Министерство науки и высшего образования РФ

Читинский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Байкальский государственный университет»

Кафедра информационных технологий и высшей математики

Выпускная квалификационная работа

РАЗРАБОТКА ПЛАТФОРМЫ ДЛЯ ДИСТАНЦИОННОГО ОБУЧЕНИЯ, ОЦЕНКИ ЗНАНИЙ И ПОМОЩИ ПО СОЗДАНИЮ ПРОЕКТОВ НА ПРИМЕРЕ ГКУ «РЕСУРСНЫЙ ЦЕНТР РАЗВИТИЯ НЕКОММЕРЧЕСКИХ ОРГАНИЗАЦИЙ ЗАБАЙКАЛЬСКОГО КРАЯ

|  |  |  |
| --- | --- | --- |
| Исполнитель(ли) |  | Коноплёва Альбина Павловна, ИСТУ-21 |
|  | (подпись, дата) |  |
| Руководитель(ли) |  | старший преподаватель кафедры ИТиВМ Печерина Александра Валерьевна |
|  | (подпись, дата) |  |
| Рецензент (ты) |  | начальник отдела учебно-методического и информационного обеспечения Куклина Ольга Константиновна |
|  | (подпись, дата) |  |
| Нормоконтроль |  | старший преподаватель кафедры ИТиВМ Печерина Александра Валерьевна |
|  | (подпись, дата) |  |
| К защите допускаю  Зав.кафедрой |  | к.ф.-м.н., доцент Трухина Людмила Ивановна |
|  | (подпись, дата) |  |

Чита, 2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc199965619)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc199965620)

[1.1 Общая характеристика организации 7](#_Toc199965621)

[1.2 Основные направления деятельности организации 9](#_Toc199965622)

[1.3 Описание функциональной структуры предприятия 10](#_Toc199965623)

[1.4 Общая информация о предоставлении учебных материалов, обучении и мерах поддержки НКО с использованием информационных технологий в ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края» подразделении "Центр по работе с некоммерческими организациями" 11](#_Toc199965624)

[1.5 Анализ бизнес-процессов предметной области 13](#_Toc199965625)

[1.6 Бизнес-процессы веб-приложения 16](#_Toc199965626)

[1.7 Системный анализ 17](#_Toc199965627)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 20](#_Toc199965628)

[2.1 Построение диаграмм IDEF0, DFD и IDEF3 20](#_Toc199965629)

[2.2 Построение ER-диаграммы 24](#_Toc199965630)

[2.3 Описание проекта в нотации UML 26](#_Toc199965631)

[2.4 Планирование 30](#_Toc199965632)

[ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОЕКТА 32](#_Toc199965633)

[3.1 Выбор инструментов для разработки 32](#_Toc199965634)

[*3.1.1 Язык программирования* 32](#_Toc199965635)

[*3.1.2 Веб-фреймворк* 33](#_Toc199965636)

[*3.1.3 СУБД* 35](#_Toc199965637)

[3.2 UX-разработка 37](#_Toc199965638)

[3.3 Описание общей структуры проекта 37](#_Toc199965639)

[3.4 Настройки проекта 39](#_Toc199965640)

[3.5 Авторизация и регистрация 42](#_Toc199965641)

[3.6 Заявки на курсы 42](#_Toc199965642)

[3.7 Заполнение курсов и просмотр их содержимого 42](#_Toc199965643)

[3.8 Посещаемость и тесты 42](#_Toc199965644)

[3.9 Обсуждение внутри курса 42](#_Toc199965645)

[3.10 Новости 42](#_Toc199965646)

[3.11 Статистика 42](#_Toc199965647)

[3.12 Создание бота для уведомлений 42](#_Toc199965648)

[3.13 Описание интерфейса веб-приложения 42](#_Toc199965649)

[3.14 Публикация проекта на веб-хостинге GitHub 43](#_Toc199965650)

[ГЛАВА 4. ТЕСТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ 45](#_Toc199965651)

[ГЛАВА 5. АНАЛИЗ ЭФФЕКТИВНОСТИ 47](#_Toc199965652)

[ЗАКЛЮЧЕНИЕ 48](#_Toc199965653)

# **ВВЕДЕНИЕ**

Грантовая система, как и в принципе само понятие социального проектирования зародилось изначально в европейских странах. В свою очередь многие российские некоммерческие организации хорошо финансировались зарубежными фондами целью которых является развитие социально-значимых проектов. Однако несмотря на все возможные преимущества такого финансирования для бюджетных учреждений всё же имелись определённые риски и обязательства перед иностранными фондами. Поэтому в России была сформирована своя внутренняя система, не зависимая от фондов других стран и направленная на финансирование НКО по средством ранжирования на конкурсной основе. В результате формирования такой системы в стране появилось огромное количество федеральных фондов, принимающих заявки на финансовое обеспечение социально-значимых проектов со стороны НКО. Естественно, помимо федерального уровня было создано региональное направление грантовой системы, что позволило стимулировать развитие инфраструктуры городов регионам самостоятельно. И на данном этапе появилась проблема. Откуда региональным НКО получить опыт подачи конкурсных заявок на уровне организаций, имеющих его на уровне сотрудничества с зарубежными фондами? Конечно же Забайкальский край не стал исключением и тоже столкнулся с подобной проблемой. И так в августе 2020 года было образовано ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края». Неотъемлемой частью его деятельности является консультирование, проведение обучения в сфере социального проектирования и организация тренингов для всех НКО региона.

Несмотря на возможность использования современных информационных технологий в формате социальных сетей для информирования о формах подачи грантовых заявок и распространения обучающей информации процесс работы центра с НКО не стал удобней для обеих сторон, так как не учитывается необходимость автоматизации учебного процесса. Ведь по итогу нет централизованного сетевого ресурса для бюджетных учреждений, нуждающихся порой в совершенно разных мерах грантовой поддержки.

Настоящая работа посвящена всестороннему анализу бизнес-процессов организации, деятельность которой связана с обучением граждан в сфере социального проектирования, оказанием консультационной поддержки при разработке собственных грантовых инициатив и информированием населения о законодательных изменениях и новых мерах государственной поддержки. Основной акцент сделан на выявлении неэффективных участков в текущих процессах, понимании причин их возникновения и поиске решений, направленных на устранение выявленных проблем.

Основными целями дипломного проекта являются развитие навыков проведения аналитических исследований, освоение методик моделирования информационных систем в соответствии с современными стандартами и получение практического опыта работы с программным обеспечением, применяемым в реальных условиях. В рамках работы планируется изучение структуры выбранной организации, определение ключевых направлений её функционирования и документирование существующих бизнес-процессов. Анализ текущего состояния показал отсутствие систем автоматизации, что существенно снижает эффективность выполнения задач, особенно в сфере бухгалтерского и операционного учёта.

С целью устранения этих недостатков запланирована разработка автоматизированной информационной системы, способной минимизировать затраты времени сотрудников и повысить точность обработки информации. На основании подготовленного технического задания, включающего функциональные и пользовательские требования, выполнено проектирование интерфейса пользователя и создана структура базы данных. Для реализации функциональности системы был проведён обзор современных технологий, из которых был сформирован оптимальный стек разработки, включающий язык программирования, соответствующие библиотеки, фреймворки и СУБД.

Результатом выполнения проекта станет готовое к использованию веб-приложение, внедрённое в работу организации и способное существенно повысить эффективность предоставляемых услуг. Помимо самого программного продукта, будут подготовлены рекомендации по его применению и техническому сопровождению. Разработанная система станет универсальным инструментом для расширения образовательных возможностей сотрудников некоммерческих организаций, повышения их компетентности в сфере грантовой деятельности, а также автоматизации процессов информирования и консультирования.

Для достижения целей сформулированы следующие задачи:

* провести анализ предметной области, изучить структуру организации и зафиксировать основные бизнес-процессы;
* определить процессы, подлежащие оптимизации с помощью информационной системы;
* разработать архитектуру системы с применением методологий функционального моделирования (IDEF0, DFD, IDEF3) и объектного моделирования (UML-диаграммы);
* сформировать поэтапный план реализации проекта с контролем выполнения задач;
* выбрать и обосновать инструменты и технологии, необходимые для разработки;
* создать веб-приложение с учетом особенностей конкретной организации и адаптировать его к текущим условиям эксплуатации.

# **ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

В данной главе проводится сбор и анализ информации, необходимой для понимания специфики рассматриваемой предметной области, в качестве которой выступает конкретная организация. Этот процесс позволяет определить ключевые особенности внутренней структуры и функционирования организации, что в дальнейшем способствует более точному планированию проектной деятельности, выявлению уникальных требований к системе и определению сроков выполнения каждого этапа практической работы.

Первым шагом в рамках анализа является построение модели предметной области, отражающей основные процессы, участников и их взаимодействие. Созданная модель служит основой для дальнейшего проектирования системы и позволяет формализовать требования к будущему решению.

На следующем этапе проводится оформление полученной модели в виде документации, включающей как графические, так и текстовые описания. Это необходимо для обеспечения четкого понимания всех аспектов модели как со стороны разработчиков, так и заинтересованных лиц в организации.

