

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка социальных и экономических информационных систем

(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка веб-приложения для выгрузки и просмотра учебных материалов»

Обучающийся

Д.А. Якимов

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н., доцент, О.В. Аникина

ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

И.Ю. Усатова

(ученая степень, звание, И.О. Фамилия)

Тольятти 2024

Аннотация

Тема бакалаврской работы – «Разработка веб-приложения для работы с учебными материалами онлайн-школы».

Сейчас дополнительное онлайн-образование является крайне крупной индустрией с большими возможностями заработка. Однако большая распространённость разнообразных школ, курсов, менторов порождает огромную конкурентность на рынке. Любой подобной организации необходимо совершенствовать все свои аспекты день ото дня для продолжения эффективной деятельности. Одним из таких аспектов является удобство взаимодействия с учебными материалами.

Объектом исследования бакалаврской работы является разработка современного веб-приложения для упрощения взаимодействия учащихся с учебными материалами. Предметом исследования бакалаврской работы является проект по реализации веб-приложения для взаимодействия обучающихся с учебными материалами.

Цель бакалаврской работы – проектирование и разработка веб-приложения, позволяющего с должным уровнем качества и удобства взаимодействовать с учебными материалами.

Актуальность работы подчёркивается необходимостью внедрения более качественных технологий для повышения качества продукта и его конкурентоспособности.

Практическая значимость бакалаврской работы заключается в разработке веб-приложения, увеличивающего качество конечного продукта.

Данная работа состоит из введения, трёх разделов, заключения и списка литературы.

Бакалаврская работа включает 46 страниц текста, 20 рисунков, 5 таблиц и 20 источников.

Abstract

The topic of the bachelor's thesis is "Development of a web application for working with online school teaching materials".

In the current reality, additional online education is an extremely large industry with great earning opportunities. However, the high prevalence of various schools, courses and mentors generates huge competition in the market. Any such organization needs to improve all its aspects day by day in order to continue its effective activities. One of these aspects is the convenience of interacting with educational materials.

The object of the bachelor's research is the development of a modern web application to simplify the interaction of students with educational materials.

The subject of the research of the bachelor's work is a project on the implementation of a web application for the interaction of students with educational materials.

The purpose of the bachelor's work is to design and develop a web application that allows you to interact with educational materials with an appropriate level of quality and convenience.

The relevance of the work is emphasized by the need to introduce better technologies to improve the quality of the product and its competitiveness.

The practical significance of the bachelor's work lies in the development of a web application that multiplies the quality of the client's final product.

Содержание

Введение.....	5
1 Проектное моделирование и описание требований	7
1.1 Характеристика ООО «Изикод»	7
1.2 Выбор методологии концептуального моделирования	10
1.3 Разработка и анализ модели бизнес-процесса «Как есть».....	11
1.4 Разработка требований к ПО	17
2 Проектирование программного обеспечения. Выбор программных средств разработки.....	20
2.1 Описание инструментов для разработки программного обеспечения	20
2.2 Проектирование статической структуры системы.....	22
2.3 Выбор СУБД.....	32
3 Реализация веб-приложения для выгрузки и просмотра учебных материалов	37
3.1 Функциональность веб-приложения для выгрузки и просмотра учебных материалов.....	37
3.2 Тестирование веб-приложения.....	41
Заключение	43
Список используемой литературы	44
Приложение А Основные модели пользователя, учебного курса и урока	47
Приложение Б Настройка админ-панели.....	48
Приложение В Настройка обработчика сообщений в чате	49
Приложение Г Основные представления (функции обработчики)	50
Приложение Д Скрипт чата	51

Введение

Вместе с бурным развитием IT-индустрии в последнее десятилетие, в сферу устремилось огромное количество людей, желающих освоить неизведанную профессию. Тяжело отрицать привлекательность данного направления, о высокой зарплате и работе из дома мечтают многие люди. А там, где появляется желание учиться, появляются и учителя.

Вместе с безумным ростом популярности сферы информационных технологий появилось огромное количество онлайн школ, курсов, менторов и прочих мест, предоставляющих разнообразный спектр учебных материалов, программ, и курсов.

Всё это породило огромную конкуренцию, в которой побеждает тот, кто лучше рекламирует себя и тот, кто предоставляет более качественный уровень услуг. Ввиду данных обстоятельств появилась острая необходимость улучшения текущего продукта для поддержания конкурентоспособности на рынке дополнительного образования.

Одним из этапов улучшения продукта стало создание веб-приложения, позволяющего учащимся контактировать с преподавателем, просматривать учебные материалы в любое время, а также коммуницировать с другими учащимися.

Проектирование и реализация готового IT-решения данного запроса является необходимым шагом и представляет практический интерес для компании.

Объектом исследования бакалаврской работы является проектирование и разработка современного веб-приложения для упрощения взаимодействия учащихся с учебными материалами, позволяющее упростить и оптимизировать обучение. Предметом исследования выступает проект по реализации с нуля полноценного IT-решения для компании.

Целью бакалаврской работы является проектирование и разработка веб-приложения, включающего в себя методы для взаимодействия пользователей

в системе, с использованием современных инструментов веб-разработки.

План работ по реализации проекта состоит из следующих пунктов:

- полный разбор и анализ бизнес-процессов;
- проектирование полноценной логической модели;
- полноценное формулирование требований к программному продукту;
- проектирование программного обеспечения;
- выбор средств разработки;
- разработка продукта;
- тестирование продукта;
- оценка эффективности продукта;

В работе будут использованы методологии структурного и объектно-ориентированного проектирования информационных систем. Основной упор направлен на разработку веб-приложения, представляющего собой платформу для взаимодействия преподавателей и учащихся.

Практическая ценность работы заключается в явном повышении конкурентоспособности продукта на обширном рынке дополнительного образования. Эта работа состоит из введения, трёх частей, заключения и списка использованной литературы.

В первом разделе работы мы сосредоточимся на анализе текущих бизнес-процессов организации, создании моделей «Как есть» и «Как должно быть». Сформулируем основные требования к нашей системе. Во втором разделе будут выбраны и описаны инструменты проектирования программного обеспечения. После будут разработаны логические модели информационной системы. В третьем разделе приложение будет реализовано и полноценно протестировано. В заключении подводятся результаты проведенной выпускной квалификационной работы [9].

Бакалаврская работа включает 46 страниц текста, 20 рисунков, 5 таблиц и 20 источников.

1 Проектное моделирование и описание требований

1.1 Характеристика ООО «Изикод»

Информационная система взаимодействия между учащимися и преподавателями компании «Изикод» – веб-приложение, предоставляющее комплекс функций для более качественного обучения, отслеживания успеваемости, взаимодействия с преподавателями и менеджерами для каждого клиента и сотрудника компании «Изикод». Оно обеспечивает удобный и функциональный способ взаимодействия с учебными материалами и коммуникации с преподавателем.

Профильная организация специализируется на предоставлении услуг дополнительного образования в сферах: информационные технологии, изучение иностранных языков, разработка развлекательного интерактивного контента. Компания ориентирует свой продукт на детей и подростков в возрасте от 7 до 18 лет.

Основные функции и возможности информационной системы представляют собой:

- регистрация пользователя через админ-панель менеджером: при приобретении того или иного продукта компании пользователю создаётся аккаунт, самостоятельная регистрация должна быть недоступна, для преподавателя также аккаунт создаётся через админ-панель, самостоятельная регистрация должна быть недоступна;
- взаимодействие ученика и преподавателя: ученик должен иметь возможность загружать и отправлять домашнее задание преподавателю через платформу в разных форматах (fbx, ru, rar, maуa и др.) Преподаватель должен иметь возможность загрузить себе эти файлы и отправить их в ответ. Должна быть реализована возможность отправки текстовых сообщений;

- просмотр записи занятий: ученики должны иметь возможность просматривать записи занятий, заведомо загруженные преподавателем;
- выгрузка записей: преподаватель должен иметь возможность загрузить запись занятия в формате mp4;
- просмотр учебных материалов: учащиеся должны иметь возможность просматривать учебные материалы, прикрепляемые преподавателем к записи;
- выгрузка учебных материалов: преподаватель должен иметь возможность выгружать в систему учебные материалы вместе с записью занятий;
- выбор курса: у ученика должно быть интерактивное меню, где он может выбрать курс и посмотреть по нему учебные материалы.

Информационная система взаимодействия учащегося и преподавателя является неотъемлемой частью качественного продукта компании.

Схема организационной структуры компании ООО «Изикод» представлена на рисунке 1.

