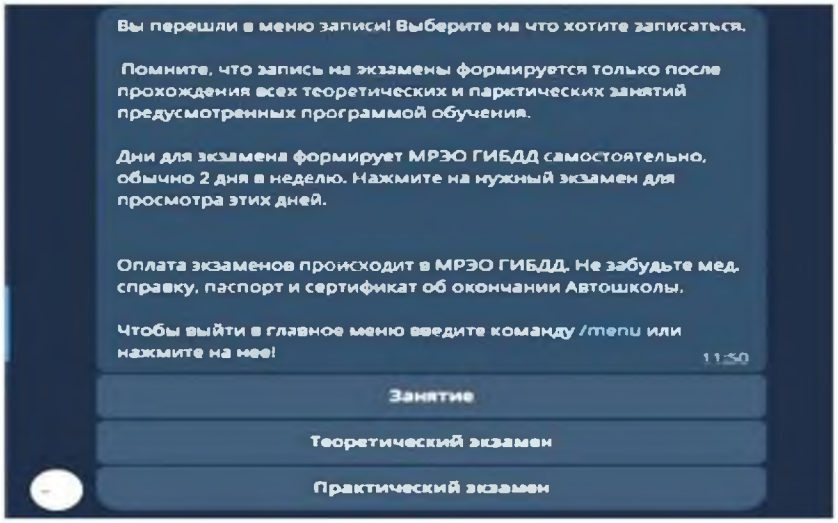


Рисунок 2.23 – Меню записи

Следующий раздел, который требует более пристального внимания, включает в себя составлению расписания учебных занятий. В данном разделе пользователь получает возможность изменений и ознакомления с текущими занятиями, само диалогое окно представлено на рисунке 2.24.

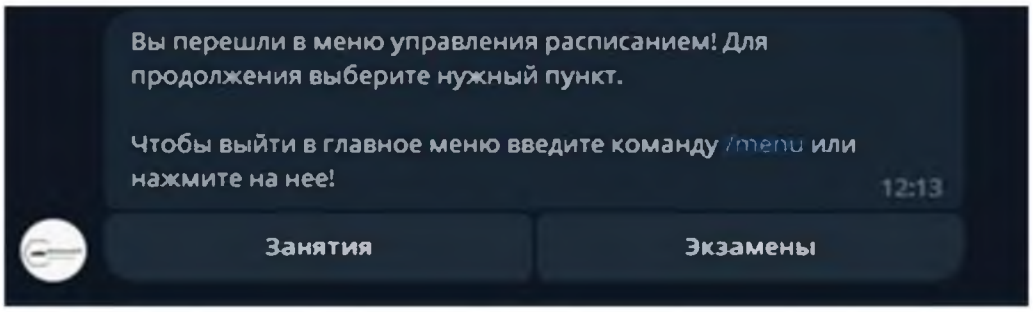


Рисунок 2.24 – Управление расписанием

В первую очередь, следует обратить внимание на функционал управления учебными занятиями. При активации соответствующей кнопки, курсант автошколы получает доступ к просмотру расписания своих текущих занятий. Пользователь может также ознакомиться с детальной информацией по каждому занятию конкретно, а при необходимости, аннулировать его, учитывая, что дата проведения занятия еще не наступила.

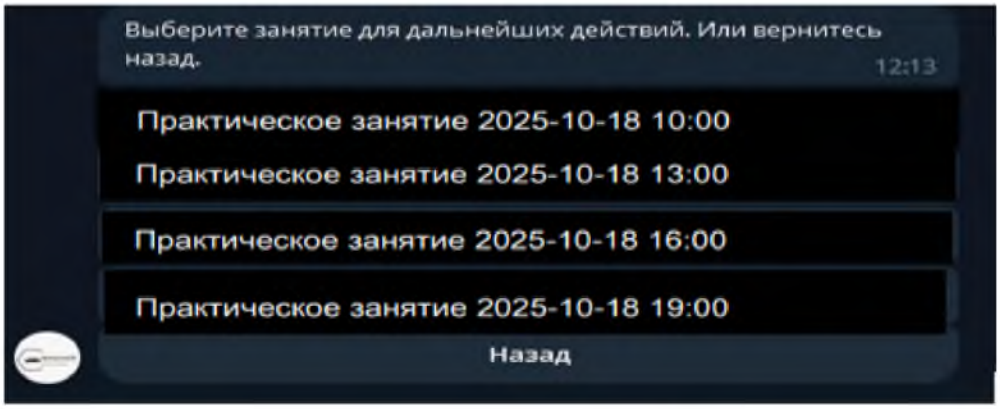


Рисунок 2.25 – Просмотр расписания

Иллюстрация рисунка 2.25 демонстрируетзанятия, упорядлченные по времени в хронологическом порядке, поэтому для того, чтобы получить дополнительную информацию или отменить занятие, курсанту слудет кликнуть на нужную кнопку. Далее рисунок 2.26 показывает пример самого занятия.

Важно подчеркнуть, что теоретическое занятие является групповым и пользователь не может никак повлиять на его изменения.

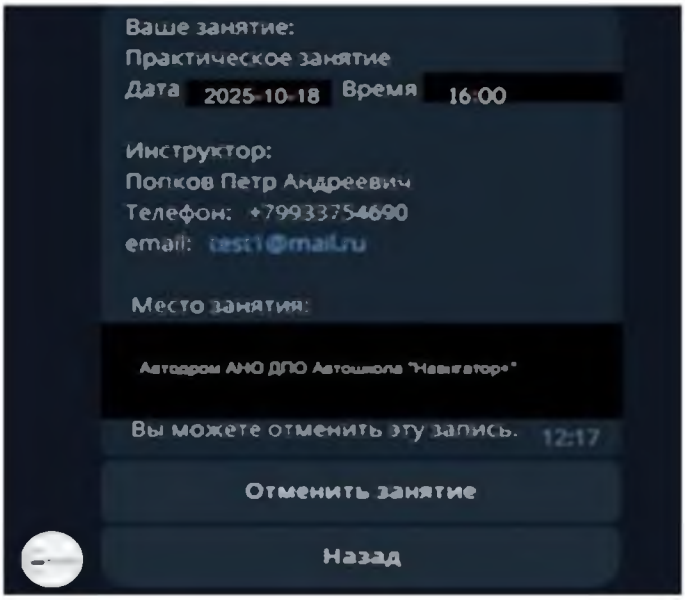


Рисунок 2.26 – Запись на практическое занятие

Далее рассмотрим, меню управления экзаменами, алгоритм работы которых работает аналогично занятиям и это видно на рисунке 2.27.

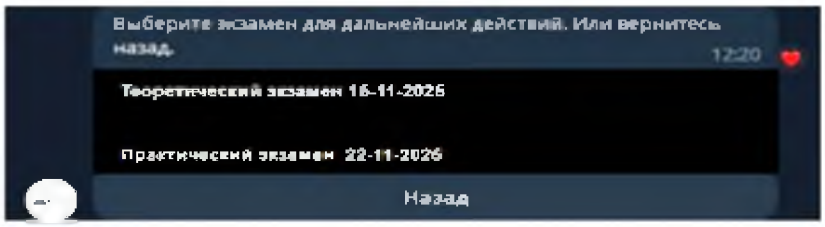


Рисунок 2.27 – Управление экзаменами

В дальнейшем необходимо выделить момент, что основными критериями оценки пройденного материала являются теоретические и практические экзамены. Причем в первую очередь они являются внутриклубными, а затем сторонними на базе ГИБДД. Нельзя не выделить, что экзамены подразделяются на несколько подразделов, каждый из которых определяетр уровень согласно компетенций. Поэтому следующий раздел, который будет изучен касается имено теоретического тестирования, которое как ни что другое отражает базовый уровень подготовки курсанта для итогового экзамена в ГИБДД.

В последствие разделделитсянанеобходимыеподразделыдляподготовки,

арежим «Экзамен», сделан на основе реального алгоритма создания экзаменационного билета. Меню показано на рисунке 2.28.

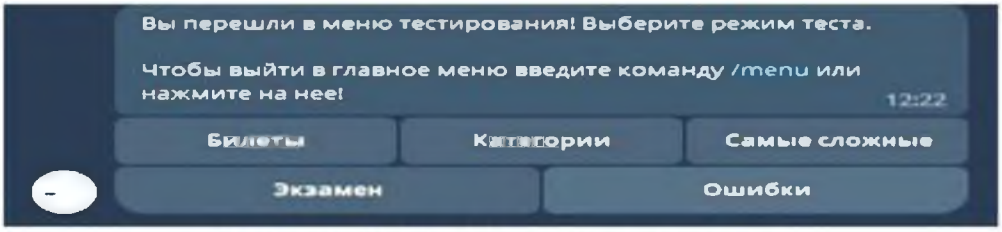


Рисунок 2.28 – Меню тестирования

Первый режим, это «Билеты», в данном режиме курсант сам может прорешать необходимый ему билет, билеты для подготовки формируются ГИБДД и имеют единый вид. Процедура выбора билета, и ответов на вопросы показана на рисунке 2.29.

Рисунок 2.29 – Выбор билета

Стоит заметить, что в данном режиме присутствуют подсказки-аннотации. Которые можно увидеть только если нажать на них, как показано на рисунках 2.30 - 2.31.

Рисунок 2.30 – Просмотр вопроса билета

Пользователь может в любой момент завершить тест или дорешать его до конца, после чего бот считает количество верных и не верных ответов и результат продемонстрирует пользователю, это видно на рисунке 2.32.