Финальной стадией раздела является переход от теоретической модели к практической реализации, включая подготовку к внедрению информационной системы и планирование действий, связанных с её вводом в эксплуатацию.

## **1.1 Общая характеристика организации**

Осуществление проектирования и разработки веб-приложения будет выполняться на базе ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края» в структурном подразделении "Центр по работе с некоммерческими организациями".

Данный подразделение осуществляет свою деятельность с целью поддержки НКО Забайкальского края.

Основные функции подразделения:

* помощь в формировании и подаче заявок некоммерческих организаций Забайкальского края для участия в грантовых проектах;
* помощь региональным операторам в работе над социально-значимыми проектами и обработке информационных запросов;
* курирование сотрудников НКО по грантовым направлениям;
* обучение сотрудников НКО по направлению социального проектирования;
* информирование населения об изменениях в законодательстве и появлении новых мер грантовой поддержки.

Государственное казённое учреждение «Ресурсный центр развития некоммерческих организаций Забайкальского края» было создано в 2020 году по инициативе губернатора региона. Основной задачей центра является содействие развитию гражданского общества через поддержку социально ориентированных некоммерческих организаций. Это достигается путём консультирования, методической помощи, а также сопровождения проектной и грантовой деятельности.

Учреждение активно способствует распространению современных технологий и эффективных практик, направленных на решение социально значимых задач. Особое внимание уделяется формированию устойчивых механизмов поддержки некоммерческого сектора, что позволяет организациям повышать качество своей работы и расширять сферу влияния.

Важным элементом структуры Ресурсного центра стало его объединение с Аппаратом Общественной палаты Забайкальского края. Такое объединение позволило выстроить более эффективное и всестороннее взаимодействие с НКО региона, а также наладить диалог между гражданским обществом и органами власти.

Таким образом, деятельность Ресурсного центра направлена на создание благоприятной среды для развития некоммерческого сектора, формирование устойчивых социальных инициатив и укрепление института гражданского участия в решении значимых общественных вопросов.

## **1.2 Основные направления деятельности организации**

Одним из ключевых направлений деятельности ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края» является предоставление образовательных услуг в сфере социального проектирования. Центр активно занимается обучением сотрудников некоммерческих организаций, нуждающихся в поддержке при разработке и реализации грантовых инициатив. Образовательный процесс построен на сочетании теоретических и практических форм, позволяющих участникам получить как базовые знания, так и индивидуальное сопровождение на всех этапах проектной деятельности.

Обучение организуется в различных форматах, включая очные и дистанционные консультации, лекции, тренинги, а также распространение текстовых учебных материалов. Семинары по проектным технологиям охватывают весь путь — от первоначальной идеи до полной реализации проекта. Обучение может проходить как индивидуально, так и в группах, с возможностью выбора удобного формата — очного или дистанционного. В зависимости от потребностей НКО доступны следующие формы участия:

* индивидуальное сопровождение на всех этапах работы над проектом (очно или онлайн);
* групповое консультирование и обучение в рамках курсов по социальному проектированию;
* специализированное персональное обучение, включающее поэтапную проработку проекта;
* возможность закрепления постоянного куратора, сопровождающего организацию на протяжении всей грантовой кампании.

Кроме образовательной деятельности, подразделение выполняет и другие значимые функции. Оно занимается организацией собраний и пленарных заседаний Общественной палаты Забайкальского края, что обеспечивает постоянный диалог между институтами гражданского общества и органами власти. Также центр проводит социально значимые мероприятия в рамках региональной политики, направленные на развитие гражданской активности.

Дополнительно Ресурсный центр координирует волонтёрскую деятельность, предоставляя платформу для взаимодействия добровольцев и НКО. По запросу, организациям предоставляется доступ к материально-технической базе — оборудованию и ресурсам, необходимым для эффективной работы. Также регулярно проводятся региональные конкурсы среди НКО, стимулирующие развитие новых социальных инициатив и повышение профессионального уровня их реализации.

Таким образом, деятельность центра носит комплексный характер и охватывает как обучение и сопровождение НКО в вопросах проектирования, так и создание условий для полноценного функционирования и развития некоммерческого сектора в регионе.

## **1.3 Описание функциональной структуры предприятия**

Структура ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края»:

* руководитель;
* Аппарат общественной палаты (руководитель подразделения, главные специалисты). Работа с социально значимыми волонтёрскими организациями и сопровождение их деятельности;
* Центр по работе с НКО (руководитель подразделения, главные специалисты, специалист по связям с общественностью). Осуществляет непосредственное взаимодействие с НКО по грантовым направлениям;
* Управление (главный бухгалтер, ведущий бухгалтер, главный специалист по кадровым вопросам и делопроизводству, технический редактор, юрисконсульт) (Рисунок 1). Регулировании совместно с отображением административной и хозяйственной деятельности организации.

Рисунок 1.1 – Структура ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края»

На основе собранной информации о предметной области удалось сформировать целостное представление о специфике возникновения и развития организации.

Благодаря ей стало возможным точно определить ключевых участников бизнес-процессов и выстроить модель взаимодействия, необходимую для последующей автоматизации

## **1.4 Общая информация о предоставлении учебных материалов, обучении и мерах поддержки НКО с использованием информационных технологий в ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края» подразделении "Центр по работе с некоммерческими организациями"**

В рамках реализации своей ключевой задачи — содействия развитию гражданского общества в регионе — ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края» активно организует и проводит разнообразные мероприятия, направленные на поддержку социально ориентированных некоммерческих организаций (НКО). Деятельность центра ориентирована не только на образовательную и консультативную помощь, но и на создание условий для устойчивого цифрового взаимодействия с представителями некоммерческого сектора.

Цифровизация процессов взаимодействия с НКО имеет чётко сформулированные цели и задачи. Во-первых, это обеспечение коммуникации между учреждениями края без необходимости очного присутствия, что особенно актуально для отдалённых районов. Во-вторых, ведётся активная работа по популяризации грантовой деятельности — одним из важнейших инструментов поддержки инициатив гражданского общества. Кроме того, в числе приоритетов — развитие инфраструктуры региона, выявление и поддержка малых и начинающих некоммерческих организаций, а также повышение уровня прозрачности в их деятельности, что способствует росту общественного доверия и привлечению добровольцев.

Сейчас в качестве основного цифрового инструмента центр использует официальную группу во «ВКонтакте». Через неё быстро публикуются новости, выкладываются полезные материалы, проводятся опросы — в целом, это удобный способ оставаться на связи с некоммерческим сообществом и понимать, какие темы сейчас наиболее актуальны. Но чтобы вывести взаимодействие с НКО на новый уровень, одной группы в соцсети уже недостаточно.

Однако в целях повышения эффективности взаимодействия с НКО, целесообразно рассмотреть возможность разработки специализированной онлайн-платформы. Такая система позволит автоматизировать ряд ключевых бизнес-процессов, включая регистрацию организаций на курсы, подачу заявок на гранты, получение консультаций и сопровождение проектов. Это не только снизит нагрузку на сотрудников центра, но и обеспечит большую точность обработки данных, уменьшит вероятность ошибок и упростит документооборот. Внедрение подобной платформы станет важным шагом в сторону модернизации всей системы поддержки НКО и повышения её результативности.

## **1.5 Анализ бизнес-процессов предметной области**

Для обоснования необходимости создания и внедрения специализированной онлайн-платформы необходимо провести детальный анализ текущих бизнес-процессов, отражающих взаимодействие центра с некоммерческими организациями в условиях отсутствия автоматизированной информационной системы. Такой анализ позволит выявить слабые стороны и определить узкие места в действующих процессах, а также заложить основу для предложений по их модернизации. Наиболее наглядной формой представления существующих процессов является построение модели AS-IS — схемы, отражающей текущее состояние дел без внедрения новых решений.

Модель AS-IS представляет собой инструмент визуального анализа, который позволяет зафиксировать последовательность действий и информационные потоки в существующей организационной среде. Её основное назначение — помочь понять, как на текущий момент функционирует организация, и выявить дублирующие действия, неэффективные участки и слабые звенья, подлежащие оптимизации. Такой подход также способствует формированию стратегического видения будущих преобразований.

В качестве примера приведена контекстная диаграмма бизнес-процесса «Проведение обучения» в его текущем виде (AS-IS), которая показана на рисунке А.1. Из анализа этой модели видно, что обучение сотрудников НКО осуществляется преимущественно вручную и ограничено взаимодействием с внешними участниками через бумажные или электронные формы заявок. Сертификаты о прохождении обучения доставляются либо лично, либо по почте. Важную роль играет необходимость соблюдения политики обработки персональных данных, а также наличие методических и нормативных материалов, используемых в процессе обучения. Основной канал коммуникации между слушателями и руководителем курса — личные встречи и телефонные разговоры, что существенно ограничивает охват и оперативность.

На выходе из системы формируются статистические отчеты и сертификаты, отражающие результаты прохождения обучения. В целом, данная модель демонстрирует, что существующая система «Дистанционное обучение» полностью встроена в ручной процесс, что сдерживает её развитие и масштабируемость.