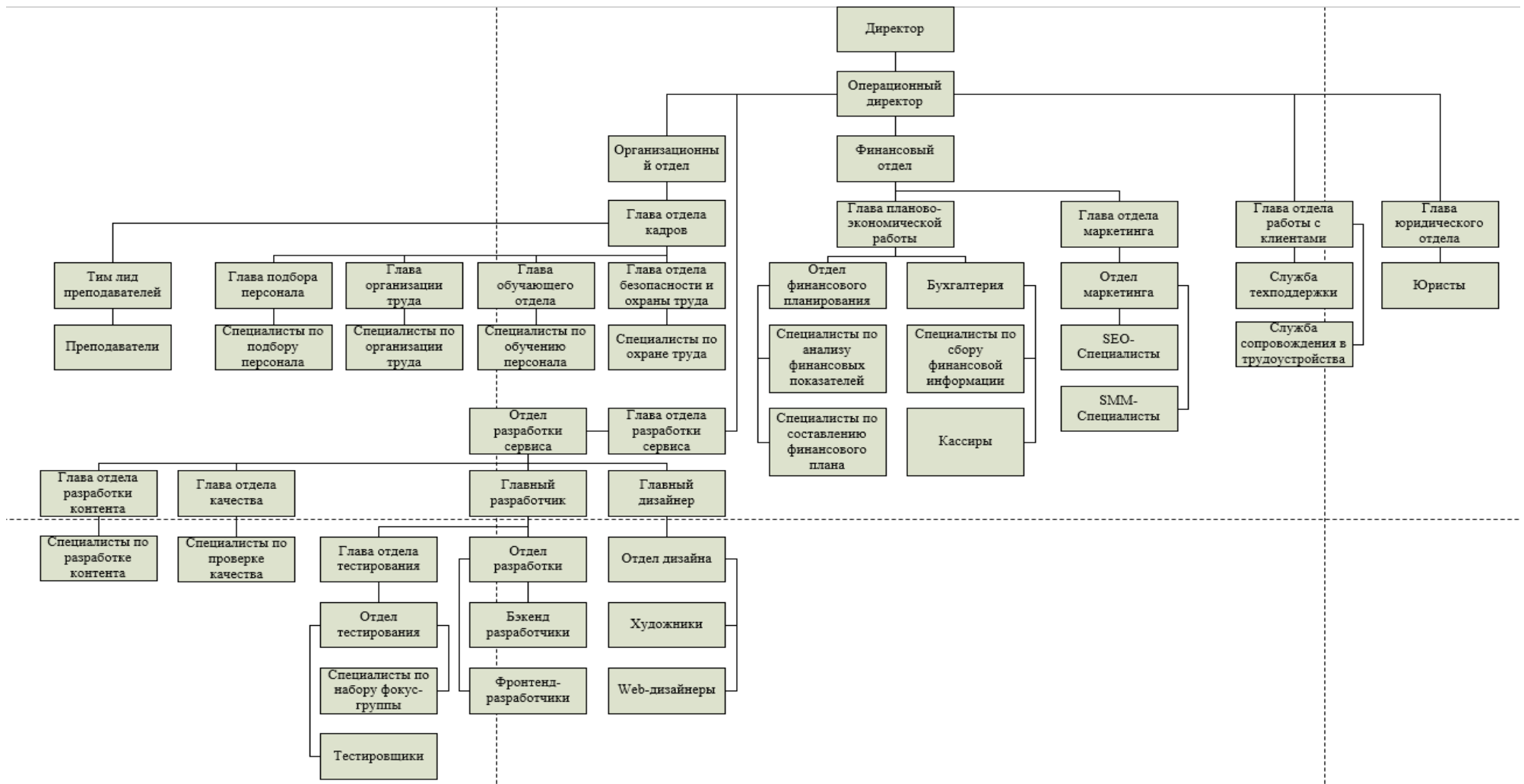


Рисунок 1 - Схема организационной структуры компании ООО «Изикод»

1.2 Выбор методологии концептуального моделирования

В данном разделе описаны существующие и наиболее востребованные в современном мире методологии моделирования бизнес-процессов и не только.

BPMN, или же «Business Process Model and Notation», представляет собой современный стандарт для моделирования бизнес-процессов. BPMN позволяет смоделировать все необходимые элементы, а именно: задачи, события, потоки данных, шлюзы и другие [5].

Диаграмма данной модели состоит из следующих элементов:

- события (начальные, промежуточные и конечные), которые происходят в течение всего бизнес-процесса;
- задачи определяют цель работы процесса, конкретные действия, которые необходимо выполнить;
- шлюзы (исключающие, включающие, параллельные) существуют для разветвления, соединения и параллельно деятельности внутри всего бизнес-процесса;
- потоки показывают ход выполнения всех действий в бизнес-процессе;
- артефакты существуют для описания дополнительных элементов (аннотаций).

Явным преимуществом данной методологии являются простота и интуитивность при визуализации, просто для коммуникации при построении бизнес-процесса [5].

BPMN в современном мире используется в огромном количестве областей, в том числе и в информационных технологиях.

UML, или же «Unified Modeling Language», представляет собой стандартизированный язык моделирования, используемый для проектирования и документирования информационных систем. UML

представляет собой набор графических символов и правил для создания сложных и комплексных структур [3].

Основные элементы UML это:

- диаграммы классов, определяющие классы и их атрибуты, а также взаимоотношение классов [19];
- диаграммы вариантов использования существуют для определения функционального взаимодействия сущностей системы;
- диаграммы последовательностей существуют для отслеживания времени и порядка взаимодействия сущностей в системе;
- диаграммы состояний существуют для демонстрации возможных состояний сущностей системы.

UML аналогично BPMN сильно упрощает визуализацию сложных взаимосвязей и позволяет быстрее и комфортнее отлаживать процессы внутри компании.

IDEF0, или же «Integration Definition for Function», представляет метод моделирования функциональных аспектов системы. Его чаще используют для визуализации и документирования текущих и необходимых бизнес-процессов и зависимостей [2].

Элементы IDEF0 это:

- блоки функций, которые представляют основные функции системы;
- стрелки для иллюстрации потоков управления и информации;
- механизмы управления показывают, как функции управляются.

1.3 Разработка и анализ модели бизнес-процесса «Как есть»

Функциональная модель IDEF0 отображает текущий функционал исследуемого объекта, а именно текущие действия и взаимосвязи между ними [12].

Контекстная диаграмма состоит из 3 уровней. На верхнем уровне, представленном на рисунке 2, находится основной блок, описывающий

текущий процесс взаимодействия учащихся с учебными материалами.

В таблице 1 представлены компоненты управления, входа, выхода и механизмы.

Таблица 1 - Компоненты управления, входа, выхода и механизмы

Входы	Выходы	Управление	Механизмы
Запрос записи и материалов	Удовлетворённый ученик с материалами	Внутренние регламенты	Ученик
Информация			Отдел сопровождения

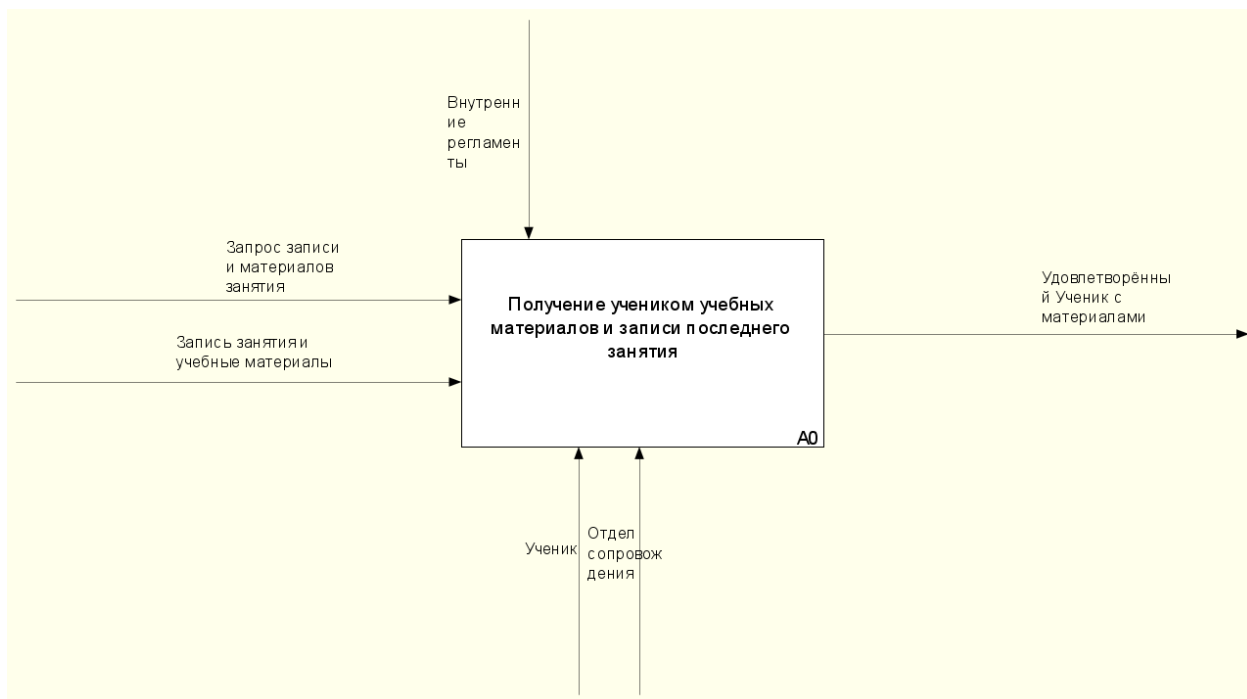


Рисунок 2 - Контекстная диаграмма бизнес-процесса «Взаимодействие обучаемого с учебными материалами» AS-IS

На рисунке 3 представлена декомпозиция контекстной диаграммы AS-IS.