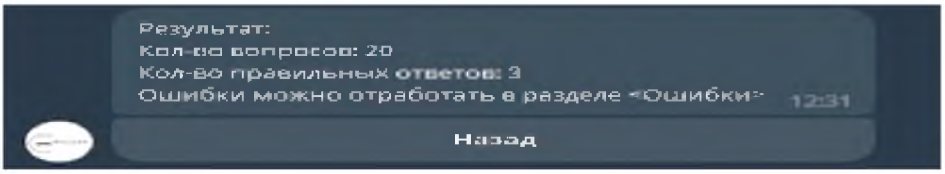


Рисунок 2.32 – Результат тестирования

Следующий раздел - это «Категории» в котором курсант может решать вопросы, разделяя их по категориям, меню сделано удобно, с возможностью листать список категорий и выбирать необходимую (рисунок 2.33).

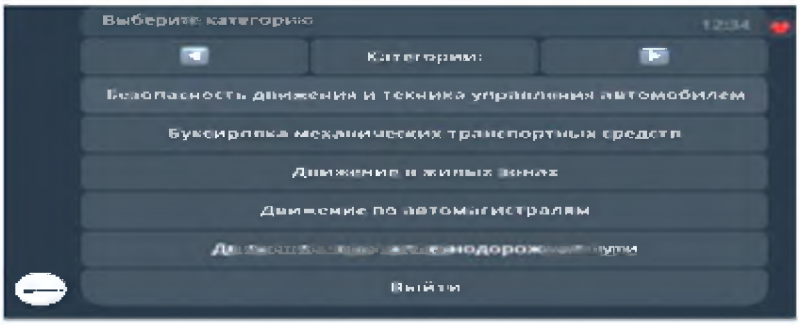


Рисунок 2.33 – Выбор категории

Завершая обзор информационной системы, акцентируем внимание на последнем небольшом разделе, доступном всем курсантам – «Информация об автошколе», который, как и предыдущие, открыт для всех пользователей без исключения и не зависит от статуса авторизации. В данном разделе легко можно найти информацию общего и контактного характера (рисунок 2.34).

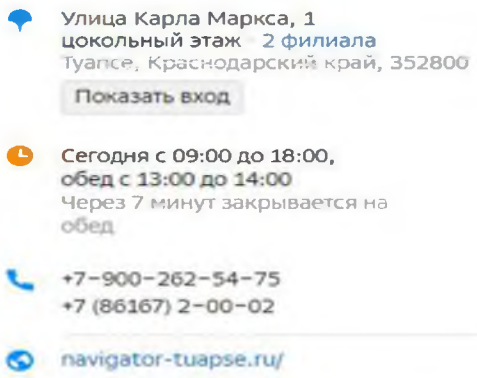


Рисунок 2.34 - Контактная информация об автошколе

Далее рассмотрим блок управления администратора автошколы, который может управлять всеми базами данных, а именно, добавлять, удалять и изменять поля определённой таблицы. Для доступа к этому меню, администратор пишет команду /admin. После этого бот запрашивает пароль, хэширует и сверяет его с хэшем, записанным в конфигурационном файле с паролем. Процесс доступа к данному меню показан на рисунках 2.35 – 2.36

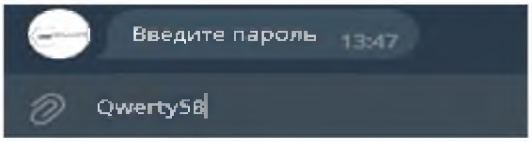


Рисунок 2.35 – Авторизация администратора

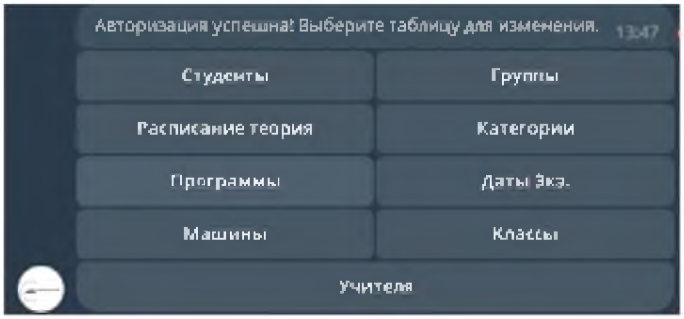


Рисунок 2.36 – Выбор таблицы

Теперь, для примера рассмотрим полное изменение таблицы «Студенты». Для этого нужно нажать на соответствующую кнопку «Студенты», а далее выбрать план действий, для начала добавить курсанта (рисунок 2.37).

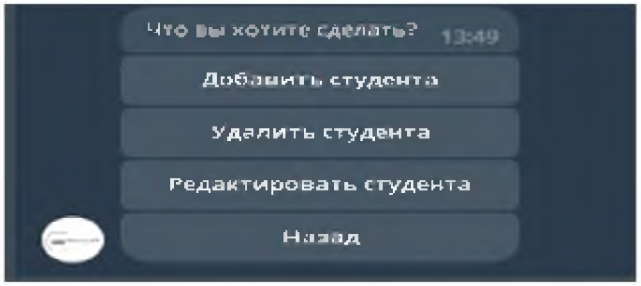


Рисунок 2.37 – Выбор действия

Ввод новых данных для бота может быть настроен по-разному в зависимости от платформы. Это может включать:

— использование форм ввода в конструкторе бота, которые записывают

данные пользователя в переменную;

— настройка валидации ответа — бот проверяет, соответствует ли ввод ожидаемому типу данных (дата, email, номер телефона и т. д.);

— интеграцию с внешними сервисами — бот может получать данные со стороннего сервиса и записывать их в переменные.

Далее бот информирует о том, как нужно передать данные для новой записи (рисунок 2.38).

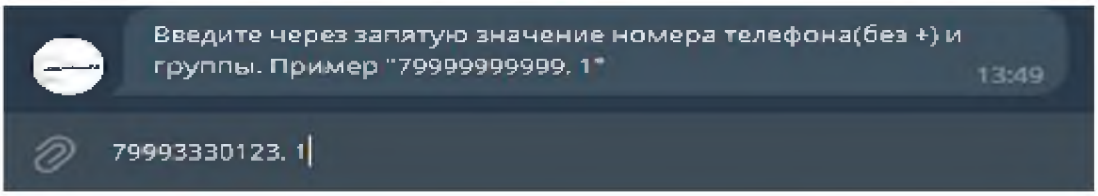


Рисунок 2.38 – Ввод новых данных

Редактирование обучающегося инициируется нажатием кнопки «Редактировать студента». Система запрашивает ID для поиска соответствующей записи (в конкретном случае это 5) и затем осуществляет выбор целевой колонки (например, name). После ввода нового значения, бот выполняет команду по обновлению полей (рисунок 2.39).

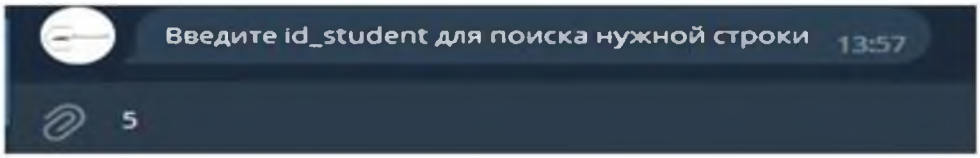


Рисунок 2.39 – Поиск нужного поля

Далее необходимо акцентировать внимание на выборе колонки, в которую будут внесены соответствующие корректировки относительно определенных параметров по индивидуальным критериям курсанта автошколы (рисунок 2.40).

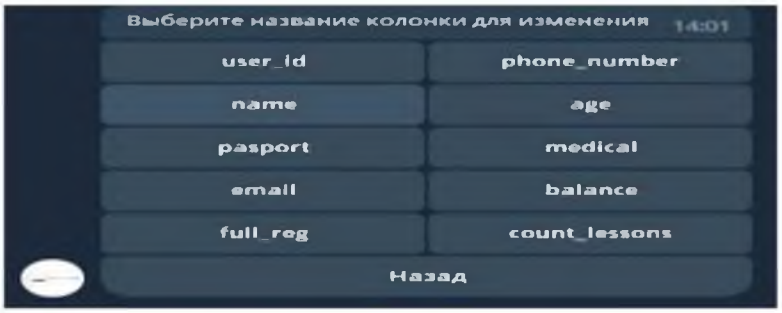


Рисунок 2.40 – Выбор колонки

Связывание поля ввода и подписи по id. Атрибут id также используется при создании форм. К полям ввода в форме часто нужно добавлять подписи для описания того, что нужно ввести пользователю (рисунок 2.41-2.42).

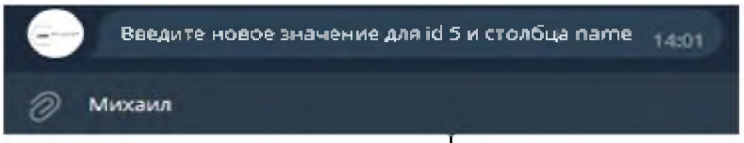


Рисунок 2.41 – Ввод значения

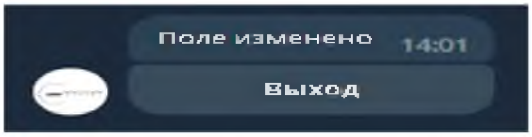


Рисунок 2.42 – Уведомление об изменении

Теперь проверим нашу запись непосредственно в СУБД, для этого найдем студента с *id* 5. Результат показа на рисунке 2.43.

	id_student [PK] integer	user_id bigint	phone_number text	name text	age date	pasport text	medical text
1	1	4B523B345	79521955345	Яцков Владислав Вадимович	2001-05-28	1234 53232	12 123123321
2	5		79993330123	Михаил			

Рисунок 2.43 – Просмотр БД

Далее нужно удалить этого студента, для этого нажимаем на кнопку «Удалить студента» и вводим *id*студента, которого мы хотим удалить. В нашем случае это 5. Процесс удаления показан на рисунке 2.44.