Далее процесс проведения обучения детализируется на подэтапы (см. рисунок А.2), включающие:

* размещение новостей и объявлений;
* регистрацию участников;
* управление содержанием и расписанием курсов;
* роведение самих занятий;
* формирование отчётности и аналитики.

На начальной стадии сотрудники центра получают заявки от представителей НКО Забайкальского края, заинтересованных в обучении или консультативной поддержке. Эти обращения могут поступать через электронную почту или по телефону (см. рисунок А.3). Этот этап позволяет оценить ожидаемое количество слушателей и определить формат будущих занятий — групповой или индивидуальный.

После этого наступает этап подготовки и управления курсами (рисунок А.4). Руководитель курса подбирает справочные материалы, формирует черновик образовательной программы, которая затем уточняется и дополняется с учётом специфики темы и требований технического задания. После утверждения программы осуществляется формирование групп и рассылка информации о расписании и доступных форматах обучения. Особое внимание уделяется тому, чтобы сотрудники НКО, находящиеся в удалённых районах, имели возможность участвовать в обучении дистанционно.

На следующем этапе — «проведение обучения» (рисунок А.5) — участникам предоставляются материалы по электронной почте, а руководитель взаимодействует с группой или индивидуально. Этот процесс предполагает активное участие обучающихся и выполнение ими заданий, необходимых для оценки уровня усвоения материала.

Завершающий этап — подготовка отчетности (рисунок А.6). Центр отправляет слушателям сертификаты, подтверждающие прохождение обучения, и собирает подтверждающие документы от самих участников, чтобы исключить возможные нарушения или недобросовестное поведение. Проводится статистический анализ эффективности программы: учитываются результаты выполнения заданий, сравниваются итоги с прошлыми периодами, рассчитываются затраты и делаются выводы о целесообразности и качестве проводимого обучения.

Из проведённого анализа видно, что на всех стадиях бизнес-процесса приходится работать с большим объёмом документов, как бумажных, так и цифровых. Подавляющее большинство действий выполняется вручную — от обработки заявок до формирования отчетов, что требует значительных трудозатрат, увеличивает финансовые расходы и ведёт к потерям рабочего времени.

Особенно остро ощущается перегруженность сотрудников центра в периоды повышенного количества заявок из разных уголков региона. Наличие исключительно очного формата и отсутствие системы для контроля удалённого обучения затрудняет коммуникацию с организациями, находящимися на значительном расстоянии от центра. Это влечёт за собой дополнительные расходы на логистику, доставку материалов и планирование консультаций. Кроме того, неорганизованность обучающихся и невозможность мониторинга прогресса снижают общую эффективность образовательной программы.

При проведении итоговой оценки знаний сотрудникам центра часто приходится работать сверхурочно. Это увеличивает риск ошибок в проверке работ, влияет на качество консультирования и может отрицательно сказаться на дальнейшей реализации грантов обучившихся сотрудников НКО.

Таким образом, наиболее критичной проблемой текущей модели является практически полное отсутствие автоматизации всех ключевых этапов обучения и сопровождения. Это приводит к перегрузке персонала, снижению эффективности и высокому риску ошибок в работе.

Решением данной проблемы может стать внедрение целевой информационной системы, призванной автоматизировать процессы, сократить объём рутинной ручной работы и повысить прозрачность всех этапов обучения.

Ключевые цели автоматизации включают:

* снижение временных и трудозатрат на всех этапах обучения и консультирования;
* обеспечение прозрачности процесса изучения и контроля знаний;
* минимизация риска ошибок при сопровождении обучающихся и оценке их результатов;
* расширение охвата обучающихся за счёт дистанционного формата;
* повышение точности сбора и анализа статистических данных.

После внедрения такой платформы сотрудники центра смогут эффективнее организовывать работу, быстрее обрабатывать заявки, автоматически рассылать материалы, проводить оценку знаний с использованием цифровых инструментов и формировать отчётность без избыточной нагрузки. Слушатели из отдалённых районов получат полноценный доступ к обучающим материалам, консультациям и инструментам взаимодействия. Всё это позволит создать современную и устойчивую систему поддержки НКО в рамках грантовой и проектной деятельности, обеспечивая высокий уровень прозрачности и результативности.

## **1.6 Бизнес-процессы веб-приложения**

Рассмотрим подробный бизнес-процесс реализации обучения и информирования населения:

Основное назначение веб-приложения заключается в модернизации и оптимизации ключевых процессов образовательной и консультативной деятельности, осуществляемой ГКУ «Ресурсный центр развития некоммерческих организаций Забайкальского края». Создание цифровой платформы позволит существенно упростить ведение внутренней и внешней отчетности, обеспечить высокую степень надёжности хранения информации, минимизировать объём ручного труда и повысить общий уровень удобства взаимодействия для сотрудников НКО — как действующих, так и потенциальных обучающихся.

В ходе анализа существующих бизнес-процессов и их текущей реализации были выявлены значимые недостатки: высокая зависимость от бумажного документооборота, отсутствие единого хранилища данных, перегрузка сотрудников ручной работой и недостаточная оперативность информирования участников обучения. Эти проблемы не только замедляют работу центра, но и снижают качество оказываемых услуг, особенно при масштабировании программ или увеличении количества слушателей.

На основе выявленных проблем можно сделать вывод о необходимости перехода к современным цифровым решениям. Разработка специализированного веб-приложения представляется наиболее эффективным и логичным шагом. Такая платформа позволит:

* автоматизировать процесс подачи заявок и регистрации на обучение;
* централизованно хранить все данные о слушателях, курсах и результатах;
* централизованно информировать участников о предстоящих мероприятиях, изменениях в расписании и новых возможностях;
* предоставлять доступ к обучающим материалам в онлайн-режиме;
* внедрить систему проверки знаний и формирования итоговой отчётности;
* обеспечить непрерывную обратную связь между сотрудниками центра и представителями НКО, включая отдалённые населённые пункты.

## **1.7 Системный анализ**

После определения ключевой проблемы и выявления причин её возникновения, следующим логическим шагом становится углублённый анализ ранее сформулированной цели проекта. В целях структурированного подхода и повышения прозрачности планируемых действий, необходимо на основе составленного дерева целей выделить иерархически упорядоченные задачи и подзадачи, направленные на её достижение.

Для наглядного представления такой структуры эффективно использовать диаграмму дерева решений, которая служит завершающим этапом моделирования в рамках анализа целей и актуальных проблем. В отличие от дерева проблем, где фокус сосредоточен на анализе причин и следствий негативных факторов, дерево решений отображает путь к решению — оно акцентирует внимание на поставленной цели и разбивает её на логические блоки задач, ведущих к её достижению.

Диаграмма дерева решений представляет собой визуальную модель, в которой основная цель проекта выступает в качестве корневого узла. От неё отходят ветви — это ключевые направления работы, ведущие к достижению цели. Каждая из этих ветвей разветвляется на подзадачи, которые, в свою очередь, могут быть разделены на ещё более мелкие шаги. Такая иерархическая структура обеспечивает удобство анализа, расстановку приоритетов и пошаговое планирование действий.

Центральным элементом дерева решений для данного проекта выступает упрощение и автоматизация процесса обучения сотрудников некоммерческих организаций. Это и есть стратегическая цель всей инициативы (см. Рисунок 1.4). От неё исходят несколько основных направлений: оптимизация взаимодействия с пользователями, цифровизация обучающих материалов, автоматизация документооборота, а также внедрение систем контроля знаний и сбора аналитики.

Каждый «лист» дерева, то есть конечный элемент, представляет собой конкретную задачу, необходимую для реализации одного из направлений. Например, автоматизация выдачи сертификатов, создание личного кабинета пользователя, разработка базы обучающих модулей, внедрение онлайн-системы тестирования — все эти элементы составляют логическую структуру шагов на пути к конечной цели.

Анализируя такую диаграмму, можно чётко проследить, как именно система должна измениться в ходе проекта и какие действия необходимо предпринять на каждом этапе. Более того, дерево решений служит основой для составления дорожной карты проекта, планирования ресурсов и оценки рисков. Это мощный инструмент для стратегического управления, позволяющий на ранних этапах спрогнозировать возможные трудности и наметить пути их преодоления.

Таким образом, диаграмма дерева решений не только структурирует процесс достижения цели, но и повышает управляемость проекта, обеспечивая ясное понимание логики развития системы и последовательности необходимых действий.

Основные задачи, которые могут быть представлены в дереве решений:

* устранение ошибок при резервировании номеров.

Рисунок 1.4 – Дерево решений

# **ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

Проектирование занимает ключевое место в процессе создания веб-приложения, так как именно на этом этапе происходит формализация ранее поставленных целей и задач, направленных на решение выявленной основной проблемы. Этот этап служит связующим звеном между анализом предметной области и непосредственной реализацией решения, обеспечивая основу для стабильной, надёжной и функциональной разработки.

Основой грамотного проектирования является формирование функциональных требований к конечному программному продукту. Эти требования отражают, каким именно образом система должна выполнять поставленные задачи, с какими данными работать и какой результат обеспечивать пользователю. Параллельно с функциональными аспектами особое внимание уделяется требованиям к информационной безопасности, поскольку приложение будет оперировать персональными данными обучающихся, а также внутренней служебной информацией. Надёжная защита данных, соблюдение законодательства и внутреннего регламента — всё это критически важно на всех этапах жизненного цикла проекта, начиная с проектирования и заканчивая эксплуатацией.