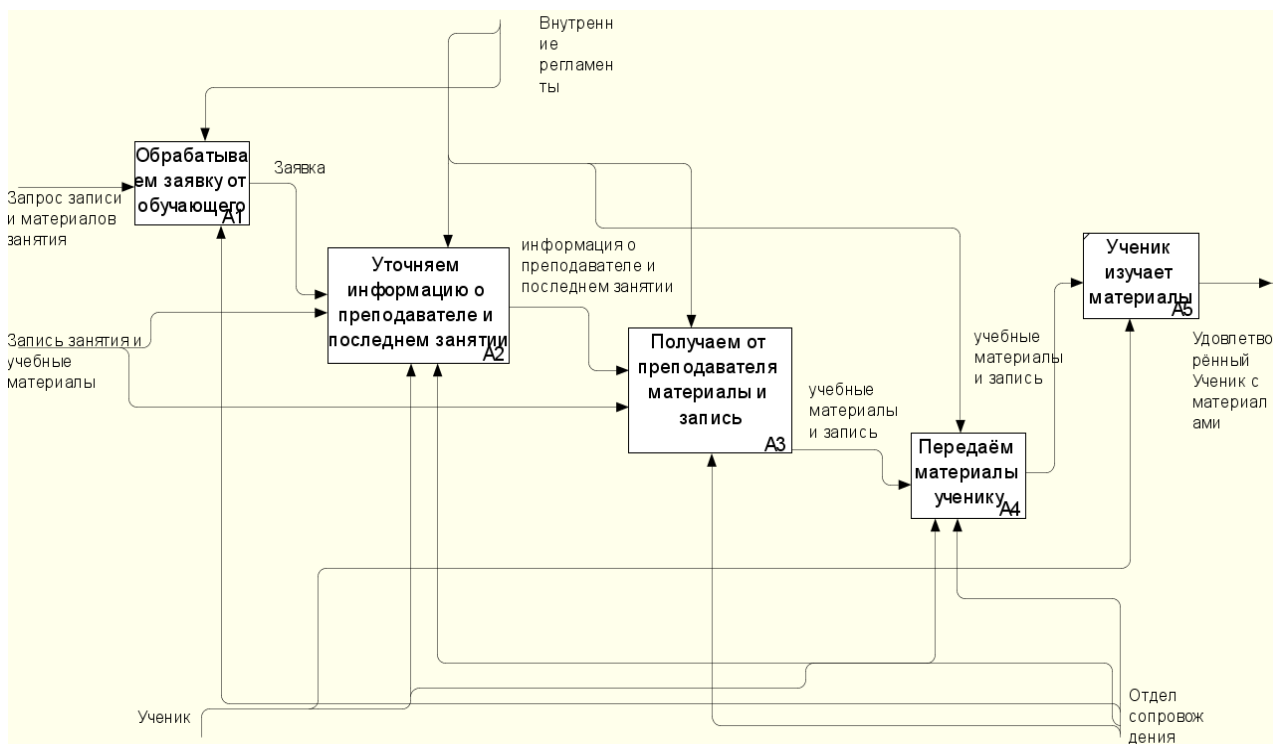


Рисунок 3 - Декомпозиция контекстной диаграммы бизнес-процесса
«Взаимодействие обучаемого с учебными материалами»

В данный момент взаимодействия ученика с учебными материалами сильно завязано на скорости работы отдела сопровождения и преподавателя.

Сам процесс состоит из пяти этапов:

- «Обработка заявки от обучающегося» – процесс получения и обработки информации от обучающегося о необходимости получения учебных материалов;
- «уточнение информации о преподавателе и последнем занятии» – процесс уточнения данных о том, когда было занятие, кто преподаватель обучаемого, на каком курсе он учится;
- «Получение от преподавателя материалов и записи» – процесс передачи преподавателя записи и учебных материалы по необходимому занятию. Данный процесс является самым долгим и более подробно представлен на рисунке 4;

- «Передача материалов ученику» – мы передаём ученику материалы и запись занятия в формате ссылок;
- «Ученик изучает материалы» – обучаемый изучает материалы и запись.

На рисунке 4 представлена декомпозиция процесса «Получение от преподавателя записи и материалов»

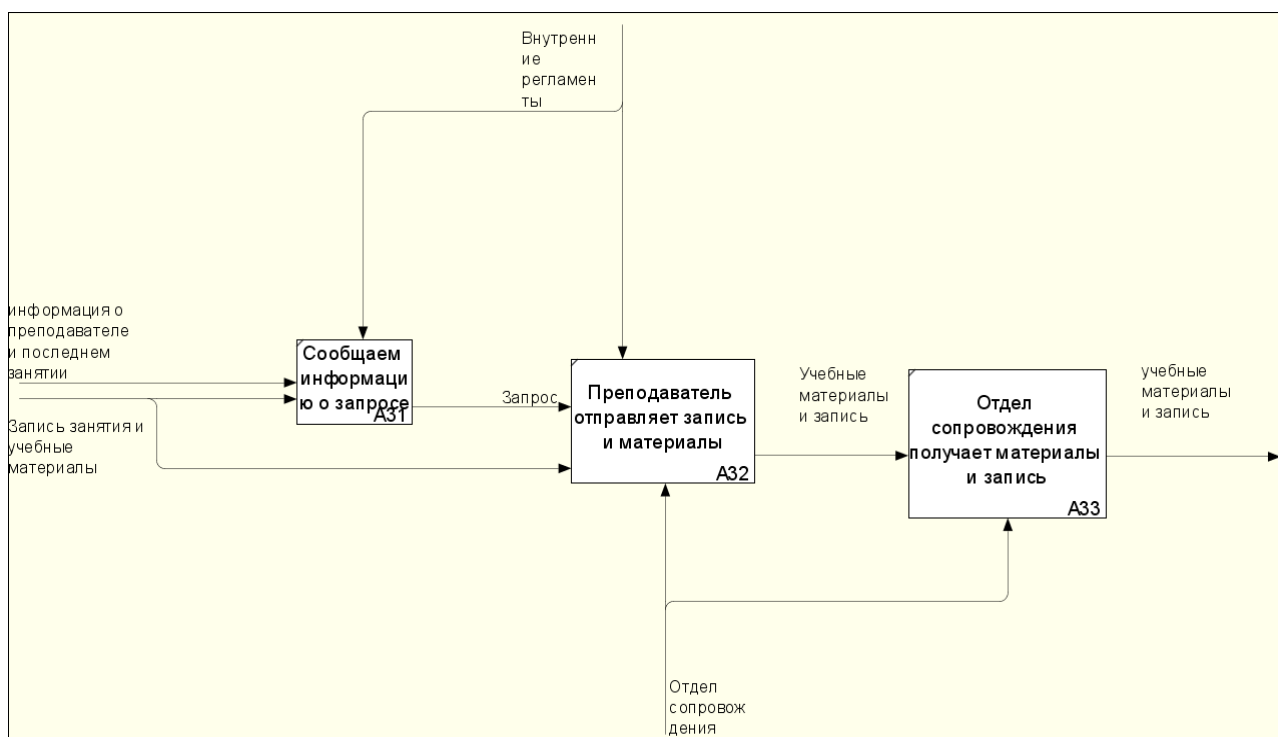


Рисунок 4 - Декомпозиция контекстной диаграммы «Получение от преподавателя записи и материалов»

Процесс «Получение от преподавателя материалов и записи» состоит из трех этапов:

- «Отправка преподавателю запроса на получение материалов» - отправляем преподавателю запрос на передачу всех учебных материалов с необходимого занятия;

- «Отправка данных преподавателем» – преподаватель загружает материалы на сторонний облачный сервис и отправляет ссылки на них менеджерам отдела сопровождения;
- «Отдел сопровождения получает материалы и запись» – отдел сопровождения получает материалы и запись от преподавателя.

Данный процесс сильно затягивается из-за загруженности отдела сопровождения и преподавателя. Необходимо большое количество времени на передачу всех материалов, не всегда преподаватель может ответить сразу. Менеджеры отдела сопровождения, как и преподаватель, не работают круглосуточно. Данные факторы сильно увеличивают риск отказа ввиду неудовлетворённости сервисом. Благодаря наличию веб-приложения можно будет убрать из всей этой цепочки работу отдела сопровождения. Преподавателю лишь нужно будет загрузить запись и материалы на онлайн-платформу, где обучаемый сам легко их найдёт и будет иметь к ним доступ в любое время суток. Для расчёта эффективности будем использовать формулу интенсивности труда, далее ИТ. $ИТ = \frac{\text{Человеко-часы}}{\text{Объём работы}}$. Оценка эффективности предложенного метода представлена в таблице 2.