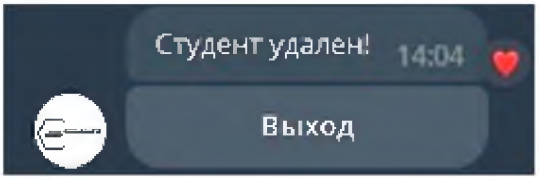


Рисунок 2.44 – Успешное удаление

В контексте систем управления базами данных (СУБД) администратор АНО ДПО Автошкола «Навигатор+» имеет возможность предоставления прав

другим пользователям, открывая доступ к ранее сформированным таблицам путем делегирования соответствующих полномочий, что в итоге позволяет:

— разрешить доступ к таблице - пользователь, который её создал может предоставить необходимые права (SELECT, INSERT, UPDATE и др.) другим пользователям, для этого используется оператор;

— ограничить доступ - привилегии INSERT и UPDATE могут ограничиваться лишь отдельными столбцами таблицы, в этом случае пользователю разрешается модифицировать значения только указанных столбцов.

Аналогичным образом привилегия REFERENCES может распространяться исключительно на отдельные столбцы таблицы, что позволяет использовать их имена в формулировках требований защиты целостности данных (например, в предложениях CHECK и FOREIGN KEY, входящих в определение других таблиц);

— передавать права - благодаря параметру в операторе GRANT пользователи имеют право передавать все предоставленные им в отношении указанного объекта привилегии другим пользователям, которые, в свою очередь, будут наделены точно таким же правом передачи своих полномочий. Если этот параметр не будет указан, получатель привилегии не сможет передать свои права другим пользователям.

2.5 Пользовательский интерфейс информационной системы

Пользовательский интерфейс информационной системы представляет собой набор инструментов, как программных, так и аппаратных, которые предоставляют пользователю эффективное взаимодействие с самой системой. Именно интерфейс включает в себя все элементы, через которые проивходит сам непосредственный процесс управления и получение необходимого формата информации, начиная от навигационных меню и форм ввода данных и заканчивая уведомлениями об ошибках и встроенной помощи.

Проектирование информационной системы именно для автошколы - это особый процесс создания автоматизированной системы, которая будет реализовать функции, непосредственно связанные с управлением процессов в внутри учебного заведения. Для разработки информационной системы могут предположительно быть использованы, например:

- веб-приложения на архитектуре «клиент-сервер» для привлечения потенциальных клиентов в процессе обучения;

- мобильные приложения для контроля и планирования занятийц, например, для записи на экзамены и процесс общения с преподавателем.

Проектирование информационной системы включает несколько этапов разработки:

- анализ, проектирование и реализация. На этапах анализа и проектирования происходит непосредственное построение самой архитектуры. Базовая архитектура системы, включающая перенос данных на сервер, называется «клиент-сервер».

В данной модели клиентские устройства, а именно компьютеры или смартфоны, несомненно, контактируют с самим сервером, который предоставляет необходимые ресрусы и услуги. В свою очередь сервер обрабатывает запросы клиентов и возвращает соответсвтующие ответы.

Для реализации архитектуры «клиент-сервер» используется сервер баз данных, иными словами, приложение или сервер операционной системы, который управляет данными и выполняет запросы клиентских приложенйи. Также часть бизнес логики может быть реализована ввиде хранимых процедур сервера, что в свою очередь, позволяет значительно сократить объем операций обмена между клиентом и сервером.После реализации в полном объеме всего функционала информационной системы, происходит тельный процесс переноса его с локальной среды на удаленный сервер.

Сам процесс перемещения ИС на сервер обусловлен необходимостью повышеняия доступности, масштабируемости, упрочения основ безопасности и оптимизации самого обслуживания. После выбора подходящего хостинга и

серверных параметров происходит детальная и скурпулезная настройка самого сервера.

Для начала подключаемся и обновляем пакеты на сервере командой sudoapt - y upgrade, также устанавливаем нужные языки и выбираем кодировку эти делается с помощью команд dpkg- reconfigurelocales и LANG=en_US.U

TF-8. После этого нужно перейти к настройке pip3 и средств разработки. Следующим этапом будет установка виртуального окружения venv и установка PostgreSQL. Установка виртуального окружения (venv) в Python включает создание изолированной среды для выполнения Python-программ с помощью встроенного модуля venv. Каждое окруженне содержит собственную директорию с установленными библиотеками и не влияет на системный Python и другие окружения (рисунок 2.45 - 2.46).

Рисунок 2.45 – Установка venv

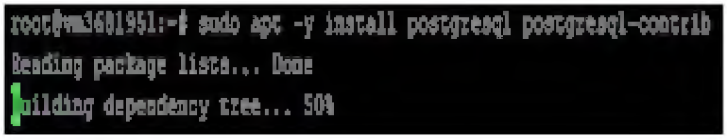


Рисунок 2.46 - Установка PostgreSQL

Для начала подключаемся и обновляем пакеты на сервере командой sudoapt -y upgrade, также устанавливаем нужные языки и выбираем кодировку эти делается с помощью команд dpkg- reconfigurelocales и LANG=en_US.U

TF-8. После этого нужно перейти кнастройке pip3 и средств разработки.Следующим этапом будет установка виртуального окружения venvи установка PostgreSQL.

Установка виртуального окружения (venv) в Python включает создание изолированной среды для выполнения Python-программ с помощью встроенного модуля venv.Каждое окружение содержит собственную

директорию с установленными библиотеками и не влияет на системный Python и другие окружения (рисунок 2.45 -2.46).

PostgreSQL как система управления базами данных (СУБД) предназначена в первую очередь для работы непосредственно с открытым кодом, которая в свою очередь опирается в своем функционале на стандартный язык SQL.

Алгоритм последовательности действий следующие: перенос всех файлов программы на сервер, создание резервной копии БД и установка ее на сервер. Список перенесенных файлов представлен в иллюстрированном варианте на рисунке 2.47.

После этого, нужно создать новую БД на сервере с помощью команды createdbautoschool и поместить в нее резервную копию БД, командой pg_dump -Upostgres - dautoschool >backup.sql. Когда БД будет перенесена, нужно установить все пакеты, которые необходимы для успешной компиляции программы, все используемые библиотеки хранятся в файле requirements.txt.

Рисунок 2.47 – Файлы программы на сервере

Для установки нужно войти в виртуальное окружение и воспользоваться командой pip3 install - rrequirements.txt.

Когда все готово к запуску программы, нужно создать скрин, с помощью которого бот будет работать автономно. Это делается с помощью команды screen-S bot. Далее нужно запустить бот и убедиться в правильности его работы, пример показан на рисунке 2.49.

Рисунок 2.49 – Запуск бота на сервере

Файлы бота можно загрузить на сервер через сторонний FTP-клиент или панель управления VPS/VDS, если она имеется. Понадобятся данные для входа на сервер: логин, пароль и IP-адрес.

3Обоснование экономической эффективности результатов ВКР

3.1Реализация информационной системы для автошколы

Разработка и реализация информационной системы для конкретной автошколы – это первый шаг к полной автоматизации управления всего процесса обучения, это процесс упорядочивания всей документации и как следствие, росту эффективности. Это ни просто определенный помощник в виде телеграм-бота – это прочный фундамент новой информационной системы дополнительного образования, подготовленной к коренным изменениям, росту и развитию.

Процесс внедрения информационной системы сосредоточен тна трех основных этапах: формировании физической модели данных, кодировании системы на предпочтительном языке программирования и заполнение базы данных непосредственной информацией. Сама физическая модель, являющаяся продуктом этапа физического проектирования данных, непосредственно располагается на предварительно подготовленной логической модели. Она аккумулирует исчерпывающие сведения, достаточные для автоматизированного создания всех требуемых объектов в базе данных[13,с.81].

Выделим основные задачи физического проектирования базы данных:

— выбор эффективного размещения базы данных на внешних носителях для обеспечения наиболее эффективной работы приложения;

— выбор вычислительных ресурсов, необходимых для функционирования системы (типа и конфигурации ЭВМ, операционной системы).

Процесс разработки и в последствие реализации информационной системы совместно с ее диаграммой развертывания непосредственно связаны между собой. Сама диаграмма развертывания, по сути, является визуальным представлением того, на сколько система эффективно работает в реальной среде. Именно за счет нее можно получить данные о задействованных

аппаратных компонентам(процессоров и устройств) совместно с программными процессами, которые между собой систематически взаимодействуют. Таким образом, диаграмма развертывания моделирует физическую структуру системы, демонстрируя ее архитектуру.Диаграмма развёртывания (Deploymentdiagram) является элементом проектирования информационной системы в унифицированном языке моделирования (UML). Это структурная схема, которая показывает архитектуру системы, включая узлы (аппаратные или программные среды выполнения) и связующее их промежуточное программное обеспечение.Диаграмма развертывания, проиллюстрированная на рисунке 3.1 является ключевым инструментом для понимания физической структуры системы. Она наглядно демонстрирует, как компоненты приложения распределены по различным аппаратным устройствам и серверам, и как данные компоненты контактируют между собой на физическом уровне. Данный процесс позволяет получить четкое представление об архитектуре самойц системы и ее инфраструктуре.