Один из важнейших шагов в рамках проектирования — моделирование бизнес-процесса, связанного с обучением сотрудников некоммерческих организаций. Эта модель необходима для того, чтобы описать и визуализировать последовательность действий, участников процесса, потоки информации и способы их взаимодействия. Только при наличии чёткого представления о логике функционирования будущей системы можно обеспечить её устойчивость и пригодность для реального использования.

## **2.1 Построение диаграмм IDEF0, DFD и IDEF3**

Процесс моделирования по IDEF0 начинается с создания контекстной диаграммы, которая задаёт общий уровень описания системы, определяет её границы, субъект моделирования, цели и основные параметры. Контекстная диаграмма служит отправной точкой для дальнейшей декомпозиции функций на более детальные уровни. Такой подход помогает постепенно раскрыть структуру бизнес-процессов и выявить ключевые взаимосвязи, что способствует более глубокому анализу и последующему совершенствованию системы.

Проектирование информационных систем начинается с моделирования бизнес-процессов, отражающих текущую и предполагаемую (целевую) логику функционирования организации или системы. С этой целью применяются различные методы структурного анализа, среди которых особое значение имеют диаграммы IDEF0, DFD и IDEF3. Эти нотации позволяют наглядно представить процессы, потоки информации и взаимодействие между элементами системы, способствуя формализации требований и точному описанию структуры будущей информационной системы.

Методология IDEF0 (Integrated Definition for Function Modeling) предназначена для моделирования функций и процессов на высоком уровне абстракции. В этой нотации система представляется в виде набора функций, каждая из которых получает на вход определенные данные, управляется определенными условиями и производит результат, задействуя определённые механизмы (ресурсы). Главной целью использования IDEF0 является отображение структуры деятельности: какие функции выполняются, в каком порядке, и какие ресурсы и данные при этом задействуются.

Каждая функция на диаграмме IDEF0 изображается в виде прямоугольника с четырьмя сторонами. Входы поступают слева, управляющие воздействия — сверху, выходы — справа, а механизмы (ресурсы) — снизу. Такой подход позволяет строго формализовать описание процессов и построить иерархическую структуру функций — от общего контекста до детализированных подпроцессов.

На рисунке ниже представлена контекстная диаграмма процесса «Проведение обучения», отражающая целевое (TO-BE) состояние системы после внедрения онлайн-платформы

Рисунок А.7 – Контекстная диаграмма «Проведение обучения» (TO-BE)

После построения контекстной диаграммы осуществляется декомпозиция верхнего уровня на более детализированные процессы, составляющие основную функцию. Это позволяет последовательно детализировать функциональные блоки и получить более глубокое понимание логики системы.

Рисунок А.8 – Диаграмма декомпозиции контекстной диаграммы «Проведение обучения» (TO-BE)

Такая иерархическая декомпозиция охватывает ключевые процессы LMS-платформы: регистрацию пользователей, управление курсами, проведение обучения и контроль с отчетностью. Каждый из них впоследствии разбивается на более узкие задачи.

Рисунок А.9 – Диаграмма декомпозиции процесса «Регистрация пользователей» (TO-BE)

Рисунок А.10 – Диаграмма декомпозиции процесса «Управление курсами» (TO-BE)

Рисунок А.11 – Диаграмма декомпозиции процесса «Обучение пользователей» (TO-BE)

Рисунок А.12 – Диаграмма декомпозиции процесса «Контроль и отчетность» (TO-BE)

Таким образом, использование нотации IDEF0 позволяет формализовать общую архитектуру функционирования системы, что важно на этапе концептуального проектирования, когда необходимо согласовать требования всех заинтересованных сторон.

DFD (Data Flow Diagram) — это диаграммы потоков данных, используемые для моделирования логики информационных потоков в системе. Основной целью DFD является представление того, как информация перемещается внутри системы, какие процессы обрабатывают данные, какие внешние участники взаимодействуют с системой, и где хранятся данные.

Диаграммы DFD состоят из четырех ключевых компонентов:

Процессы (обычно изображаются кругами или овалами) – преобразуют входные данные в выходные.

Хранилища данных (двойная линия или открытый прямоугольник) – места постоянного хранения данных.

Потоки данных (стрелки) – указывают направление движения информации.

Внешние сущности (прямоугольники) – источники и получатели данных вне системы.

DFD-диаграммы могут строиться на различных уровнях детализации. Начинают обычно с контекстной диаграммы (уровень 0), которая отображает систему как один процесс, взаимодействующий с внешними сущностями. Затем создаются диаграммы уровня 1 и ниже, декомпозирующие основной процесс на составляющие.

Применение DFD в проекте позволяет определить, какие данные циркулируют между компонентами системы: регистрация пользователей, прохождение курсов, выполнение заданий, сохранение результатов и формирование отчетности. Такая диаграмма помогает выявить возможные «узкие места» и избыточные перемещения информации ещё до начала программной реализации.

Рисунок А.13 – Диаграмма потоков данных «Проведение обучения»

Использование DFD повышает наглядность и понятность архитектуры системы даже для специалистов без технического образования, что делает её удобным инструментом на этапах анализа требований и технического задания.

Нотация IDEF3 (Integrated DEFinition for Process Description Capture) используется для моделирования логики поведения системы во времени, а также для отображения последовательности выполнения операций. В отличие от IDEF0, где акцент сделан на структурной декомпозиции функций, IDEF3 ориентирована на динамическое поведение и позволяет описывать процессы как последовательности действий, событий и условий.

Диаграммы IDEF3 строятся в виде блоков, соединённых стрелками, отображающими логические переходы между действиями. В этой нотации используются объекты-события и объекты-действия, которые могут располагаться последовательно, параллельно или с разветвлениями. Применяются специальные логические операторы, такие как AND, OR и XOR, для построения разветвлённой структуры процессов.

Для онлайн-платформы применение IDEF3 позволяет смоделировать не только архитектуру и потоки данных, но и пользовательский сценарий прохождения курса: от регистрации и выбора курса до получения сертификата и оценки преподавателем. Также можно отразить различные сценарии поведения системы при возникновении исключений — например, при недоступности курса или истечении срока прохождения.

Рисунок А.14 – Диаграмма потоков работ «Проведение обучения»

Таким образом, метод IDEF3 предоставляет разработчику возможность описать фактическое поведение системы, то есть как процессы происходят на практике. В сочетании с диаграммами IDEF0 и DFD он обеспечивает многоуровневое, комплексное понимание как архитектуры, так и поведения LMS-платформы.

## **2.2 Построение ER-диаграммы**

При проектировании логической структуры базы данных одним из важнейших этапов является построение диаграммы «сущность–связь» (ER-диаграммы). Этот инструмент визуального моделирования используется для наглядного отображения объектов предметной области и связей между ними, что позволяет заранее спроектировать базу данных, избежав избыточности, дублирования и логических противоречий.

Суть ER-моделирования заключается в том, чтобы представить все значимые элементы предметной области в виде сущностей, то есть объектов, обладающих определенным набором характеристик, и определить между ними логические взаимосвязи. Каждая сущность отображает какой-либо важный объект, который будет храниться в базе данных. Например, в случае с онлайн-платформой такими сущностями могут быть «Пользователь», «Курс», «Тест», «Сертификат». Внутри сущностей задаются атрибуты, которые описывают свойства объектов — имя пользователя, описание курса, дата начала обучения и т.п.

Связи между сущностями обозначают, каким образом данные сущности взаимодействуют друг с другом. Эти связи характеризуются кардинальностью, то есть определяют, сколько экземпляров одной сущности может быть связано с экземплярами другой. В простейших случаях это может быть связь один-к-одному (например, между пользователем и его личным профилем), однако гораздо чаще встречаются связи типа один-ко-многим (например, один преподаватель — множество курсов) или многие-ко-многим (многие студенты — множество курсов). Для корректного отображения таких связей при построении ER-диаграммы используются условные обозначения — прямоугольники для сущностей, овалы для атрибутов и ромбы для связей, соединённые линиями.

ER-диаграмма также помогает определить ключевые поля — уникальные идентификаторы для каждой записи. Эти поля обеспечивают целостность данных и используются при построении связей между таблицами в физической модели базы данных. Кроме того, в ER-диаграммах могут отображаться слабые сущности, которые не могут существовать без связи с другой сущностью, и наследование, когда одна сущность может быть расширением другой — например, сущность «Преподаватель» может быть унаследована от сущности «Пользователь».

В контексте создаваемой онлайн-платформы, в качестве ключевых сущностей были определены: «Пользователь» (User), «Курс» (Course), «Модуль» (Module), «Задание» (Assignment), «Прохождение курса» (Enrollment), «Тест» (Test), «Сертификат» (Certificate), а также «Обратная связь» (Feedback). Каждая сущность на диаграмме сопровождается набором атрибутов, отражающих их свойства, и логическими связями, демонстрирующими, как они взаимодействуют между собой. Так, пользователь может быть связан со многими курсами через сущность «Прохождение курса», которая также хранит информацию о прогрессе, дате начала, статусе и результате обучения.