Таблица 2 - Сравнение эффективности работы

Модель	Количество человеко-часов, необходимое для выполнения задачи	Объём работы	Коэффициент интенсивности труда
Текущий вид взаимодействия через менеджеров	4	100	0,04
Взаимодействие через веб-приложение	1	100	0,01

В результате сравнения явно уменьшение загруженности сотрудников при внедрении нового процесса. Однако самое важное здесь – значительный рост скорости получения информации учеником. Всё, что необходимо для преподавателя – загрузить данные на новую платформу и обучаемый легко сможет их найти.

Реорганизуем рассматриваемый процесс, добавив в него механизм «веб-приложение». Можно полностью убрать из процесса механизм «отдел сопровождения» и вход «информация». Диаграмма процесса «как должно быть» представлена на рисунке 5. Таким образом, по итогу процесс сократился до трёх этапов и стал в разы эффективнее.

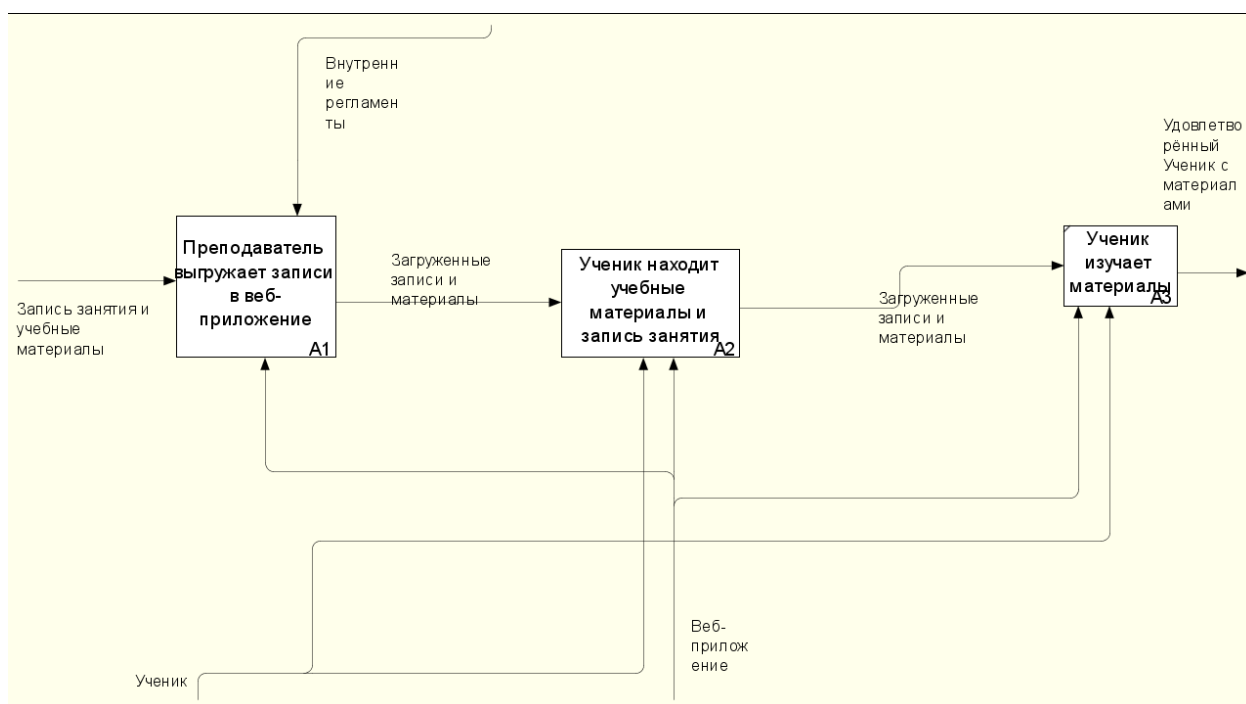


Рисунок 5 - Контекстная диаграмма бизнес-процесса «Взаимодействие обучающегося с учебными материалами» TO-DO

На диаграмме видно, насколько уменьшился весь бизнес-процесс. Данная оптимизация должна значительно увеличить удержание обучаемого.

1.4 Разработка требований к ПО

Для разработки требований к программному обеспечению (ПО) мы будем использовать методологию FURPS+.

FURPS+ – это методология для оценки и управления качеством программного обеспечения [13]. Каждая буква означает одно из требований к системе:

- **Functionality** – функциональность, способность программного продукта выполнять все необходимые функции и решать поставленные задачи;
- **Usability** – удобство использования, простота и удобство пользования итоговым продуктом, интуитивность интерфейса;
- **Reliability** – надежность программного обеспечения, возможность работы программы без сбоев в течение заданного периода;
- **Performance** – производительность всей системы, возможность работы необходимого количества пользователей, скорость работы всей системы;
- **Supportability** – возможность поддержки программного обеспечения после реализации.

Стоит отметить, что разрабатываемое в данный момент веб-приложение является тестовой версией, предназначенной для фокус-группы. Это нужно для более качественного анализа требований клиента. Ввиду этого требования к отказоспособности и производительности сильно упрощены и будут увеличиваться по ходу дальнейшей работы над проектом.

В таблице 3 представлены требования к системе.

Таблица 3 - Требования к ПО по методологии FURPS+

№	Требование	Статус	Полезность	Риск	Стабильность
Functionality — Функциональные требования					
1	Возможность выгрузки записей и материалов в форматах: mp4, docs, fbx	Одобрено	Критическая	Минимальный	Высокая
2	Возможность чата с преподавателем в реальном времени	Одобрено	Средняя	Высокий	Средняя
3	Возможность просмотра обучающимся записи и материалов в любое время	Одобрено	Критическая	Минимальный	Высокая
Usability— Требования к удобству использования					
4	Дружественный интуитивный интерфейс	Одобрено	Критическая	Средний	Средняя
Reliability— Требования к надежности					
5	Допустимая частота/периодичность сбоев: 1 раз в 150 часов	Одобрено	средняя	Средний	Средняя
6	Среднее время сбоев: 1 час	Одобрено	Важная	Средний	Средняя
7	Возможность восстановления системы после сбоев: 5 часов	Одобрено	Важная	Средний	Средняя
8	Режим работы: 24/7/365	Одобрено	Важная	Средний	Средняя
Performance — Требования к производительности					
9	Допустимое количество одновременно работающих пользователей: 100	Одобрено	Средняя	Средний	Средняя

Продолжение таблицы 3

10	Время реакции на возникновение аварийной ситуации: 30 минут	Предложенное	Важная	Низкий	Средняя
11	Возможность к масштабируемости систем до большего числа фокус-группы	Предложенное	Маловажная	Низкий	Средняя
Supportability — Требования к поддержке					
12	Время устранения критических проблем: в течение 5 часов после сбоя	Одобренное	Важная	Средний	Средняя
13	Возможность переноса ПО на другие платформы	Предложенное	Важная	Средний	Низкая
Проектные ограничения					
14	Возможность переноса ПО на другие платформы после этапа тестирования на фокус-группе	Предложенное	Важная	Средний	Низкая
15	Средняя стоимость владения	Предложенное	Важная	Средний	Средняя

Вводы по разделу 1

В данном разделе представлен анализ предметной области, основных бизнес-процессов и сформулированы требования к ПО. При анализе с использованием моделей «AS-IS» и «TO-BE» показана необходимость оптимизации работы компании с использованием информационных систем. Выявлены основные минусы текущего бизнес-процесса «Взаимодействие преподавателя и обучаемого» и предложено решение для его кардинального изменения и улучшения. В результате анализа требований к ПО мы обозначили основные моменты разработки. Полученные данные будем использовать как основу для реализации программного решения, целью которого является улучшение показателей удовлетворённости клиента.

2 Проектирование программного обеспечения. Выбор программных средств разработки

2.1 Описание инструментов для разработки программного обеспечения

Создание серверной части веб-приложения.

Для создания серверной части веб-приложения мы будем использовать python версии 3.10. Python высокоуровневый язык программирования с динамической типизацией [18]. Он достаточно прост, ввиду чего обрёл огромную популярность. В нём легко соблюдать принцип DRY (Don't repeat yourself). Из-за особенностей обособления и контроля вложенности код на этом языке хорошо читаем. Python обладает мощной стандартной библиотекой и огромным количеством фреймворков. Его используют для создания веб-приложений, в работе с Big Data и машинном обучении.

Django является самым популярным фреймворком для создания серверной части веб-приложений. В нём присутствует огромное количество готовых решений для любых задач, таких как: авторизация и аутентификация, работа с формами, работа с админ-панелью и многим другим [16].

Явным преимуществом Django является объектно-реляционная модель отображения. Она позволяет писать SQL запросы прямо внутри представлений, что сильно упрощает работу с базами данных.

Django отлично масштабируется, что позволит нам создать приложение, способное в будущем выдержать большие нагрузки. Для Django существует огромное количество сторонних библиотек, одной из них мы воспользуемся [1].

Websocket – протокол связи, обеспечивающий полнодуплексную связь между клиентом и сервером через единое TCP-соединение [20]. Отличием от

HTTP является возможность двустороннего одновременного обмена данными, что является основной всех онлайн-чатов.