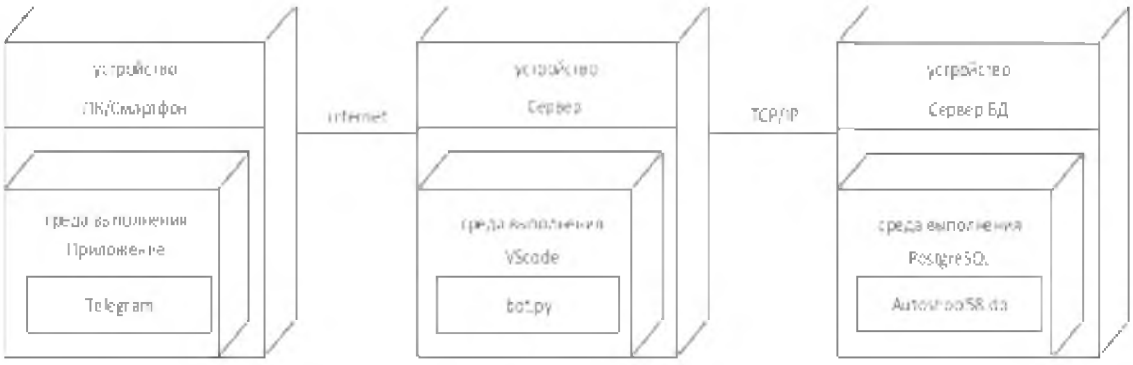


Рисунок 3.1 - UML-диаграмма развертывания

Файл.gitignore - это файл конфигурации, используемый в системе контроля версий Git для указания файлов и папок, которые должны быть проигнорированы и не отслеживаемы Git при выполнении операций коммита и других действий.

Файл admin.txt - файл, в котором хранится логин и хэш пароля админа, используемый для аутентификации и авторизации администратора в системе.

Файл create_bot.py - файл, в котором хранятся конфигурационные константы

для успешного запуска и использования бота.

```
from db import Database host_db="localhost", port_db = "5432", database="Autoshool", user="postgres", password_db="Qwerty"
payment_token = '401643678:TEST:c7d71cee-0944-40ea-9955- 3a04a75bc28a'
bot= Bot(token='6132924794:AAEGdkuvbN_lOCnlThLVFHCPwvN afdgmic')
dp = Dispatcher(bot,storage=MemoryStorage()) mail = 'autoschool@mail.ru'
email_token='cdjMzQQ96qt6fHnMbxss' host='smtp.mail.ru'
port=465
```

Файл start_bot.py содержит в себе main функцию, а также инициализирует все переданные в бот хэндлеры

```
from handlers import menu, lk,rec_lesson, exam_pdd, other, payment, edit_lesson, command_menu
from create_bot import dp
menu.register_handlers(dp)
lk.register_handlers(dp)
rec_lesson.register_handlers(dp)
payment.register_handlers(dp)
edit_lesson.register_handlers(dp)
exam_pdd.register_handlers(dp)
command_menu.register_handlers(dp)
other.register_handlers(dp)
if _____name__ == '___main__':
executor.start_polling(dp, skip_updates=True)
```

Визуальный интерфейс телеграм-бота в Python можно создавать с использованием TelegramBotAPI и различных библиотек, таких как python-telegram-bot.Нижеперечисленынекоторыевидывизуальныхэлементов,которые можно использовать в интерфейсе телеграм-бота [5,с.108]:

— система способна инициировать отправлние текстовых сообщений абоненту, которые могут представлять собой как лаконичные ответы, так и детализированные типовые инструкции, направляемые пользователю;

— предусмотрена возможность для предоставления пользователю

интерактивных опций или запуска определенных функци применения встроенного вида кнопок, которые могут быть интегрированы непосредственно в тело текстового сообщения;

— Reply кнопки показываются внизу экрана в виде списка;

— инлайн-клавиатура — это набор кнопок, которые отображаются внизу экрана и позволяют пользователю выбирать опции или выполнять действия в контексте текущего сообщения, таким образом сделан календарь, а также выбор категории для тестирования;

— всплывающие сообщения обычно реализуются с использованием Inline-режима или Inline-уведомлений, они могут содержать дополнительную информацию или предложения для пользователя и отображаться поверх текущего чата или сообщения;

— вложения (фото, видео, аудио, файлы) - бот может отправлять пользователю вложения, такие как фотографии, видео, аудио или файлы;

— скрытый текст в сообщении.Скрытый текст в сообщении телеграм-бота представляет собой текстовую информацию, которая скрыта от обычного отображения и требует специальных действий или команд пользователя для его просмотра. Это может быть полезно, когда вы хотите предоставить пользователю дополнительную информацию или контекст, но не хотите перегружать основное сообщение.

При поступлении ввода пользователем либо команды, либо текста происходит фильтрация и последующая обработка сведений в специализированном модуле. Перечегь данных модулей, организованным по разделам интерфейса, представлен ниже и содержит соответствующие функциональные возможности:

— Модуль menu.py – главное меню, в которое попадает пользователь после команды /start, а также проходит авторизацию или подает заявку на обучение. Функция cmd_menu отображает главное меню в телеграм-боте, предоставляя пользователю доступ к различным функциям, в зависимости от наличия авторизации. Функция cmd_start используется для обработки команды

старта. Ее цель - авторизовать пользователя и предоставить доступ к главному меню, если пользователь не был ранее зарегистрирован;

— Функция get_auto используется в дипломной работе для обработки получения авторизационных данных от пользователя. Она проверяет наличие номера телефона в базе данных и предоставляет доступ к регистрации в личном кабинете или возможность оставить заявку на обучение, в зависимости от результата проверки. Функция info_menu используется для отображения информации о автошколе и ее контактных данных. Она также предоставляет кнопку для просмотра местоположения на карте. Ниже приведен пример вызова этих функций с помощью хэндлеров.

```
Defregister_handlers(dp:Dispatcher):  dp.register_message_handler(cmd_menu, commands=['menu']) dp.register_message_handler(cmd_menu,lambdamsg:msg.text
    == "Отклонить")
    dp.register_message_handler(info_menu, lambda msg: msg.text
    == " ? Информация")
    dp.register_message_handler(cmd_start, commands=['start'] )
    dp.register_callback_query_handler(sign, lambda c: c.data.startswith('sign_'))
    dp.register_callback_query_handler(bd_add, lambda c: c.data.startswith('s_'))
    dp.register_callback_query_handler(delete, lambda c: c.data.startswith('del'))
    dp.register_message_handler(get_auto,
content_types=types.ContentType.CONTACT )
```

Модуль lk.py– личный кабинет, подменю в котором пользователь может взаимодействовать с личными данными.

Функция with_puree используется для обработки запроса в личный кабинет пользователя. Она отображает информацию о пользователе, такую как никнейм, статус и баланс, а также предоставляет кнопки для заполнения, просмотра и изменения данных, а также просмотра маршрута.

Функция auto выполняет действия в личном кабинете пользователя, такие как заполнение данных, изменение уже заполненных данных, просмотр данных и просмотр маршрута, в зависимости от значения callback.data.

Функция set_data обрабатывает нажатия на кнопки для заполнения персональных данных, в зависимости от выбранной кнопки задает соответствующий вопрос и устанавливает состояние пользовательского состояния.

Функция set_personal_data сохраняет введенные пользователем данные в базе данных и выводит подтверждение, предлагая пользователю верифицировать данные.

Аналогичным образом работают функции edit_data и edit_personal_data, но предназначены для изменения уже заполненных данных. Ниже приведен пример вызова этих функций с помощью хэндлеров.

```
defregister_handlers(dp: Dispatcher):
    dp.register_message_handler(lk, lambda message:"Личный кабинет" in message.text)
    dp.register_callback_query_handler(with_puree,lambda callback: callback.data in ["back_to_lk"])
    dp.register_callback_query_handler(auto, lambda callback: callback.data in ["log in","view_data","edit_data","view_marshrut","history_balance"] )
    dp.register_callback_query_handler(edit_data, lambda callback: callback.data in ["edit_name","edit_pasport","edit_medical","edit_email","edit_age"])
    dp.register_callback_query_handler(set_data, lambda callback: callback.data in ["set_name","set_pasport","set_medical","set_email","set_age"])
    dp.registermessage_handler(set_personal_data,state=[UserState.name,UserState passport,UserState.medical,UserState.email,UserState.age],content_types=types.
    ContentTypes.TEXT)
    dp.register_message_handler(edit_personal_data,state=[UserState.name_edit,
    UserState.passport_edit,
    UserState.medical_edit,UserState.email_edit,UserState.age_edit],content_types=
types.ContentTypes.TEXT)
```

Модуль rec_lesson.py – под меню, в котором пользователь может записаться на занятия и экзамены. В функции rec_menu обрабатывается

сообщение пользователя в меню записи на вождение. Затем формируется клавиатура с кнопками «Занятие», «Теоретический экзамен» и «Практический экзамен». После этого отправляется сообщение с информацией о записи, инструкциями и доступными опциями.

Система предоставляет пользователю инструкцию по возврату в главное меню. В функции rec обрабатывается выбор пользователя в меню записи на вождение.