Связи и кардинальности в диаграмме позволяют определить, как данные будут храниться в реляционной базе данных. Например, связь между курсами и модулями представляет собой отношение «один-ко-многим», поскольку каждый курс может содержать несколько модулей. Аналогично, каждый модуль может содержать множество заданий или тестов, а завершение курса может привести к выдаче одного или нескольких сертификатов. Такой подход обеспечивает логическую последовательность в проектировании и последующем программировании модели данных.

На следующем рисунке представлена диаграмма «сущность–связь», построенная на основе анализа предметной области. Она отражает ключевые сущности системы, связи между ними, а также наборы атрибутов, которые будут реализованы в структуре базы данных.

Рисунок 2.2 – Диаграмма «сущность–связь»

Применение данной диаграммы в процессе проектирования информационной системы позволяет разработчику последовательно перейти от логической модели к физической реализации, минимизируя ошибки проектирования и обеспечивая структурную целостность данных. Также диаграмма служит отправной точкой для генерации миграций и моделей в фреймворке Django, который будет использоваться при разработке платформы. Таким образом, построение ER-диаграммы играет ключевую роль в формировании основ хранения информации, на которых будет базироваться функционал всей онлайн-платформы.

## **2.3 Описание проекта в нотации UML**

UML (Unified Modeling Language) — это стандартизованный язык моделирования, предназначенный для визуального описания, проектирования и документирования компонентов программных систем. Он был разработан с целью создания универсального способа представления архитектуры программного обеспечения, понятного как техническим специалистам (разработчикам, архитекторам), так и заказчикам. UML не является языком программирования, но он позволяет строить модели, которые отражают структуру и поведение системы ещё до начала её непосредственной реализации.

Одним из главных преимуществ UML является то, что он предоставляет набор диаграмм, каждая из которых отвечает за определённый аспект системы. Например, диаграммы классов описывают структуру и связи между сущностями; диаграммы прецедентов отражают взаимодействие пользователей с системой; диаграммы последовательностей показывают порядок обмена сообщениями между объектами во времени. Это позволяет не только визуализировать проект, но и выявить потенциальные логические ошибки на раннем этапе.

Использование UML особенно важно при разработке сложных проектов, таких как образовательные онлайн-платформы, где существует множество компонентов, пользовательских ролей и сценариев взаимодействия. Использование диаграмм UML помогает формализовать требования заказчика, лучше понять бизнес-логику приложения и структурировать архитектуру проекта до начала программирования. Это снижает риски, связанные с недопониманием, и упрощает коммуникацию между членами команды.

Таким образом, UML играет ключевую роль в процессе анализа и проектирования программного обеспечения. Он позволяет обеспечить согласованность между требованиями, архитектурой и реализацией, а также служит удобным инструментом для документирования проекта. В контексте дипломной работы по созданию веб-приложения, UML способствует системному подходу к разработке и позволяет обеспечить качественную основу для дальнейшей реализации платформы.

Диаграмма прецедентов — это один из наиболее часто используемых инструментов UML, предназначенный для моделирования взаимодействия пользователей (актеров) с системой. Основная цель этой диаграммы — представить поведение системы с точки зрения конечного пользователя, выделить функциональные возможности, которые система должна обеспечивать.

Диаграмма прецедентов состоит из следующих основных элементов:

* актеры (Actors) — внешние участники, которые взаимодействуют с системой. Это могут быть как реальные пользователи (например, студент, преподаватель, администратор), так и другие системы;
* прецеденты (Use Cases) — действия или функции, которые может выполнять система в ответ на запросы акторов (например, «Регистрация», «Прохождение курса», «Оценка знаний»);
* связи (Relationships) — линии, связывающие акторов и прецеденты. Они могут быть простыми ассоциациями или расширенными отношениями типа include (включение) и extend (расширение).

Систему на диаграмме прецедентов, как правило, ограничивают рамкой (system boundary), которая показывает границы системы.

Построение диаграммы прецедентов начинается с идентификации всех типов пользователей системы и определения того, какие функции каждый из них должен иметь доступ. Например, студент может иметь доступ к регистрации, прохождению курсов и просмотру оценок, в то время как администратор может управлять пользователями и создавать курсы.

Рисунок Б.1 – Диаграмма прецедентов

Диаграмма классов используется для моделирования статической структуры системы. Она показывает, из каких классов состоит система, какие у них свойства (атрибуты) и методы (операции), а также каким образом эти классы связаны между собой. Это одна из наиболее важных диаграмм в объектно-ориентированном проектировании.

Основные элементы диаграммы классов:

* класс — представлен прямоугольником, разделённым на три части: имя класса, список атрибутов и список методов;
* атрибуты — характеристики, присущие классу (например, у пользователя могут быть имя, логин, email);
* методы — действия, которые может выполнять класс (например, зарегистрироваться, пройти курс);
* ассоциации — связи между классами. Они могут быть однонаправленными или двунаправленными;
* множественности — указывают количество объектов, которое может участвовать в ассоциации (например, один курс может иметь много студентов);
* наследование (generalization) — отображается стрелкой с треугольником и показывает, что один класс наследуется от другого.

При построении диаграммы классов для онлайн-платформы важно выделить основные сущности системы: пользователь, курс, модуль, тест, сертификат, сообщение и пр. Между этими сущностями устанавливаются отношения — например, один пользователь может записаться на несколько курсов, каждый курс состоит из модулей, а каждый модуль может содержать тесты.

Рисунок Б.2 – Диаграмма классов

Диаграмма последовательности отображает взаимодействие между объектами системы во времени. Она используется для моделирования конкретных сценариев использования и показывает, какие сообщения передаются между объектами и в каком порядке. Эта диаграмма относится к типу поведенческих и особенно полезна на этапе проектирования логики взаимодействий.

Основные компоненты диаграммы последовательности:

* объекты (lifelines) — вертикальные пунктирные линии, представляющие участники взаимодействия (например, пользователь, веб-сервер, база данных);
* сообщения (messages) — горизонтальные стрелки, указывающие, какие сообщения передаются между объектами;
* временная ось — идёт сверху вниз, демонстрируя последовательность вызовов;
* активности (activation boxes) — прямоугольники на линиях жизни, показывающие, когда объект активен.

Диаграммы последовательности в проекте используются для иллюстрации ключевых сценариев, например, таких как процесс регистрации пользователя, прохождение тестирования, получение сертификата. Благодаря такой диаграмме можно наглядно увидеть, какие компоненты системы взаимодействуют в каждом процессе и в какой последовательности происходят вызовы.

Рисунок Б.3 – Диаграмма последовательности

## **2.4 Планирование**

Планирование является одним из наиболее значимых этапов при создании любого программного продукта, в том числе онлайн-платформы для обучения и информирования. На этой стадии определяется общий вектор проекта: формулируются цели, этапы разработки, временные рамки и ресурсы. Без чёткого плана реализация проекта рискует столкнуться с несогласованностью действий команды, затягиванием сроков, несоответствием результата ожиданиям заказчика и даже техническими ошибками из-за неправильной архитектуры или недооценки сложности задач.

Для успешной разработки важно не только определить, что именно должно быть реализовано, но и в какой последовательности, и в какие сроки. Планирование позволяет заранее учесть все ключевые компоненты: от разработки пользовательского интерфейса и подключения базы данных до написания логики, тестирования, публикации и поддержки проекта. Кроме того, оно способствует равномерному распределению нагрузки между участниками команды и помогает избежать ситуаций, когда часть задач оказывается не выполненной в последний момент.

В рамках планирования проекта по созданию образовательной платформы была использована диаграмма Ганта — инструмент, позволяющий наглядно представить этапы разработки, продолжительность каждого этапа и зависимости между задачами. Диаграмма Ганта — это тип календарного графика, где задачи отображаются в виде горизонтальных полос на временной шкале. Такая визуализация делает процесс контроля за выполнением плана максимально прозрачным и удобным для всех участников проекта.

Выбор диаграммы Ганта обусловлен её наглядностью, простотой понимания и эффективностью при управлении проектами с линейной или итеративной структурой. Она позволяет легко адаптировать план в случае изменения требований или сроков, выявлять критические точки и определять возможные риски на ранней стадии.

Кроме того, именно на стадии проектирования стали очевидны участки, где есть риск срыва сроков или технические сложности. Для визуализации хода проекта была использована диаграмма Ганта — инструмент, позволяющий представить каждую задачу в виде горизонтальной полосы на временной шкале. По оси X откладывается календарное время (дни, недели), по оси Y — список задач.

Структурно проект был разделён на четыре крупных блока: подготовка окружения, backend-часть (база данных, логика Django-приложений), frontend (интерфейс, стили, подключение библиотек) и тестирование. Уже на этом этапе было видно, что, например, реализацию раздела с викторинами стоит начинать только после настройки модели пользователя и базы. Также диаграмма позволила учесть время на исправление ошибок и отладку — без этого любой график будет слишком оптимистичным

В контексте проекта диаграмма Ганта сыграла важную роль в организации работы и позволила грамотно спланировать реализацию всех ключевых функциональных компонентов платформы в рамках установленного графика.