Django channels – это библиотека для Django, добавляющая поддержку WebSocket и других протоколов связи в проект. Это позволит сделать наше приложение асинхронным, чтобы обрабатывать события в реальном времени.

Frontend часть веб-приложения

Визуальная часть веб-приложения была изготовлена без дизайн-проекта ввиду того, что разрабатываемое ПО было тестовым и первоочередной задачей была разработка функциональной части.

HTML – язык гипертекстовой разметки, используемый для создания структуры веб-страницы. Структура создаётся при помощи специальных тегов, определяющих его содержание: img, a, h1, p и другие. Все они служат контейнерами для каких-либо данных [10].

CSS – каскадные таблицы стилей. Они обладают огромным количеством свойств, определяющих внешний вид страницы, таких как: шрифты, размеры, цвета, расположение элементов и другие [4]. Создание веб-страницы на чистых HTML и CSS является трудоёмким и проблемным занятием. К нему прибегают на финальных стадиях разработки, когда весь функционал определён и составлен дизайн макет. Мы же будем использовать готовое решение для нашего проекта.

Bootstrap – фреймворк для разработки веб-страниц. Он представляет собой набор готовых CSS классов для определения внешнего вида страницы и скриптов на JavaScript для придания странице интерактивности. Мы будем использовать его из-за простоты и скорости работы с ним [15].

Выбор IDE

IDE – программа, предназначенная для разработки ПО. Обычно представляет из себя набор инструментов и функций, таких как: редактор кода, автодополнения, инструменты контроля версий, отладчик, компилятор и другие [7].

PyCharm – IDE, разработанная компанией JetBrains. Она создана под язык программирования Python и отлично работает с Django, имея встроенные инструменты отладки под данный фреймворк [17]. PyCharm имеет систему контроля версий Git, полностью совместимую с GITHUB, удобный редактор кода с визуальными сообщениями об ошибках и автодополнениями. Также отлично поддерживает работу с HTML и CSS.

2.2 Проектирование статической структуры системы

Статическая структура будет спроектирована с использованием одной из диаграмм UML, а именно «Диаграммы классов» [6].

Диаграмма классов представляет собой графическое представление взаимосвязей и отношений между классами, их атрибутов и методов.

Django основан на объектах, ввиду чего данная диаграмма идеально подходит для проектирования статической структуры нашей системы.

Подробнее разберём основные элементы данной диаграммы.

Классы являются основой объектно-ориентированного программирования. Они представляют собой абстракцию объектов, которые от них наследуются. В классах описываются все атрибуты (поля) и методы (операции), которые задают и контролируют поведение объекта. На диаграмме он представлен как прямоугольник с тремя секциями: название класса, атрибуты класса, методы класса [6].

Отношения между классами указывают на связи и зависимости между классами. Существует большое количество видов отношений, но мы разберём используемые нами:

Наследование: один из классов наследует свойства и методы другого, имея свои особенные свойства. Наследование создаёт чёткую и строгую иерархию классов внутри проекта [8].

Ассоциация: объекты одного класса связаны с объектами другого. Связь может быть односторонней и двусторонней, а также иметь разные

связи. Виды связей бывают следующими: один к одному, один ко многим, многие к многим, многие к одному [8].

Атрибуты являются данными, которые хранят в себе классы, а именно значения их переменных. Например, атрибутами класса автомобиль будут: двигатель, дверь, цвет и другие. У каждого атрибута должно быть значение и тип данных [6].

Методы внутри классов определяют поведение для всех объектов данного класса. В программировании это функции, выполняемые классом. Например, у класса автомобиль есть следующие методы: завестись, ехать, тормозить, поднять крышку. Это значит, что у объекта «Жигули», который наследуется от этого класса будут все перечисленные методы [6].

Модификаторы доступа контролируют уровень доступа к различным атрибутам и методам тех или иных классов.

Существуют следующие модификаторы: `public`, `private`, `protected`, `default`. `Public` предоставляет неограниченный доступ к атрибутам и методам из любого места программы. `Private` предоставляет доступ только внутри класса. `Protected` же добавляет возможность доступа только из класса и его наследников. `Default` предоставляет доступ только из того же пакета [6].

Диаграмма классов представлена ниже на рисунке 6.

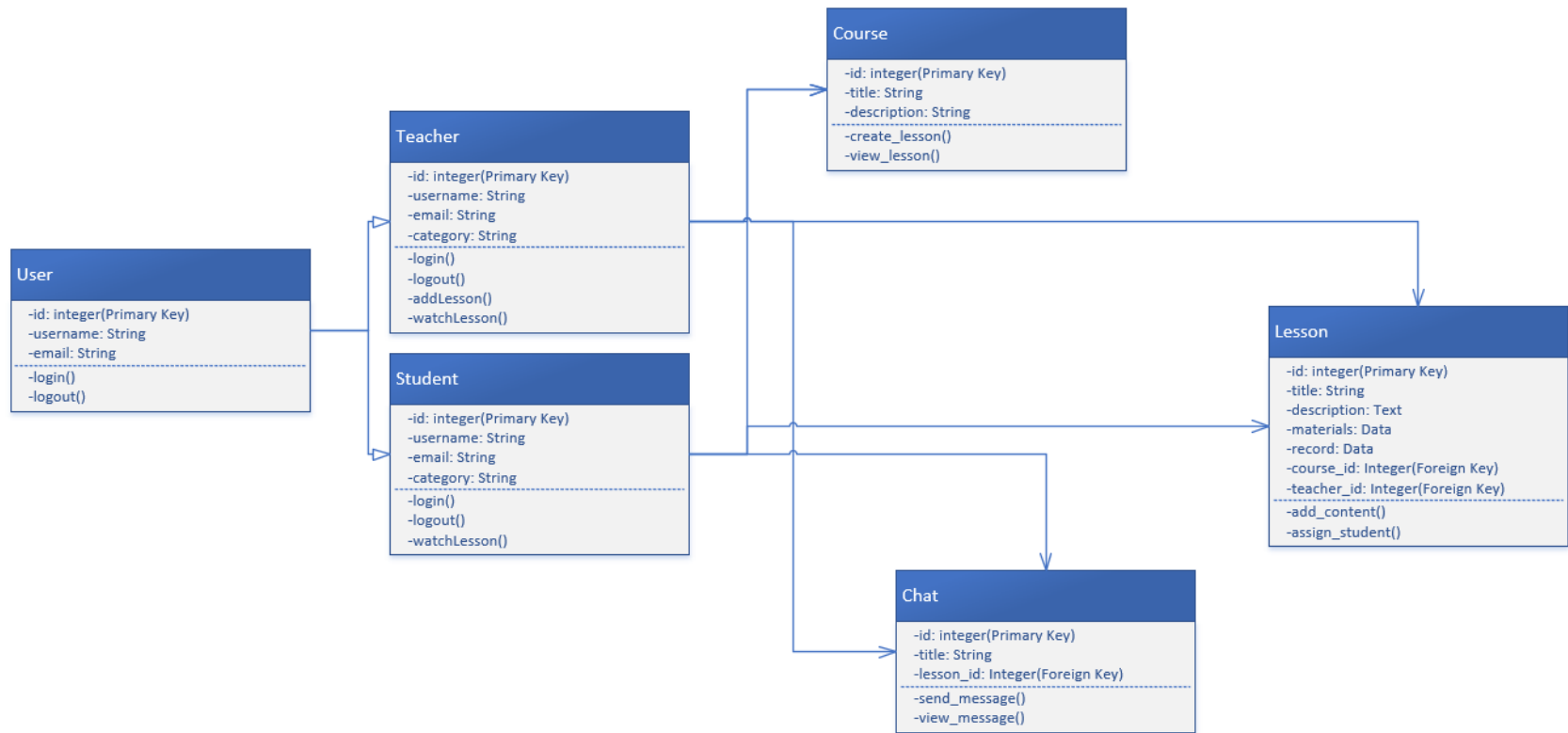


Рисунок 6 - Диаграмма классов

В таблице 4 представлен перевод используемых терминов.

Таблица 4 – перевод используемых терминов

Термин	Значение
User	Пользователь
Username	Имя пользователя
Email	Электронная Почта
Login	Войти в аккаунт
Logout	Выйти из аккаунта
Category	Категория
Id	Идентификационный номер
Title	Заголовок
Description	Описание
Create	Создать
View	Посмотреть
Lesson	Урок
Content	Контент
Course	Курс
Student	Студент
Teacher	Преподаватель
Chat	Чат
Send	Отправить
Message	Сообщение

Рассмотрим подробнее классы, представленные на диаграмме.

На рисунке 7 изображён класс «User».



Рисунок 7 - Класс «User»

Данный класс имеет следующие атрибуты: «id», «username», «email».

Методы, у класса, следующие: «login», «logout»:

- «id» является первичным ключом для связи с другими классами, а также контролирует уникальный номер каждого созданного объекта;
- «username» обозначает условное имя пользователя на платформе, которое он может использовать для входа, также оно служит для выбора в поле формы создания урока как тот, для кого создаётся урок;
- «email» содержит почту пользователя, привязанную к его аккаунту в системе;
- «login» метод, позволяющий войти на платформу под своим логином и паролем, выданными менеджерами;
- «logout» метод, позволяющий выйти с платформы.