В зависимости от выбранной опции («Занятие», «Теоретический экзамен» или «Практический экзамен»), устанавливаются соответствующие флаги lesson, t_exam и p_exam. Если выбрана опция «Занятие», проверяется количество уже записанных занятий. Если пользователь еще может записаться на занятие, вызывается функция show_calendar для отображения календаря и возможности выбора даты занятия. В противном случае пользователю выводится сообщение о том, что он не может записаться на больше пяти занятий одновременно, и вызывается функция rec_menu для возврата к меню записи. Если выбрана опция «Теоретический экзамен» или «Практический экзамен», вызывается функция exams, которая отображает доступные даты для экзамена. В зависимости от выбранного типа экзамена (t_exam или p_exam), формируется соответствующая клавиатура с датами экзаменов. После выбора даты пользователем вызывается функция back_to_rec для возврата к меню записи. Если выбрана опция «Назад», вызывается функция rec_menu для возврата к меню записи.

Функция show_calendar отображает календарь пользователю, чтобы он мог выбрать дату. Внутри функции вызывается create_calendar, которая создает клавиатуру с датами. Клавиатура состоит из трех строк: первая строка содержит стрелки для перехода к предыдущему и следующему месяцам, вторая строка содержит названия дней недели, а остальные строки содержат дни месяца. Каждая кнопка с днем содержит соответствующий callback_data, в котором указывается тип записи (lesson, t_exam или p_exam) и выбранная дата

Функции calendar_prev_month_callback и calendar_next_month_callbackобрабатывают нажатия на кнопки «вперед» и

«назад» для переключения между предыдущим и следующим месяцами в календаре. В callback_data передаются текущий год, месяц и тип записи (lesson, t_exam или p_exam), а затем вызывается функция create_calendar для создания новой клавиатуры с выбранным месяцем.

Функция show_time_picker отображает пользователю выбор времени на определенный день. Она получает массив доступных временных слотов для выбранного дня из базы данных и создает клавиатуру с кнопками для каждого доступного времени. Клавиатура также содержит кнопку "Назад" для возврата к меню записи.

Функция process_lesson_callback обрабатывает нажатия на кнопки выбора времени вкалендаре.Взависимостиотвыбранноговремени,функцияпроверяет, доступно ли выбранное время для записи. Если время уже занято, прошло или является выходным днем, выводится соответствующее сообщение ипредлагается вернуться назад. Если время доступно для записи, выводится сообщение с подтверждением выбранной даты и времени, а также кнопкой «Записаться» и кнопкой «Назад».

Функцияrecover обрабатывает обратные вызовы при записи пользователя назанятие,теоретическийэкзаменилипрактическийэкзамен,добавляетзаписьв базу данных, а также формирует чек оплаты и высылает его на привязанный электронный адрес.

Ниже приведен пример вызова этих функций с помощью хэндлеров.

```
defregister_handlers(dp: Dispatcher): dp.register_message_handler (rec_menu,
lambda message:
message.text == "Записаться навождение")
dp.register_callback_query_handler (rec, lambda callback:
callback.data in ["lesson","t_exam","p_exam","back_to_rec"])
dp.register_callback_query_handler(recover, lambda query:
query.data.startswith('rec'))
dp.register_callback_query_handler(calendar_next_month_callback,
lambda query: query.data.startswith('calendar-next-month'))
```

dp.register_callback_query_handler (calendar_prev_month_callback, lambda query: query.data.startswith ('calendar-prev-month'))

dp.register_callback_query_handler (process_calendar_day, lambda c: c.data.startswith('calendar-day'))

dp.register_callback_query_handler(process_lesson_callback, lambda c: c.data.startswith('lesson-time-'))

Модуль edit_lesson.py – подменю, в котором пользователь может просматривать и редактировать свое расписание.

Функция les_menu используется для отображения меню управления расписанием. Затем функция отвечает пользователю, предлагая выбрать нужный пункт меню.

Функцияview_worksиспользуетсядляотображениясписказадоступных занятий или экзаменов для пользователя. В зависимости от выбранного типа (занятия или экзамены), функция извлекает соответствующую информацию из базы данных и формирует соответствующую клавиатуру с кнопками для каждого элемента списка. Затем функция отправляет сообщение с выбором пользователю.

Функция edit_works используется для отображения информации о выбранном занятии или экзамене. В зависимости от типа (занятие или экзамен) и даты, функция формирует сообщение с информацией о записи и правилах отмены. Если запись может быть отменена (если дата еще не наступила), функция добавляет кнопку «Отменить запись». Затемфункцияотправляетсообщениесинформациейпользователю.

Функция return_works применяется в системе для обработки запросов на отмену учебных занятий или экзаментов. В зависимости от типа отменяемого события, функция инициирует вызов соответствующих методов базы данных для непосредственной корректировки и удаления записи полностью. При провизведении отмены самого занятия дополнительно выполняется обновление баланса пользователя, формирование платежного документа (чека) на возврат денежных средств, что направляется непосредственно пользователю. При

завершении всех операций функция возвращает подтверждение об отмене и перенаправляет пользователя в меню управления расписанием. Примеры вызова данной функции при помощи хэндлеров представлены ниже.

```
defregister_handlers (dp: Dispatcher): dp.register_message_handler (les_menu,
lambda message:
    message.text == "Управлениерасписанием")
    dp.register_callback_query_handler (view_works, lambda query:
    query.data.startswith ('view_')) p.register_callback_query_handler(edit_works,
    lambda query: query.data.startswith ('work_'))
    dp.register_callback_query_handler(return_works, lambda
    query: query.data.startswith ('return_'))
```

Модульexam_pdd.py – подменю, в котором пользователь может проходить теоретическое тестирование и готовиться к экзамену.

Функция menu_exam используется для отображения меню тестирования. При вызове функции происходит удаление предыдущего сообщения, если пользователь выбрал опцию «Пройти экзамен ПДД». Затем функция создает клавиатуру с пунктами меню, такими как «Билеты», «Категории», «Самые сложные», «Экзамен» и «Ошибки». Каждая кнопка имеет соответствующий callback_data. После создания клавиатуры, функция отправляет сообщение с меню тестирования и клавиатурой пользователю.

Функция mode_quiz обрабатывает выбор пользователя в меню тестирования. В зависимости от выбранной опции, функция выполняет определенные действия.

При активации режима «Экзамен» система формирует набор экзаменационных вопросв, регистрирует его в базе данных и инициирует отправку первого вопроса пользователю посредством вызова функции send_question. В случае выбора режима «Билеты» происходит генерация интерактивной клавиатуры с определением выбора опций билетов, которая в последствии предоставляется непосредственно пользователю При выборе режима «Сложные вопросы» осуществляется извлечение соответствующих вопросов базы данных, их

последующее хранение в качестве списка вопросов и отправки первого из них самому пользователю.При выборе опции «Категории» вызов идет на функцию show_category, которая непосредственно отражает перечень доступных категорий вопросов.В случае выбора опции «Ошибки», данная функция выполняет проверку наличия неточностей в предыдущем тесте пользователя. При полном отсутствии ошибок отправляется соответствующее уведомление, когда обнаружены типовые ошибки, функция сохраняет список вопросов с неточностями, допущенными при ответах в базе данных и вызывает функцию send_question для направления вопросов повторно, начиная с первого.

Функция bilet_mode отвечает за обработку команды пользователя для режима «Билет». Вначале она удаляет сообщение, в котором была нажата кнопка. Затем из базы данных получает все вопросы для выбранного билета. Полученные вопросы сортируются по порядку. Далее создается идентификатор теста, вызывается функция set_questions_quiz для сохранения вопросов теста и устанавливается начальный индекс вопроса. Затем отправляется первый вопрос с помощью функции send_question.

Функция category_mode отвечает за обработку команды пользователя для режима «Категория». Она также удаляет сообщение, в котором была нажата кнопка. Затем получает индекс выбранной категории и из базы данных получает все вопросы для этой категории. Полученные вопросы сортируются по порядку. Создается идентификатор теста, вызывается функция set_questions_quiz для сохранения вопросов теста и устанавливается начальный индекс вопроса. Затем отправляется первый вопрос с помощью функции send_question.

Функция send_question предназначена для интерактивного опроса пользователей. С ее помощью происходит извлечение данных относительно текущих вопросов из перечня доступных ответов. Так же именно данная функция формирует информационное сообщение, включая текст самого вопроса и варианты правильных ответов. При наличии иллюстрированных

вопросов, происходит автоматическое дополнение к сообщению. Правильный ответ на вопрос сохраняется в памяти. Созданные клавиатура и кнопка «Закончить тест» направляются пользователю.

Функция check_answer отвечает за проверку ответа пользователя. Она принимает выбранный вариант ответа, правильный ответ, номер вопроса, идентификатор теста, аналог варианта ответа и тип аннотации, иными словами способ отображения подсказки.

В случае полного совпадения выбранного ответа с правильным, количество верный вариантов увеличивается.

В последствие из базы данных извлекаются все вопросы тестоовых заданий.

Если в тесте имеются еще не отраженные при запросе вопросы, функциясheck_answer отправляет следующий вопрос на рассмотрение пользователю.

По итогу пользователь получает результат тестирования.

Функция show_category используется для отображения категорий пользователю. Она создает клавиатуру с категориями и отправляет ее пользователю.

Клавиатура содержит кнопки для выбора категорий, кнопку «назад» для перехода к предыдущим категориям, кнопку «вперед» для перехода к следующим категориям.

Ниже приведен пример вызова этих функций спомощью хэндлеров.