# **ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОЕКТА**

## **3.1 Выбор инструментов для разработки**

### ***3.1.1 Язык программирования***

В качестве основного языка программирования для реализации веб-приложения был выбран Python. Это решение было принято не только на основании анализа технических требований проекта, но и исходя из указания со стороны заказчика, который обозначил Python как предпочтительный язык для реализации серверной логики платформы. Такой выбор обусловлен как популярностью языка в академической и промышленной среде, так и его широкими возможностями для быстрой и надёжной веб-разработки.

Python зарекомендовал себя как язык, сочетающий лаконичность синтаксиса с высокой читаемостью кода, что особенно важно при реализации как командных, так и одиночных проектов. Python показался особенно удобным на этапе активной разработки — благодаря понятному синтаксису и логичной структуре легко сосредоточиться на логике самого проекта, а не на деталях языка.

Естественно, даже если основной проект выбран, это не означает, что не нужно рассматривать другие языки как вспомогательные или как альтернативу. PHP — язык, с которого традиционно начиналась веб-разработка. Он до сих пор используется в огромном количестве проектов, особенно в сочетании с CMS. Но в случае создания чего-то более гибкого, вроде образовательной платформы с регистрацией, обсуждениями и тестами, PHP уже не даёт такого комфорта в проектировании архитектуры, как Python с его фреймворками.

Java тоже рассматривалась. У неё есть плюсы: производительность, строгость, масштабируемость. Но минус в том, что она требует больше времени на реализацию даже базовых вещей. Для проекта, ограниченного сроками и предполагающего итеративную разработку, это не самый удачный вариант. Ruby с Rails тоже звучал интересно, но его экосистема в последние годы заметно ослабла, особенно в русскоязычном пространстве. В итоге выбор Python оказался вполне логичным.

Таким образом, выбор Python позволяет быстро реализовать надёжную, масштабируемую и понятную архитектуру образовательной платформы, соответствующую ожиданиям заказчика и требованиям дипломного проекта. Для задач образовательной онлайн-платформы он оказался особенно удачным вариантом: у Python богатый набор инструментов — такие библиотеки и фреймворки, как Django, Flask, Jinja2, SQLAlchemy позволяют покрыть практически все нужды проекта без необходимости подключать дополнительные технологии.

### ***3.1.2 Веб-фреймворк***

В качестве основного инструмента серверной части веб-приложения был выбран фреймворк Django, один из самых популярных и зрелых веб-фреймворков в экосистеме Python. Django — это высокоуровневый фреймворк с широким набором встроенных инструментов, реализующий архитектурную модель MVT (Model–View–Template), которая позволяет чётко структурировать проект и ускорить процесс разработки за счёт автоматизации многих рутинных задач.

Одной из главных причин выбора Django является его философия «батарейки включены» (англ. batteries included), которая предполагает, что фреймворк предоставляет готовые решения для большинства типичных задач веб-разработки: от маршрутизации и работы с базой данных до системы авторизации, админки и обработки форм. Это значительно сокращает время разработки и упрощает поддержку приложения, особенно в условиях ограниченного времени на реализацию дипломного проекта.

При выборе фреймворка рассматривались и другие популярные решения: Flask, FastAPI и Tornado. Flask — лёгкий микрофреймворк, известный своей гибкостью и минималистичным подходом. Он позволяет «с нуля» построить только те компоненты, которые действительно нужны, не навязывая структуру проекта. Однако для создания полноценного образовательного веб-приложения с системой регистрации, профилями пользователей, загрузкой и отображением контента, требуется интеграция множества сторонних библиотек, что усложняет архитектуру и увеличивает технический долг. Для сложного и масштабируемого проекта такой подход может оказаться неэффективным.

FastAPI — современный асинхронный фреймворк, получивший признание за высокую производительность и поддержку спецификации OpenAPI. Он отлично подходит для создания REST API и микросервисной архитектуры. Однако его основной фокус — это API, а не традиционные веб-приложения с шаблонами и административными интерфейсами. Разработка полноценного пользовательского интерфейса в рамках FastAPI требует дополнительных инструментов и интеграции с другими библиотеками, что делает его менее удобным для задач, ориентированных на конечного пользователя через браузер.

Tornado — ещё один мощный асинхронный фреймворк, подходящий для высоконагруженных систем, особенно когда требуется постоянное соединение (например, WebSocket). Однако он значительно сложнее в освоении и требует более глубокого понимания неблокирующего программирования. Для образовательной платформы, основное назначение которой — отображение статичных и динамических страниц, личных кабинетов, материалов и статистики — Tornado избыточен.

В отличие от указанных фреймворков, Django предоставляет встроенный административный интерфейс, который автоматически генерируется на основе моделей. Это даёт возможность администраторам платформы без программирования управлять курсами, пользователями, статистикой и контентом. Также Django предлагает систему аутентификации «из коробки», защищённую от типичных уязвимостей, таких как CSRF, XSS и SQL-инъекции, что критически важно при работе с персональными данными пользователей.

Дополнительным плюсом Django является его масштабируемость и широкое сообщество. В случае возникновения трудностей легко найти готовые решения или советы от других разработчиков. Подробная документация, огромное количество плагинов и готовых библиотек делают фреймворк идеальным инструментом для учебных и коммерческих проектов.

Таким образом, несмотря на наличие интересных альтернатив, Django был выбран как наиболее подходящий фреймворк для реализации онлайн-платформы обучения и информирования, так как он обеспечивает быстрый старт, безопасную и модульную архитектуру, встроенные инструменты управления и масштабируемость. Эти качества делают его оптимальным решением для дипломного проекта с акцентом на функциональность, надёжность и структурированность

### ***3.1.3 СУБД***

Основной системой управления базами данных для онлайн-платформы была выбрана объектно-реляционная СУБД с открытым исходным кодом —PostgreSQL. Её использование обусловлено как техническими преимуществами, так и хорошей интеграцией с Django. PostgreSQL предоставляет высокую надёжность, масштабируемость и гибкость при работе с различными типами данных.

Одним из ключевых преимуществ PostgreSQL является её соответствие стандартам SQL и поддержка сложных типов данных, включая массивы, JSON, UUID и даже географические координаты (через PostGIS). Это делает её особенно удобной для проектов с разнообразной структурой данных. В рамках данной платформы, которая хранит не только учебные материалы, но и статистику, тесты, профили пользователей и другой разнородный контент, подобные возможности оказываются крайне полезными.

При выборе СУБД были рассмотрены также альтернативные решения, в том числе MySQL, SQLite и MongoDB. MySQL является одной из самых популярных СУБД и обладает высокой производительностью при простых запросах, однако в плане расширенных возможностей, таких как работа с JSON или транзакциями, уступает PostgreSQL. Кроме того, MySQL по умолчанию менее строг в отношении согласованности данных, что может привести к неочевидным ошибкам.

SQLite, в свою очередь, представляет собой встраиваемую СУБД, работающую без отдельного сервера. Её главное преимущество — простота использования, однако она не подходит для многопользовательских веб-приложений, поскольку не обеспечивает достаточного уровня масштабируемости и параллелизма.

В свою очередь MongoDB представляет собой нереляционную базу данных, работающую с документами формата BSON, что делает её удобной для гибкой схемы данных. Однако в рамках текущего проекта, где структура и связи между объектами строго определены, реляционный подход оказался предпочтительнее.

PostgreSQL отличается высокой производительностью и способностью обрабатывать большие объёмы данных с минимальными задержками. Её механизм транзакций обеспечивает целостность данных, а система индексов — быструю выборку информации. Всё это позволяет построить надёжную и масштабируемую систему хранения данных. Кроме того, встроенная поддержка расширений (например, pg\_stat\_statements, pg\_trgm) позволяет гибко адаптировать базу данных под задачи проекта и оптимизировать производительность.

Подключение PostgreSQL к Django осуществляется через драйвер psycopg2, который обеспечивает полную поддержку ORM Django. Благодаря этому можно работать с базой данных на уровне Python-объектов, не прибегая к написанию «ручных» SQL-запросов, что ускоряет разработку и снижает вероятность ошибок.

По итогу, несмотря на наличие других популярных вариантов СУБД, PostgreSQL был всё же рассмотрен как оптимальный вариант. Подобное решение обусловлено особенностями системы. А именно такими качественными характеристиками как: надёжности, функциональности, совместимости с Django и возможностью масштабирования. Таким образом, СУБД позволит обеспечить стабильную работу онлайн-платформы, эффективную обработку запросов и сохранность пользовательский данные, в частности персональные данные. Соответственно, среди рассмотренных СУБД PostgreSQL определён оптимальной СУБД для реализации образовательной онлайн-платформы.