На рисунке 8 изображён класс «Teacher», наследуемый от «User».

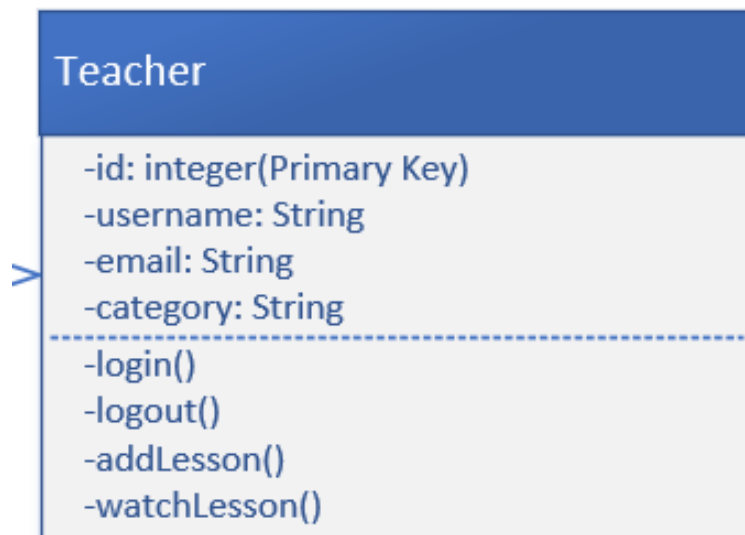


Рисунок 8 - Класс «Teacher»

Данный класс имеет следующие атрибуты: «id», «username», «email», «category». Методы, у класса, следующие: «login», «logout», «addLesson», «watchLesson»:

- «id» является первичным ключом для связи с другими классами, а также контролирует уникальный номер каждого созданного объекта;
- «username» обозначает условное имя пользователя на платформе, которое он может использовать для входа, также оно служит для выбора в поле формы создания урока как тот, для кого создаётся урок;
- «email» содержит почту пользователя, привязанную к его аккаунту в системе;
- «category» содержит информацию о том, что данный класс является преподавателем;
- «login» метод, позволяющий войти на платформу под своим логином и паролем, выданными менеджерами;
- «logout» метод, позволяющий выйти с платформы;

- «addLesson» проводит проверку на то, может ли данный объект добавить урок;
- «watchLesson» проводит проверку на то, может ли данный объект просмотреть урок.

На рисунке 9 изображён класс «Student», наследуемый от «User».

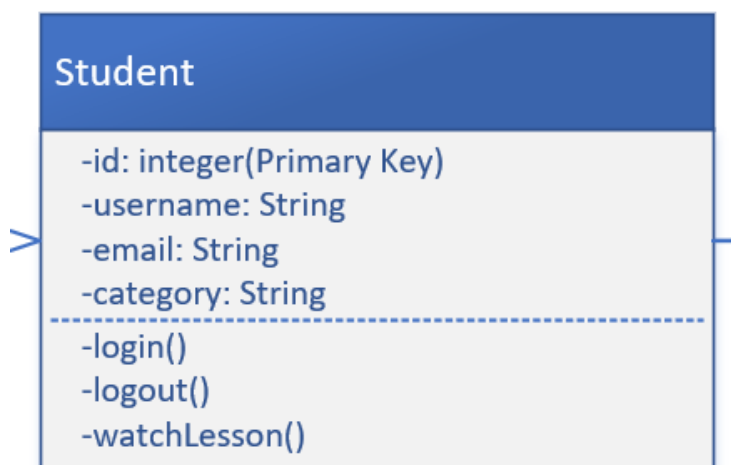


Рисунок 9 - Класс «Student»

Данный класс имеет следующие атрибуты: «id», «username», «email», «category». Методы, у класса, следующие: «login», «logout», «watchLesson»:

- «id» является первичным ключом для связи с другими классами, а также контролирует уникальный номер каждого созданного объекта;
- «username» обозначает условное имя пользователя на платформе, которое он может использовать для входа, также оно служит для выбора в поле формы создания урока как тот, для кого создаётся урок;
- «email» содержит почту пользователя, привязанную к его аккаунту в системе;

- «category» содержит информацию о том, что данный класс является студентом;
- «login» метод, позволяющий войти на платформу под своим логином и паролем, выданными менеджерами;
- «logout» метод, позволяющий выйти с платформы;
- «watchLesson» проводит проверку на то, может ли данный объект просмотреть урок.

На рисунке 10 изображён класс «Course».

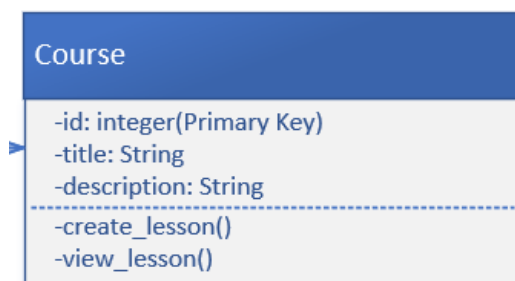


Рисунок 10 - Класс «Course»

Данный класс имеет следующие атрибуты: «id», «title», «description».

Методы, у класса, следующие: «create_lesson», «view_lesson»:

- «id» является первичным ключом для связи с другими классами, а также контролирует уникальный номер каждого созданного объекта;
- «title» содержит в себе название текущего курса;
- «description» содержит полное описание текущего курса;
- «create_lesson» метод, позволяющий внутри данного курса создавать уроки преподавателю, выделяя студента, для которого урок создаётся;
- «view_lesson» метод, позволяющий посмотреть все уроки в данном курсе для преподавателя.

На рисунке 11 изображён класс «Lesson».



Рисунок 11 - Класс «Lesson»

Данный класс имеет следующие атрибуты: «id», «title», «description», «materials», «record», «course_id», «teacher_id». Методы, у класса, следующие: «add_content», «assign_student»:

- «id» является первичным ключом для связи с другими классами, а также контролирует уникальный номер каждого созданного объекта;
- «title» содержит в себе название текущего урока;
- «description» содержит полное описание текущего урока;
- «materials» содержит файл с материалами к уроку, добавленный преподавателем;
- «record» содержит запись данного урока, добавленную преподавателем;
- «course_id» вторичный ключ для соединения с моделью курса, в котором лежит урок;
- «teacher_id» вторичный ключ для соединения с моделью преподавателя, который создал урок;
- «add_content» метод, позволяющий внутри данного урока добавлять материалы;

- «assign_student» метод, позволяющий к данному уроку прикрепить ученика во время его создания.

На рисунке 12 изображён класс «Chat».

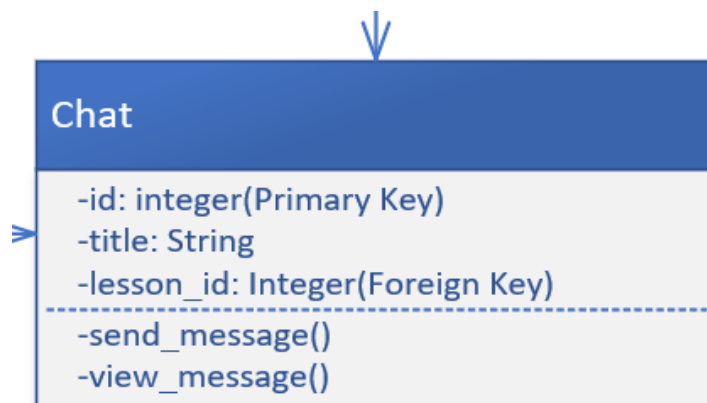


Рисунок 12 - Класс «Chat»

Данный класс имеет следующие атрибуты: «id», «title», «lesson_id».

Методы, у класса, следующие: «send_message», «view_message»:

- «id» является первичным ключом для связи с другими классами, а также контролирует уникальный номер каждого созданного объекта\$«title» содержит в себе название текущего урока;
- «lesson_id» вторичный ключ для связи чата и урока;
- «send_message» метод, позволяющий внутри данного урока отправлять сообщения в чате;
- «view_message» метод, позволяющий внутри данного урока читать сообщения в чате.

Программный код основных моделей пользователя, учебного курса и урока представлен в приложении А.

Перед выбором СУБД нам необходимо создать схему архитектуры нашего веб-приложения. Она представлена на рисунке 13.