Модульpayment.py – раздел, в котором курсант автошколы может пополнять баланс, просматривать его историю, а также выводить деньги на карту.

Функция payment_menu отвечает за отображение меню управления балансом пользователя.

Затем она создает клавиатуру с кнопками для пополнения баланса, вывода средств и просмотра истории пополнений.

Функция отправляет сообщение с этой клавиатурой пользователю.

defregister_handlers(dp:Dispatcher):dp.register_message_handler(menu_exam, lambda message:

message.text="✅Пройти экзамен ПДД") dp.register_callback_query_handler (mode_quiz, lambda query:

query.data.startswith('mode_')) dp.register_callback_query_handler(bilet_mode, lambda query: query. data. Startswith ('bilet_'))

dp.register_callback_query_handler(check_answer, lambda query: query.data.startswith('answer_'))

dp.register_callback_query_handler(cat_next_callback, lambda query: query.data.startswith('cat-next-'))

dp.register_callback_query_handler(cat_prev_callback, lambda query: query.data.startswith('cat-prev-'))

dp.register_callback_query_handler(category_mode, lambda query: query.data.startswith ('category_'))

Функция confirm_return_money отвечает за подтверждение запроса на возврат средств. Она получает баланс пользователя, сохраняет информацию о возврате денежных средств в базе данных и обновляет баланс пользователя. Затем функция отправляет ответ пользователю с помощью метода answer и вызывает функцию payment_menu для отображения меню управления балансом.

Функция operation_money_callback обрабатывает нажатие кнопок в меню управления балансом. Если пользователь выбирает опцию пополнения баланса, функция проверяет наличие корректного email адреса у пользователя. Если email адрес указан некорректно или не указан вовсе, функция отправляет сообщение с кнопкой «Назад». В противном случае, функция создает клавиатуру с кнопками для выбора суммы пополнения и отправляет сообщение пользователю. Если пользователь выбирает опцию запроса на возврат средств, функция проверяет баланс пользователя. Если баланс равен 0, функция отправляет сообщение с кнопкой «Назад».

В противном случае, система инициирует создание пользовательского интерфейса, включающего кнопку подтверждения дейсвтий возврата и отправляет соответствующее уведомление пользователю. При выборе

пользователем опции просмотра истории пополнений, происходит извлечение данных о транзакциях пользователя из хранилища данных. Дадее формируется текстовое представление информационного плата о всех транзакциях, которое в в последствии поступает пользователю.В случае отсутствия транзакций у пользователя, система так же уведомляет его об этом.

Функция add_balance обрабатывает выбор суммы пополнения баланса. Если пользователь выбирает кнопку «Назад», функция вызывает функцию payment_menu для отображения меню управления балансом. В противном случае, функция удаляет сообщение с кнопками пополнения, создает объект с информацией о платеже и отправляет запрос на оплату пользователю с помощью метода send_invoice.

Функция payment формирует объект с информацией о платеже на основе данных из выбранной кнопки пополнения. Она возвращает chat_id и словарь с информацией о платеже.

Функция checkout_process обрабатывает результат предварительной проверки платежа. Если платеж не прошел успешно, функция отправляет ответ с ошибкой. В противном случае, функция отправляет успешный ответ.

Функция process_successful_payment обрабатывает успешный платеж. Она получает информацию о платеже из сообщения и сохраняет его данные в базе данных с помощью функции set_payment. Затем функция обновляет баланс пользователя и генерирует, и отправляет чек на email. Ниже приведен пример вызова этих функций с помощью хэндлеров.

```
defregister_handlers(dp: Dispatcher):
dp.register_message_handler(payment_menu, lambda message:
message.text = "Управление балансом" )
dp.register_callback_query_handler (operation_money_callback, lambda query:
query.data.startswith('operation_'))
dp.register_callback_query_handler(confirm_return_money, lambda query:
```

query.data = 'confirm_return')

dp.register_callback_query_handler(add_balance,lambda query:

query.data.startswith('add_money_'))

dp.register_pre_checkout_query_handler(checkout_process,lambdaq: True)

Модуль command_menu.py – раздел, в котором объединены интерфейсы инструктора и администратора.

Функция admin_sign отвечает за авторизацию администратора. Она проверяет, является ли отправитель сообщения администратором, и, если это так, запрашивает пароль.

Функция принимает сообщение message, объект состояния state и необязательный параметр id, который используется для указания альтернативного идентификатора пользователя (например, при входе в систему от имени другого пользователя).

Функция начинается с удаления сообщения, чтобы скрыть информацию о логине и пароле администратора.

Затем она читает данные администратора из файла admin.txt, включая логин и хэшированный пароль.

Функция проводит процедуру верификации роли отправителя, сопоставляя его идентификатор с учетными данными администратора.

В случае успешного совпадения, система переходит к следующей стадии, а именно, хэшированию пароля администратора, который сохраняется в переменном состоянии state.

После этого формируется и отправляется запрос на возможность ввода пароля, что позволяет установить соответствующее состояние LoginStates.PASSWORD_STATE.

Функция password_handler является обработчиком сообщений, которые ожидают пароль администратора. Она получает введенный пароль от пользователя и сравнивает его с сохраненным хэшированным паролем администратора. Если пароль верный, функция завершает состояние state, иначе она отправляет сообщение о отказе в доступе.

Функция admin_menu отображает меню администратора после успешной авторизации. Она создает и отправляет клавиатуру с опциями для изменения различных таблиц (студенты, группы, расписание и т. д.) и ожидает ответа от пользователя.

Функция teacher_menu представляет меню преподавателя. Если пользователь имеет права преподавателя, отображается меню с различными функциями. Если пользователь не имеет прав доступа, выводится сообщение о отсутствии доступа.

Функция teacher_mode обрабатывает запросы, связанные с функциями меню преподавателя. Она принимает объект types. Callback Query, который представляет callback-запрос от пользователя.

Функция содержит несколько условий для различных типов запросов, например, запросы для просмотра данных преподавателя, просмотра расписания на день, отмены занятий или отметки ученика.В каждом условии выполняются соответствующие действия, такие как удаление сообщения, получение данных из базы данных, создание клавиатуры с кнопками и отправка сообщения с клавиатурой.

Функция get_lessons_day обрабатывает запросы, связанные с выбором дня и просмотром/отменой занятий в этот день. Она также принимает объект types.CallbackQuery.

Функция проверяет тип запроса (просмотр, отмена или отметка) и выполняет соответствующие действия. Например, если запрос связан с просмотром занятий, получается список занятий на выбранный день, создается клавиатура с кнопками для каждого занятия, и отправляется сообщение с клавиатурой.

Аналогичные действия выполняются для других типов запросов, таких как

отмена занятий или отметка ученика.

Функции teacher_menu, teacher_mode и get_lessons_day используются вместе для создания меню преподавателя и обработки связанных запросов.

Ниже приведен пример вызова этих функций с помощью хэндлеров.

```
defregister_handlers(dp: Dispatcher): dp.register_message_handler
(teacher_menu,
commands=['teacher'] )
dp.register_message_handler(admin_sign, commands=['admin'])
dp.register_callback_query_handler(teacher_mode, lambda query:
query.data.startswith ('teach_'))
dp.register_callback_query_handler(admin_mode, lambda query:
query.data.startswith ('admin_'))
dp.register_callback_query_handler(get_lessons_day, lambda query:
query.data.startswith ('tach-les'))
dp.register_callback_query_handler(admin_option, lambda query:
query.data.startswith ('adm_'))
dp.register_callback_query_handler(return_mode, lambda query:
query.data.startswith ('t-return'))
```

```
dp.register_callback_query_handler (view_mode, lambda query:
query.data.startswith ('t-view'))
dp.register_callback_query_handler (ed_mode, lambda query: query.data.
startswith ('ed-'))
dp.register_callback_query_handler(rules_mode, lambda query: query.data.
startswith('t-rul'))
dp.register_message_handler(add_mode, state=[DBStates.newProgram,
DBStates.newClass, DBStates.newTeacher],
content_types=types.ContentTypes.TEXT)
dp.register_message_handler(del_mode, state=[DBStates.delProgram, TEXT)
dp.register_message_handler (edit_mode, state= [DBStates.editProgram,
DBStates.editExam, DBStates.editTable, DBStates.editGroup,
DBStates.editStudent, DBStates.editCategory, DBStates.editCar,
DBStates.editClass, DBStates.editTeacher],
content_types=types.ContentTypes.TEXT)
dp.register_message_handler (ed_op, state= [DBStates.newrow Program,
DBStates. newrowExam, DBStates.newrow Table, DBStates. Newrow Group
DBStates.newrow Student,
DBStates.newrow Category, DBStates.newrowCar, DBStates.newrowClass,
DBStates.newrowTeacher], content_types=types.ContentTypes.TEXT
```

Модульother.py – раздел, в который попадают все сообщения, которые не попали в предыдущие модули, обработка ошибочного сообщения.

Функция echo используется для обработки сообщений, которые не могут быть распознаны или поняты моделью. Она принимает сообщение пользователя и отправляет обратно то же самое сообщение с призывом вернуться в главное меню командой /menu.

```
def register_handlers (dp:Dispatcher):
dp.register_message_handler (echo)
```

3.2 Оценка экономической эффективности проекта

Оценка экономической эффективности внедрения информационной

системы заключается в сравнении полученных от нее выгод с минимальными затратами на ее создание и техническую поддержку. Главная задача – это понять, насколько финансово оправдано внедрение, оценивая соотношение всех вложений в программный продукт и его отдача в итоге. Для понимания финансовых аспектов, рассмотрим цифровую архитектуру автошколы, предложенную И.В.Гейцем и П.К.Шафранским. По их принципиально высказанному мнению эта архитектура соответствует бизнес-модели и определяет насколько успешно будут интегрировать цифровые инструменты в работе самой организации [8,с.159].