## **3.2 UX-разработка**

Даже если платформа функционально мощная, но в ней сложно разобраться — ей не будут пользоваться. Поэтому на этом этапе вместе с заказчиком обсуждались сценарии использования: кто заходит на сайт, с какой целью, как быстро он должен находить нужное, что для него приоритет, а что можно спрятать «вглубь». Всё это — часть UX-разработки (от англ. user experience — «пользовательский опыт»). Её задача — не просто сделать красиво, а спроектировать такой интерфейс, в котором удобно работать, ничего не раздражает и всё интуитивно понятно.

Чтобы не рисовать интерфейсы на бумаге и не делать «вслепую» вёрстку, использовался инструмент под названием Figma. Это онлайн-сервис для создания макетов и прототипов. Он оказался удобным по нескольким причинам: паботает прямо в браузере — не нужно ставить ничего на компьютер, всё доступно из любой точки; позволяет работать командой: дизайнер, разработчик и заказчик могут смотреть и комментировать макеты в реальном времени; можно делать кликабельные прототипы — пользователь может «походить» по интерфейсу ещё до написания кода; макеты легко переводятся в вёрстку — разработчику не нужно догадываться, где какой шрифт, цвет или отступ;

Макеты созданные с помощью системы слоёв и графичекских инструментов Figma прошли два этапа согласования: отсев неудачных вариантов и добавление право в нескольких вариациях Рисунок 3. Благодаря этому, ещё до начала программирования понятно, как платформа будет выглядеть и работать.

## **3.3 Описание общей структуры проекта**

Когда начинаешь работать с Django, одной из первых вещей, которую нужно понять — это архитектура MVT, на которой всё здесь строится. На словах всё просто: модель — это про данные, представление — про логику, шаблон — про внешний вид. Такой подход помогает не путаться в коде и точно знать, за что отвечает каждый кусок проекта. Модель — это, грубо говоря, описание таблиц базы данных на языке Python. В ней задаются поля, связи, ограничения — всё, что касается структуры информации. Файл представления или же view.py отвечает за связь запросив между пользователем и моделью. Можно сказать, суть этого файла в обработке данных, а также их ввод и вывод в разрабатываемую информационную систему. Представления — это функции или классы, обрабатывающие HTTP-запросы и возвращающие ответы. Они содержатся в файле views.py и отвечают за бизнес-логику проекта. Представления взаимодействуют с моделями, извлекают или сохраняют данные, и передают их в шаблоны для отображения пользователю.

Шаблоны располагаются в директории templates и представляют собой HTML-файлы с поддержкой встроенного шаблонного языка Django. Они используются для формирования визуальной части страницы, выводя данные, полученные из представлений. Это даёт возможность эффективно отделить код от дизайна. В общем, шаблон — это HTML с добавлением специальных тегов, который формирует финальную страницу для пользователя. Что особенно удобно — эти части чётко разделены. Модель ничего не знает о том, как выглядит сайт, шаблон не знает, откуда берутся данные, а представление просто связывает их между собой. трансформируются в таблицы базы данных с помощью встроенного ORM. Модели позволяют абстрагироваться от SQL и работать с данными через Python-объекты, что значительно упрощает взаимодействие с базой. Благодаря этому разработку можно вести параллельно: один человек пишет логику, другой — шаблоны, третий — модели. Исправить что-то потом тоже проще: поменял внешний вид — и не задел логику, обновил модель — и не тронул шаблон. Для онлайн-платформы это особенно важно. Здесь много разных разделов — личные кабинеты, тесты, обсуждения, посещаемость. И каждый из них удобно разложить по этим трём слоям. Это делает проект не только понятным, но и гибким — можно спокойно развивать его дальше, не боясь, что всё сломается от одного неосторожного изменения. Модели представляют собой классы, описывающие структуру таблиц базы данных. Они создаются в файле models.py каждого приложения и автоматически

Проект состоит из основного конфигурационного модуля и одного или нескольких приложений. Конфигурационный модуль управляет общими настройками, такими как базы данных, пути к статическим и медиафайлам, список подключённых приложений и другие параметры. Каждое приложение в рамках проекта выполняет отдельную задачу, например: управление пользователями, отображение курсов, тестирование и т.д.

Такая структура делает проект легко масштабируемым и удобным для разработки. Новые разработчики смогут быстро ориентироваться в проекте, поскольку архитектура предсказуема и следует общепринятым стандартам Django

## **3.4 Настройки проекта**

Веб-приложение, разрабатываемое в рамках дипломного проекта, построено с использованием фреймворка Django. Одним из важнейших элементов конфигурации проекта является файл settings.py, в котором задаются глобальные параметры, влияющие на поведение всего веб-приложения. Его корректная настройка определяет безопасность, производительность, локализацию, структуру шаблонов, поведение базы данных и множество других аспектов.

Файл settings.py формируется автоматически при создании проекта, однако требует последующей настройки в зависимости от задач приложения. Одним из первых элементов в данном файле является переменная BASE\_DIR, задающая путь к корневой директории проекта. На её основе формируются пути ко всем остальным директориям (шаблоны, статика, медиа и т. д.), что делает структуру проекта гибкой и независимой от конкретного физического расположения на диске.

Безопасность проекта обеспечивается за счёт переменной SECRET\_KEY, которая используется для криптографических операций — в частности, для генерации токенов, хэшей паролей и подписи сессий. В публичных репозиториях и продуктивной среде этот ключ должен храниться отдельно от исходного кода, в переменных окружения или .env-файле. Параметр DEBUG, установленный в True, активирует режим отладки и используется только в процессе локальной разработки. Для публикации на хостинге он обязательно должен быть отключён (False) во избежание утечки внутренней информации. Список ALLOWED\_HOSTS в разработке пока допускает соединение с любого хоста (значение '\*'), однако в продуктивной среде он должен быть строго ограничен адресами доверенных доменов.

Секция INSTALLED\_APPS определяет список подключённых приложений и библиотек. Помимо стандартных модулей Django, таких как admin, auth, sessions и messages, в проект добавлены собственные приложения main, discussion, attendance и quiz, реализующие основные блоки образовательной платформы. Также присутствуют сторонние пакеты, включая froala\_editor — визуальный редактор для ввода контента — и django\_cleanup, который автоматически удаляет неиспользуемые медиафайлы при обновлении модели.

Раздел MIDDLEWARE содержит список промежуточных компонентов, которые обрабатывают запросы и ответы между сервером и пользовательским браузером. Эти модули отвечают за безопасность, обработку сессий, защиту от CSRF-атак, аутентификацию, отображение сообщений и защиту от атак через iframe.

Особое внимание уделяется настройке шаблонов (TEMPLATES), где указывается путь к директории с HTML-шаблонами (BASE\_DIR / 'templates') и список процессоров контекста, которые передают в шаблоны служебную информацию, например, текущего пользователя или параметры запроса.

Сервер взаимодействует с базой данных, конфигурируемой в секции DATABASES. Хотя в предварительном анализе была выбрана PostgreSQL, на момент начальной настройки проекта используется MySQL в качестве временной альтернативы. Параметры подключения включают название базы (elms), хост (localhost), пользователя (root) и пароль (пока не задан). Позднее, при переносе на PostgreSQL, структура DATABASES легко адаптируется под новый движок без необходимости серьёзной переработки кода.

Система валидации паролей реализована через список AUTH\_PASSWORD\_VALIDATORS, включающий проверки на длину, простоту и схожесть с личными данными. Это соответствует требованиям безопасности современных веб-приложений. В свою очередь, список PASSWORD\_HASHERS задаёт используемые алгоритмы хеширования паролей. Здесь предусмотрены как современные (PBKDF2, Argon2, BCrypt), так и устаревшие алгоритмы, которые обеспечивают совместимость, но должны использоваться с осторожностью.

Механизмы локализации реализуются через переменные LANGUAGE\_CODE и TIME\_ZONE. В текущей конфигурации язык установлен как русский (ru), а часовой пояс — Europe/Moscow.

Для настройки работы с сессиями указано использование SESSION\_ENGINE = 'django.contrib.sessions.backends.signed\_cookies', что позволит хранение данных сессий в cookies, подписанных с помощью SECRET\_KEY. Таким образом, осуществляется сразу упрощение разворачивания проекта как системного процесса на сервере, ещё и без необходимости отдельной БД для хранения сессий, а также снижение ресурсных потребностей.

В целом, создание параметров для настроек Django-проекта через файл settings.py упрощает для заказчика масштабирование онлайн-платформы после внедрения и отражает подход к организации Django-проекта. settings.py — это как панель управления всем Django-проектом. В нём заранее настраиваются важные вещи: какая база данных используется, где лежат статические файлы, какие модули подключены, какие параметры безопасности действуют и так далее. Такой подход особенно удобен, когда проект нужно будет развивать дальше. Всё уже структурировано, не нужно «копаться» в коде — достаточно поменять нужные параметры в одном месте.