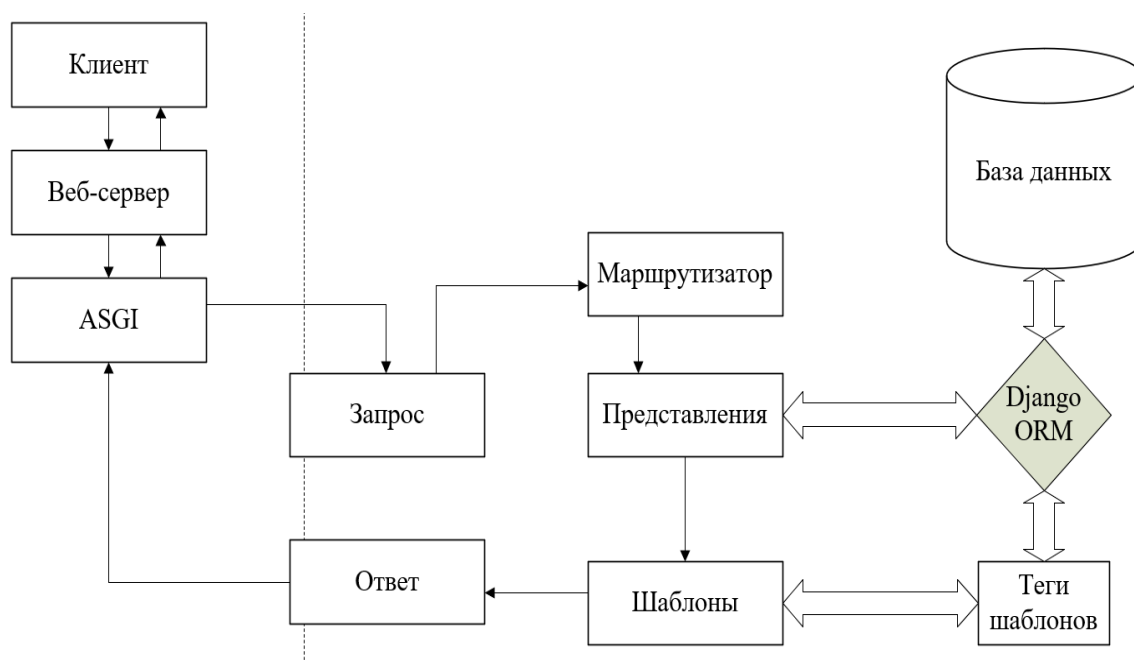


Рисунок 13 - Схема архитектуры веб-приложения

Клиент через веб-сервер и протокол ASGI асинхронного взаимодействия работает с нашим веб-приложением. Отправленный запрос принимает маршрутизатор и выполняет необходимое представление (функцию), которая, через Django ORM (объектно-реляционную модель Django), достаёт из базы данных необходимые элементы и передаёт их в шаблон, который отрисовывает представление и отправляет в качестве ответа через протокол ASGI.

2.3 Выбор СУБД

СУБД или же система управления базами данных представляет собой программное обеспечение, которое предназначено для работы с базами данных [11]. Основные функции СУБД:

- хранение данных: СУБД хранят данные в структурированном формате, обеспечивая к ним качественный доступ и взаимодействие с ними [14];

- управление данными: СУБД предоставляют интерфейс для добавления, удаления и изменения данных в БД;
- выборка данных: СУБД предоставляют интерфейс для запросов и получению данных по определённым характеристикам;
- обеспечение безопасности: СУБД предоставляют интерфейс для обеспечения безопасности данных с механизмами авторизации и аутентификации;
- целостность данных: СУБД обеспечивают целостность всех данных путём механизмов защиты данных от некорректных изменений или удалений.

СУБД бывают разных типов, в зависимости от вида управления данным. Каждый имеет свои особенности [11]. Вот основные из них:

- реляционные СУБД: данные хранятся в таблицах, для запросов используется язык SQL;
- нереляционные СУБД: хранят данные в неструктурированном виде, эффективны при работе с большими наборами данных (к примеру, при анализе);
- объектно-ориентированные СУБД: хранят объекты классов при программировании, обеспечивают прямую связь между данными и объектами.

Мы остановим свой выбор на реляционных СУБД. Они более всего подходят нам по совокупности параметров. Мы будем выбирать из PostgreSQL, MySQL и SQLite3.

Перед тем как перейти к сравнению, для начала смоделируем диаграмму вариантов использования UML. Это поможет нам сформулировать необходимые требования к СУБД.

Диаграмма изображена на рисунке 14.

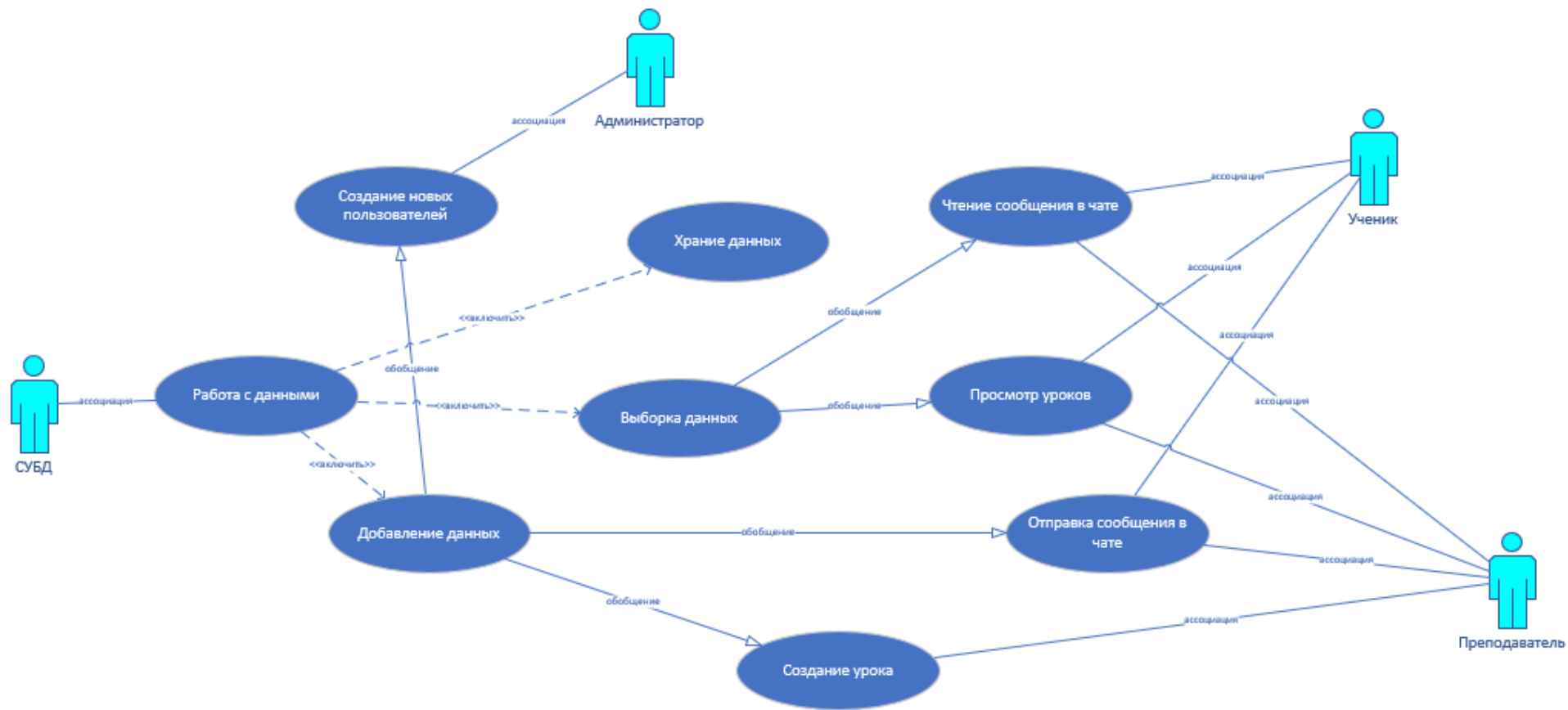


Рисунок 14 - Диаграмма вариантов использования.

Диаграмма взаимодействия состоит из следующих элементов:

- актёры — пользователи и другие системы, взаимодействующие друг с другом;
- варианты использования — функциональные возможности системы;
- связи — отношения вариантов использования и актёров между собой.

На диаграмме можно увидеть, что функционал нашей системы ограничен несколькими функциями. В данный момент времени нам важна мобильность системы и возможность практически моментального развёртывания.

Для финального выбора СУБД составим сравнительную таблицу.

Таблица 5 - Сравнение СУБД

Характеристика	SQLite3	PostgreSQL	MySQL
Тип	Встраиваемая СУБД	Серверная СУБД	Серверная СУБД
Поддерживаемые языки	C, C++, Python, Java, PHP, и другие	C, C++, Python, Java, Perl, и другие	C, C++, Python, Java, PHP, и другие
Лицензия	Общественная лицензия SQLite	PostgreSQL лицензия	GPL лицензия для Community Edition, коммерческая для Enterprise Edition
Типы данных	Поддерживает ограниченный набор типов	Обширный набор типов данных	Обширный набор типов данных
Язык SQL	Поддерживает SQL	Полностью совместим с SQL	Полностью совместим с SQL
Масштабируемость	Не рекомендуется для крупных проектов	Очень масштабируемая	Очень масштабируемая

Продолжение таблицы 5

Поддержка транзакций	Поддерживает транзакции	Полная поддержка транзакций	Полная поддержка транзакций
Развертывание	Легко развертывается	Требует настройки сервера	Требует настройки сервера
Поддержка индексов	Поддерживает индексы	Полная поддержка индексов	Полная поддержка индексов

По всем критериям SQLite3 проигрывает старшим братьям. Однако для нас она подходит идеально. SQLite3 – это встраиваемая СУБД, обеспечивающая реляционное управление базой данных. В отличие от остальных, данная СУБД представляет из себя не отдельный сервер базы данных, она встраивается напрямую в приложение, что идеально для систем с высокими требованиями мобильности. SQLite3 идеально подходит для небольших и средних проектов, как наш.