Цифровая архитектура – это готовый набор компонентов, которые обеспечивают непрерывно возможность внедрения различных технологий, таких как аппаратное обеспечение, сервисы, средства защиты и программные средства интеграции. Поэтому, прежде чем внедрить информационную систему в процесс обучения автошколы, необходимо иметь соответствующую цифровую архитектуру. Помимо очевидного преимущества в виде ускорения выполнения определенных производственнных задач, сам процесс внедрения, при условии соблюдения всех необходимых процедур, так же приносит экономические выгоды, включая следующие:

— сокращение издержек автошколы. Связано с перестройкой бизнес-модели с традиционного в смешанный цифровой характер образовательных функций, позволяет сократить долгосрочные издержки по аренде помещений для проведения занятий, а также снизить себестоимость услуг ввиду применения телеграм-бота;

— комплексного сокращения издержек по оплате труда, оптимизации бизнес-процессов, повышению эффективности обучения и т.п. результатам;

— улучшение пиар и маркетинговых функций, а именно продвижение в формате онлайн-бизнеса, что обеспечивает большую результативность применения маркетинговых инструментов, позволяет внедрять систему скидок, горячих предложений. Комплекс маркетинговых мероприятий становится более выгодным для автошколы, что сопряжено с более точным использованием

инструментов;

— качественный сбор данных и цифровая аналитика. Сбор данных о бизнес-процессах, действиях клиентов и изменениях показателей позволяет более точно реагировать на внутренние изменения и поддерживает принятие управленческих решений, что позволяет сократить уровень финансовых рисков от реализации преобразований.

Несомненно, успешная цифровая трансформация окупает вложенные денежные средства и приводит к увеличению доходов. Такие результаты подтверждают несомненно, эффективность самого процесса внедрения информационной системы, что напрямую связано с грамотно выстроенной работой цифровой архитектуры и информационной системы соответственно. Дополнительными показателями успеха можно считать повышение процента успешно обученных курсантов, минимизацию сроков обучения, выход на новые горизонты рынка образовательных услуг и т.д.

В нашем случае проект внедрения ИСбудетиметьследующиеисточники доходов:

— продажа - команда будет получать доход от разработки и настройки информационной системы для каждой автошколы или коммерческого образовательного учреждения. Стоимость разработки для каждого клиента будет составлять примерно 20 000 рублей;

— внедрение - будет начисляться оплата за процесс внедрения информационной системы в каждой организации. Это включает установку и настройку бота, адаптацию системы под потребности клиента. Стоимость внедрения составит около 5 000 рублей для каждой организации;

— поддержка - нужно будет предоставлять услуги поддержки клиентам в течение нескольких месяцев после внедрения системы. Плата за поддержку составит 1 000 рублей для каждой организации ежемесячно;

масштабируемость франшизы - предусмотрена возможность расширения системы на другие филиалы автошколы и другие коммерческие образовательные учреждения в городе и за его пределами. За каждое новое

внедрение системы предусмотрена дополнительная плата в размере 5 000 рублей за филиал.

Доходвысчитываетсяпоформуле:

$$Д = К \times Ц \quad (3.1)$$

где, Д – доход;

К – количество потребителей;

Ц – установленная цена

Доход от поддержки будет высчитываться следующим образом: 1000 руб. * 50 курсантов = 50 000 руб/мес.

Доход от масштабируемости высчитывается следующим образом: 5 000 руб. * 100 = 500 000 руб.

Таким образом, общий ожидаемый доход за год, с учетом набора 10 групп составит 13 500 000 рублей.

Для запуска, функционирования и развития бизнеса потребуются некоторые ключевые ресурсы. В нашем случае это будут:

— разработчики программного обеспечения, которые специализируются на создании телеграм-ботов и ИС;

— менеджерыпопродажам,обладающиенавыкамипродажииумением вести переговоры с потенциальными клиентами;

— компьютеры,серверы,программноеобеспечениеидругоетехническое оборудование для разработки и обслуживания системы;

— офисноепространстводляработыкомандыипроведенияявстречс клиентами;

— коммуникационноеоборудованиедлясвязи;

— финансовые ресурсы для финансирования разработки, маркетинга, продаж и операционных расходов бизнеса.

Приведенные ключевые финансовые ресурсы отражены в таблице 3.1.

Таблица 3.1 – Анализ капитальных затрат на разработку и внедрение ИС

Направление использования средств	Кол-во, шт.	Цена, руб.	Сумма, руб.
Аренда сервера		3 000	36 000
Аренда помещения		4 000	48 000
Приобретение ноутбуков	2	100 000	200 000
Приобретение другого оборудования	1	100 000	100 000
Коммунальные услуги		5 000	60 000
З/п менеджеру		30 000	360 000
Итого		242 000	804 000

Далее перейдем к оценке экономической эффективности, которая рассчитывается по формуле:

$$Э_{эф} = Э/К \quad (3.2)$$

где, $Э_{эф}$ - экономическая эффективность от реализации того или иного мероприятия;

Э – экономический эффект прибыль, руб.;

К – величина капитальных затрат на реализацию мероприятия, руб.

Учитывая ранее полученный результат экономического эффекта в размере 13 500 000 и капитальные затраты на разработку и внедрение информационной системы получим следующую экономическую эффективность: 13500000/804000=16,79≈17%

Посколько разработка программного продукта во время написания выпускной квалификационной работы находится на стадии проектирования, точная оценка ее будущей эффективности пока невозможна.

Для определения более точных перспектив вывода проекта на рынок необходимо провести SWOT-анализ, который позволит выявить внутренние сильные и слабые стороны разработки, а также понять внешние возможности и оценить потенциальные угрозы, которые могут иметь место и повлиять на

успех. На первом этапе проведем детальное изучение внутренних критериев оценки и внешних факторов деятельности.

В таблице 3.2 представлена матрица SWOT-анализа.

Таблица 3.2 – Матрица SWOT-анализа информационной системы

Сильные стороны	Возможности во внешней среде
С1. Понятное и адаптивное приложение С2. Функциональная мощность С3. Стабильность и скорость работы системы	В1. Использование актуальных и популярных инструментов разработки В2. Возрастающая потребность в цифровизации В3. Появление дополнительногоспроса на новый продукт
Слабые стороны	Угрозы во внешней среде
Сл1. Зависимость от быстро меняющихся технологий Сл2. Значительные временные и интеллектуальные затраты	У1. Несоответствие требованиям пользователей У2. Отсутствие спроса на данное решение

На втором этапе проводится детальная оценка соответствия внутренних ресурсов самого проекта, т.е. его сильных и слабых сторон внешним факторам среды.

Цель данного этапа сводится к заключению определения степени необходимости внесения стратегических корректировок.

Для облегчения анализа были разработаны интерактивные матрицы проекта (таблица 3.3-3.4), которые наглядно иллюстрируют различные варианты комбинаций взаимосвязей всех элементов SWOT-анализа.

Каждый фактор помечается либо знаком «+» – сильное соответствие сильных/слабых сторон возможностям/угрозам, знаком «–» – слабое соответствие, либо «○» в случае сомнения.

Таблица 3.3 – Интерактивная матрица сильных и слабых сторон с учетом предполагаемых возможностей

		Сильные стороны			Слабые стороны	
		С1	С2	С3	Сл1	Сл2
Возможности проекта	В1	+	○	+	+	+
	В2	+	○	+	–	–
	В3	+	+	+	–	–

Корреляциювозможностейиугрозссильнымиислабымисторонамиможнозаписать в данной форме: В1В2В3С1,В3С2,В1В2В3С3, отсюда:

— В1Сл1Сл2

— У1С1С2

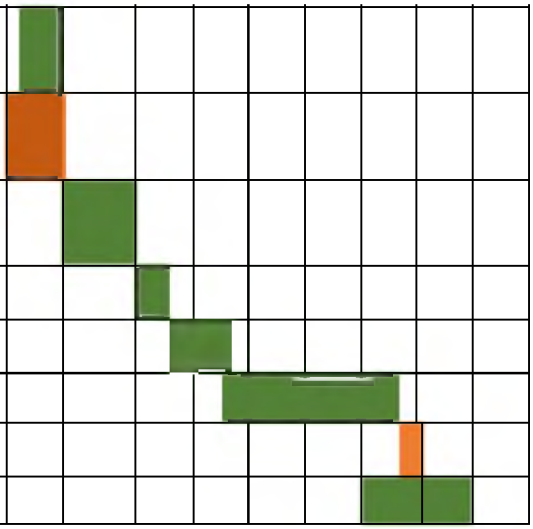
Таблица 3.4 – Интерактивная матрица сильных и слабых сторон с учетом потенциальных угроз

		Сильные стороны			Слабые стороны	
		С1	С2	С3	Сл1	Сл2
Угрозы проекта	У1	+	+	–	○	○
	У2	○	–	○	–	–

Несмотря на ограниченные ресурсы и каналы непосредственного продвижения, что естественно затрудняет развитие в дальнейшем самого проекта, высокая функциональность и адаптивность приложения позволяют получить конкурентное преимущество перед другими автошколами Туапсинского округа в будущем. Для детального планирования работ, учитывая и определения сроков исследования был составлен календарный план-график, разбитый по месяца (таблица 3.5).