## **3.5 Авторизация и регистрация**

Проект построен на архитектуре

## **3.6 Заявки на курсы**

Проект построен на архитектуре

## **3.7 Новости, курсы, просмотр их содержимого**

Проект построен на архитектуре

## **3.8 Посещаемость и тесты**

Проект построен на архитектуре

## **3.9 Обсуждение внутри курса**

Проект построен на архитектуре

## **3.11 Статистика**

Проект построен на архитектуре

## **3.12 Создание бота для уведомлений**

Проект построен на архитектуре

## **3.13 Описание интерфейса веб-приложения**

В качестве основы использовался HTML — язык разметки, на котором строится вся структура страницы. Он отвечает за то, какие элементы на странице вообще есть: заголовки, параграфы, формы, таблицы, блоки с материалами, комментарии, кнопки и так далее. HTML — это как скелет веб-страницы. А за «одежду» отвечает CSS. Именно с его помощью интерфейс становится красивым, элементы — выравниваются, а шрифты — читаемыми. Использовались возможности CSS3: медиа-запросы для адаптивности, flexbox для гибкой компоновки и базовые анимации.

Чтобы не изобретать велосипед, для вёрстки применялся Bootstrap — фреймворк, в котором уже всё продумано: сетки, кнопки, выпадающие меню, вкладки, формы. Особенно в рамках учебного проекта это оказалось крайне полезно — мы не тратили время на проработку типовых элементов, а сразу сосредоточились на логике платформы.

Интерактивность страниц — то, что делает платформу «живой». Для её реализации использовалась библиотека jQuery. Она упрощает работу с HTML-документом: позволяет, например, скрыть или показать блок без перезагрузки страницы, проверять правильность заполнения форм прямо на лету, а также обрабатывать пользовательские действия — клики, нажатия клавиш, отправку форм. Благодаря этому платформа стала более отзывчивой и удобной: пользователь моментально получает обратную связь, не дожидаясь полной перезагрузки страницы.

Для визуализации данных применялась библиотека Chart.js, с помощью которой прямо в браузере отображаются диаграммы и графики. Это особенно важно для разделов статистики и анализа прогресса. Простой список чисел мало о чём говорит, а вот график успеваемости или диаграмма посещений помогает быстро понять, как идут дела. Это делает статистику не абстрактной, а наглядной.

И, наконец, чтобы добавить немного «жизни» в интерфейс, применялась Animate.css — библиотека с готовыми CSS-анимациями. Благодаря ей страницы стали отзывчивыми: элементы плавно появляются, кнопки реагируют на наведение, формы — на ошибки. Это вроде бы мелочи, но именно из таких деталей складывается ощущение, что перед тобой не скучный сайт, а полноценная современная платформа, где учиться — приятно.

## **3.14 Публикация проекта на веб-хостинге GitHub**

Проект был размещён на GitHub — это не просто хранилище кода, а полноценная платформа для командной работы и контроля версий. Такой подход сразу задал высокий стандарт организации разработки: каждый шаг, каждая правка, каждая новая идея фиксировались в истории проекта. Это удобно и в плане прозрачности, и в плане безопасности: всегда можно вернуться к предыдущей версии, посмотреть, кто и что изменил, и почему. Началось всё с инициализации локального Git-репозитория. Сразу после формирования базовой структуры проекта — каталогов, модулей, конфигурационных файлов — был сделан первый коммит. После этого репозиторий связали с GitHub, что позволило вести разработку с сохранением всех версий и синхронизацией между локальной и удалённой копией. Git стал своего рода «черновиком» и «архивом» в одном лице. Во время работы особое внимание уделялось содержательности коммитов. Вместо абстрактных "fixed bug" или "update" использовались осмысленные сообщения: "добавлена обработка ошибок в форме регистрации", "рефакторинг модуля аутентификации", "доработан интерфейс страницы курсов". Такой подход помогает быстрее ориентироваться в истории изменений и экономит время при отладке, особенно на этапе тестирования и доработки. В репозитории также был размещён файл README.md.

Таким образом, использование Git и GitHub сделало процесс разработки прозрачным и организованным. Платформа стала удобным инструментом не только для хранения кода, но и для ведения документации, отслеживания ошибок и публикации итоговой версии проекта в открытый доступ.

# **ГЛАВА 4. ТЕСТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ**

Перед тем как запустить веб-приложение на Django, нужно подготовить среду. В первую очередь — установить все зависимости. Чтобы не делать это вручную, используется файл requirements.txt, где заранее прописаны все нужные библиотеки. Установка проходит через pip — стандартный пакетный менеджер Python. Он позволяет быстро поднять окружение под проект: подтянуть нужные версии пакетов, исключить несовместимости и при этом избежать конфликтов с глобальными библиотеками.

Обычно всё начинается с создания виртуального окружения. Это такой «песочница»-режим, в котором изолируются все зависимости конкретного проекта. Так можно не бояться, что какие-то системные библиотеки повлияют на работу платформы. Потом — команда pip install -r requirements.txt, и всё нужное устанавливается за пару минут. Это удобно и при разработке, и при переносе на другой сервер — достаточно склонировать репозиторий и запустить установку.

Когда всё было установлено и проект без проблем запускался в локальной среде, пришло время убедиться, что он работает не только «в теории». А значит — тестировать. Сначала проверялись отдельные фрагменты кода — модули, функции, классы, отвечающие за регистрацию, авторизацию, отображение курсов и прочее. Это так называемое модульное тестирование. Оно нужно, чтобы понять: каждая деталь системы делает то, что от неё требуется, и делает это корректно.

Но мало, чтобы всё работало по отдельности — важнее, чтобы элементы не конфликтовали между собой. Поэтому следующим этапом было интеграционное тестирование. Здесь уже смотрели на связки: как отрабатывают формы, как передаются данные, правильно ли система реагирует на входные параметры и что происходит при необычных действиях пользователя.

Когда основные сценарии отлажены, начинается то, что ближе к реальности — функциональное тестирование. Это когда открываешь сайт и пользуешься им так, как это сделал бы обычный человек. Прошли путь от регистрации до просмотра курсов, пробовали отправлять формы, переключались между разделами. Заодно проверяли, удобно ли всё выглядит, не съезжает ли верстка на телефоне, и не «прыгает» ли кнопка при наведении. Чтобы понять, как платформа поведёт себя в реальной жизни, дали её протестировать сотрудникам «Центра по работе с НКО» — именно на них проект и был ориентирован

По результатам тестирования .

Таким образом, .

# **ГЛАВА 5. АНАЛИЗ ЭФФЕКТИВНОСТИ**

В крае всего 1 023 некоммерческих организации, при этом сотрудники каждой из них могут пройти бесплатное обучение. Помимо этого, образовательные услуги могут получить самозанятые, частные лица, индивидуальные предприниматели и общества с ограниченной ответственностью. За 2024 год в Центре развития НКО Забайкальского края было обучено 1232 человека

На каждого обучающегося выдается анкета (3 листа формата А4, цветная печать) и договор на обучение (4 листа формата А4, ч/б печать). Средняя стоимость печати – 5 рублей за лист. Таким образом, затраты на печать базовой документации составили: 35 рублей на одного обучающегося и 43 120 рублей на всех обучающихся. В случае интеграции платформы потребность в заполнении бумажной анкеты отпадает и печать составит: 20 рублей на одного человека и за 2024 год могло бы быть затрачено 24 640 рублей.

Помимо этого, будет осуществлена экономия времени, так как не будет необходимости персонально рассылать и раздавать учебные материалы, поскольку платформа позволит осуществлять этот процесс централизованно.

Так же в таблице 1 были рассмотрены уже готовые решения в виде популярных крупных LMS-платформ, разрешенных для использования в бюджетных учреждениях ДПО.

Таблица 1

| LMS-платформа | Бесплатный пробный период | Ограничения и тарифы | Основные особенности | Ссылка |
| --- | --- | --- | --- | --- |
| iSpring Learn | - | До 10 пользователей, индивидуальный подбор тарифа | Безлимитное хранилище файлов, поддержка SCORM, предоставление статистики | ispring.ru |
| TeachBase | 14 дней | Тарифы до 10 пользователей | Тесты, интеграции с Tilda, Google Docs, предоставление статистики | teachbase.ru |
| Skill Cup | Нет стандартной пробной версии | Тарифы 15 000 ₽/мес на 50 пользователей | Предоставление статистики, наличие мобильной версии | skillcup.ru |
| Unicraft | Нет бесплатного периода | Тарифы от 10 пользователей | Простая настройка | unicraft.org |
| Эквио | Демоверсия на 12 пользователей | Индивидуальный подбор тарифа | Предоставление статистики, наличие мобильной версии | e-queo.com |

Однако все эти решения лишены гибкости в функционале и в случае необходимости не получится произвести нововведение. Помимо этого, одной из немаловажных проблем является зависимость бюджетных учреждений в закупках через систему ЕСИ в случае официального приобретения лицензии на использование готовых решений. Таким образом появляется проблема обоснования потенциальному поставщику необходимости выставления низкого ценника и отстаивании предложенного коммерческого предложения для включения в план ФХД необходимой для расходов суммы. Ранее Центре развития НКО Забайкальского края уже пробовал запросить коммерческие предложения, но все предложенные варианты оказались не рентабельными, поскольку сумма затрат на приобретение годовой лицензии варьировалась в районе двух с половиной миллионов рублей, при этом в случае закупки лицензии на упрощенную ИС сумма закупки на сайте ЕИС варьируется в пределах трех ста тысяч рублей [9].

# **ЗАКЛЮЧЕНИЕ**