Таблица 3.5 – Календарный план-график проведения исследования

№	Название работы	Продолжительность выполнения работ								
		Сентябрь	Октябрь	Ноябрь	Декабрь	Январь	Февраль	Март	Апрель	Май
1	Определение цели и задач проектирования									
2	Создание календарного плана-графика									
3	Определение содержания и согласование технических характеристик и архитектуры бота									

4	Дизайн пользовательского интерфейса Телеграм-бота	
5	Постановка требований к программному обеспечению	
6	Определение контента и коммуникационной стратегии бота	
7	Разработка бюджета проекта	
8	Разработка информационной системы	
9	Тестирование и отладка телеграм-бота	
10	Система аналитики и отчетности	
11	Подготовка документации	

Отметим, что именно календарные планы часто наглядно визуализируются при помощи диаграммы Ганта, т.е. горизонтального графика, отображающего продолжительность по времени задач, с указанием дат начала и окончания ранее намеченных мероприятий.

Социологический опрос респондентов в лице обучающихся автошкол города Туапсе в количестве 55 человек разного пола и возраста выявил единодушное одобрение внедрения телеграм-бота в учебный процесс приполучении профессии водителя различных категорий.

Таким образом, подводя итог оценки эффективности разработанной информационной системы, которая нацелена на разработку концепции развития автошколы «Навигатор+», можно с уверенностью констатировать ее высокую экономическую и социальную значимость, заслуживающую положительной оценки.

Заключение

Разработка информационной системы, как правило, выполняется для вполне определенного предприятия или организации для разработки определенной концепции всего информационного пространства.

Специфика работы предприятия, безусловно, влияет на его информационную систему. Однако, несмотря на эти отличительные характеристики, информационные системы разных компаний демонстрируют общую структурную схожесть. Каждая организация, вне зависимости от профиля, организована по принципу создания структурных подразделений, отвечающих за конкретные направления деятельности. Данная модель применима практическико всем организациям и фирмам, не зависимо от специфики деятельности и формы собственности.

Внедрение передовых информационных технологий позволяет безусловно, ускорить разработку маркетинговых и производственных проектов, сократить непроизводственные расходы, исключить допущение серьезных ошибок в различного рода документации, а именно бухгалтерской, технической и пр., что приводит к ощутимой экономической выгоде для практическ всех организаций и предприятий.

Чтобы в полной мере использовать потенциал информационной системы, важно применять аппаратное программное обеспечение, наилучшим образом отвечающее конкретным задачам. Это объясняет высокий спрос на программы по управлению и знания эффективного использования имеющейм=ся компьютерной техники на предприятиях. Результаты данной выпускной квалификационной работы соответствуют поставленной цели. В рамках исследования по разработке концепции информационной системы были детально решены следующие задачи:

— проведен анализ процессов и потребностей автошколы, исследованы требования и ожидания курсантов, что позволило определить функциональность и основные модули системы;

осуществлен выбор мессенджера. После анализа различныхмессенджеров было принято решение использовать Telegram, обусловленное его популярностью, удобством использования и наличием широкого функционала;

— выбран язык программирования. В качестве языка программирования для разработки информационной системы был определен Python, который обладает простым и понятным синтаксисом, множеством библиотек и фреймворков, что облегчает разработку и ускоряет процесс создания системы;

— определена система управления базами данных. В качестве СУБД для хранения и управления данными была выбрана PostgreSQL, которая является мощной и надежной системой, обладающей широкими возможностями, включая поддержку сложных запросов и транзакций;

— осуществление выбора хостинга. Выбран сервис TimeWeb.Cloud, обеспечиваюший надежное размещение информационной системы и обладаюший преимуществами в виде высокой производительности, надежности, безопасности, гибкости настройки и доступной цены;

— разработан функционал системы. Были реализованы основные модули системы, такие как регистрация пользователей, расписание занятий, информация об инструкторах и автомобилях, онлайн-оплата и многое другое.

Разработанная концепция информационной системы в ходе написания выпускной квалификационной работы обеспечивает полный и оперативный доступ ко всем данным АНО ДПО Автошкола «Навигатор+». Курсанты получают возможность записи в легком формате на все занятия, как теоретические, так и практические, производить оплату, получаьб уведомления различного характера и контактировать как с администрацией автошколы, так и с одногруппниками через Telegram. Экономическая эффективность при расчетах составила 17%, что несомненно подчеркивает ее успешность.

Дальнейшее развитие системы предполагает расширение функциональных возможностей системыв, за счет добавления новых модулей, улучшения интерфейса и оптимизации рабочих процессов.

Разработанная информационная система интегрирована с мессенджеромTelegram прошла этап разработки и готова к внедрению в автошколе. Ее применение позволит существенно оптимизировать коммуникационную составляющую с клиентами, повысить операционную эффективность, укрепить позиции на рынке предоставления образовательных услуг дополнительного образования, и как следствие, положительно сказаться на финансовых показателях автошколы в дальнейшем.

Список литературы

1. Адигеев, М.Г. Жизненный цикл программного обеспечения / М.Г. Адигеев. – Ростов-на-Дону: Изд-во ЮФУ, 2023. – 401 с.

2. Аквино, К., Ганди, Т. Клиентская разработка для профессионалов: Node.js, ES6, REST.- СПб.: Питер, 2021. -512 с.

3. Бабанов, А.М. Технология разработки программного обеспечения: структурный подход. – Томск: ТГУ, 2025. – 157 с.

4. Балдин, К.В. Информационные системы в экономике. – М.: Дашков и К, 2025. – 395 с.

5. Блюмин, А.М. Экономические расчеты в информатике: учеб.пособие. – М.: Издательско-торговая корпорация «Дашков и Ко», 2025. – 384 с.

6. Варзунов, А.В. Анализ и управление бизнес-процессами: учеб. пособие. – Санкт-Петербург: Университет ИТМО, 2022. – 114 с.

7. Гагарина, Л. Г. Инструменты для создания и реализации приложений: учеб. пособие. – Москва: ИНФРА - М, 2023. - 400 с.

8. Гвоздева, В.А., Ворфелева, Е.Н., Неорович, И.С. Информатика, автоматизированные информационные технологии и системы: учеб. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2025. – 544 с.

9. Долженко, А.И. Разработка React. – Ростов-на-Дону: Изд-во РГУ, 2023. – 191 с.

10. ИзучениеReact. [Электронный ресурс] – URL: https://habrahabr.ru. (дата обращения: 09.09.2025).

11. Котляров,В.П. Методы расчета экономической эффективности программных продуктов. - Москва: ИНТУИТ, 2024. – 335с.

12. Кумагина, Е.А. Модели жизненного цикла и технологии проектирования программного обеспечения. – Нижний Новгород: изд-во ННГУ, 2022. – 157 с.

13. Коцюба, И.Ю. Структура клиентского приложения: учеб. пособие.– СПб: Университет ИТМО, 2025. – 206 с.

14. Луб, Е.Н. Разработка веб-приложений в ReactJS. - М.: ДМК Пресс, 2023. - 254 с.

15. Моделирование информационных систем: учеб. пособие / по ред. Лисяк, В.В.. – Ростов-наДону: Изд- во Южного федерального университета, 2022. – 88 с.

16. Радченко, М.Г., Невершестов, Р.О. Интерактивные дашборды и приложения с Plotly и Dash. - М.: 1С-Паблишинг, **2022**. - 194 с.

17. Рудинский, И.Д. Технология проектирования автоматизированных систем обработки информации и управления: учеб. пособие / И. Д. Рудинский. – М.: Горячая линия - Телеком, 2024. - 304 с.

18. Рязанцева, Н.Е. CASE-технологии. Современные методы и средства проектирования информационных систем. - М.:БХВ-Петербург, **2022**. - 694 с.

19. Тестирование информационных систем. [Электронный ресурс] – http://services/testing/testirovanie-sistem.(датаобращения: 10.11.2025).

20. Томас, М.Т. React в действии. – Санкт-Петербург: Издательский Дом Питер. – 368 с.

21. Тюгашев, А.А. Основы программирования веб-приложений. – СПб.: Университет ИТМО, 2025. – 160 с.

22. Финансовые приложения для учета личного бюджета. [Электронный ресурс]– http://blog_897/prilozheniya-dlya-uchota-finansov.(датаобращения 29.08.2025).

23. Чистякова, В.И. Проектирование информационных систем: учеб. пособие для бакалавров. – М.: Академия, 2025. – 301 с.

24. Чистов, Д. В. Проектирование информационных систем. – М.: Юрайт, 2023. – 260 с.

25. Шеремет, А.Д. Методика финансового анализа: учеб. пособие. - 2- е изд., перераб. и доп. – М.: Наука, 2023. - 518 с.

26. Функциональное моделирование информационных процессов.[Электронныйресурс]–URL:http://www./designing/methodology _for_bp.html. (дата обращения 18.08.2025).

27. Филатова, В.И. Базы данных. - М.:Сфера, **2022**. - 256 с.

28. Харитонов, С.А., Пронь, И.И. Архитектура информационных систем: учеб.пособие для вузов. - М.:1С-Паблишинг,**2024**. - 682 с.

29. Шарина, А.А. Языки программирования. – М.: Технологии будущего, 2023. - 292 с.

30. Языки программирования высокого уровня. [Электронный ресурс] URL: https://ru.wikipedia.org (дата обращения 20.09.2025).