Выводы по разделу 2

Во втором разделе представлено описание всех необходимых инструментов для создания веб-приложения, определены программные средства для разработки веб-приложения.

Спроектирована статическая структура системы, используя диаграмму классов UML. Благодаря описанию каждого класса физическое создание веб-приложения станет в разы проще.

Проведен анализ и осуществлен выбор необходимой реляционной СУБД, составив все необходимые функциональные требования к системе при помощи диаграммы вариантов использования.

3 Реализация веб-приложения для выгрузки и просмотра учебных материалов

3.1 Функциональность веб-приложения для выгрузки и просмотра учебных материалов

Рассмотрим функционал реализованного веб-приложения. Далее подробно будут представлены основные функциональные особенности разработанного веб-приложения.

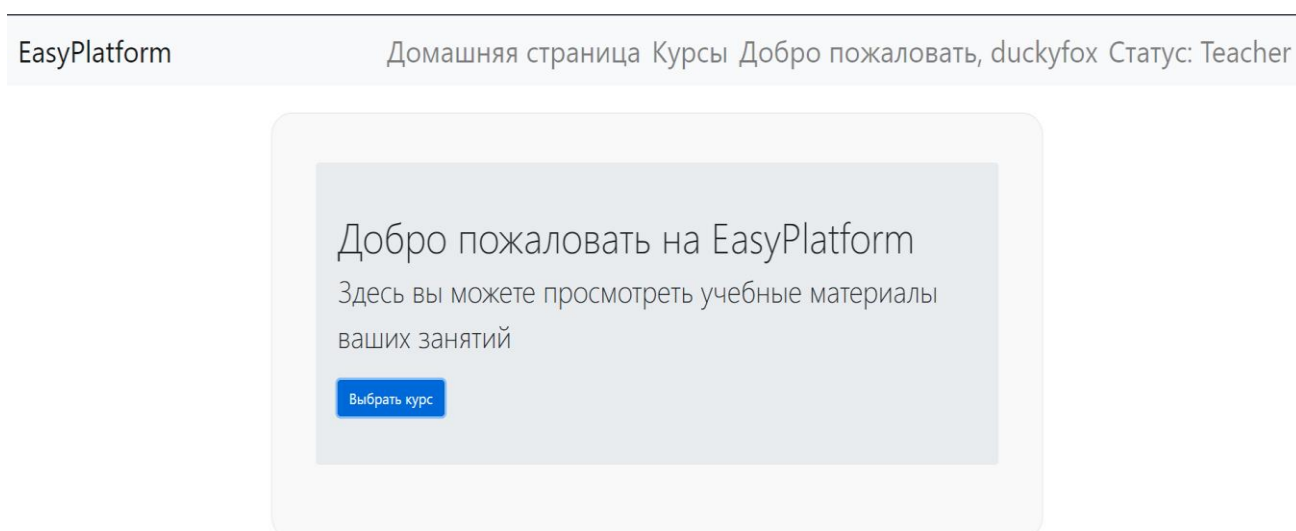


Рисунок 15 - Главная страница

На рисунке 15 представлена главная страница сайта, на ней пользователь может перейти ко входу в систему под выданными ему данными. Программный код настройки админ-панели представлен в приложении Б. Программный код настройки обработчика сообщений в чате представлен в приложении В.

Login

Username

aaa

Password

.....

Login

Рисунок 16 - Страница входа

На рисунке 16 представлена страница входа на платформу.

EasyPlatform

Список курсов

Фронтенд разработка

Бэкенд разработка

3д-дизайн

UE4

Рисунок 17 - Страница списка курсов

На рисунке 17 представлена страница со всеми существующими на данный момент курсами, нажав на один из них, пользователь попадает к урокам в этом курсе. Программный код основных представлений (функций обработчиков) представлен в приложении Г.

Бэкенд разработка

Разработка серверной части приложений

Добавить урок

Уроки

Основы python часть 1

Основы python часть 2

Рисунок 18 - Страница списка уроков

На рисунке 18 представлена страница со всеми существующими на данный момент уроками в курсе, уроки создаются для определённого ученика, другие ученики их не видят. Кнопка «добавить урок» доступна только для преподавателя. Если нажать на название урока, мы перейдём в него. Пример скрипта чата представлен в приложении Д.

Создать урок

Title:

На данном занятии мы изучили основные понятия и типы переменных

Content:

Course:

Students:

Lesson recording:

Lesson materials:

Рисунок 19 - Страница создания урока

На рисунке 19 представлена страница с формой для создания урока. В ней преподаватель задаёт название, описание, курс, к которому создаётся

урок, ученика, которому урок предназначен, прикрепляет запись и текстовый документ с материалами.

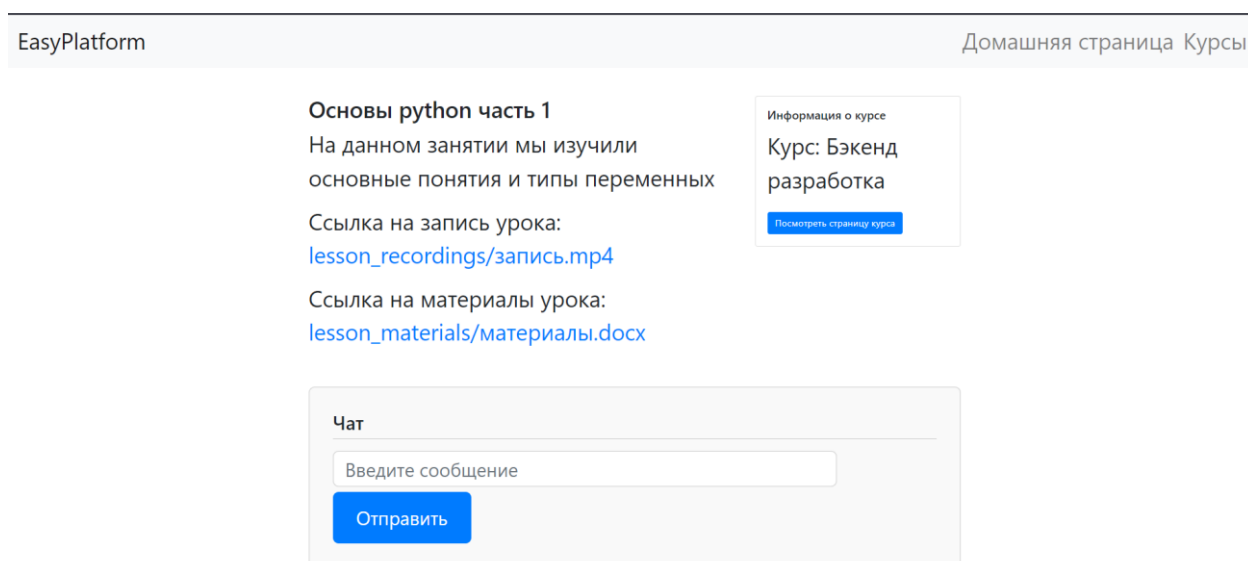


Рисунок 20 - Страница урока

На рисунке 20 представлена страница урока, на ней присутствует название, описание, ссылки на запись и материалы, информация о курсе и чат с учеником/преподавателем. При нажатии на ссылку запись документы скачиваются.

3.2 Тестирование веб-приложения

Тестирование веб-приложения проводилось по методике чёрного ящика. Для этого нами были разработаны тест-кейсы, которые выполняли несколько респондентов.

Целью тестирования была проверка работоспособности и эргономичности веб-приложения. Респонденты отметили простой и интуитивный интерфейс и лёгкость в дизайне.

Тест-кейсы:

- просмотр списка курсов. В результате проверки не выявлено ошибок, страница с текущими курсами отображается для всех пользователей системы;
- просмотр списка уроков. В результате проверки не выявлено ошибок, страница с текущими уроками отображается в зависимости от роли пользователя;
- просмотр урока. В результате проверки не выявлено ошибок, страница с текущим уроком отображается для всех пользователей системы, у которых есть к ней доступ;
- отправка сообщений. В результате проверки не выявлено ошибок, сообщения отправляются и доходят до получателей.

Тестирование проводилось вручную силами нескольких респондентов. Они не имели информации об устройстве системы, что соответствует методологии тестирования «чёрного ящика».

Заключение

Выпускная квалификационная работа, посвящённая разработке веб-приложения для выгрузки и просмотра учебных материалов.

Мы можем смело утверждать, что поставленная цель была достигнута. Используя различные методологии, мы спроектировали и разработали современное веб-приложение, которое позволяет ученикам и преподавателям эффективно взаимодействовать друг с другом, не тратя время на излишние процессы.

Во время работы мы применили методологии структурного и объектно-ориентированного проектирования информационных систем, что позволило создать эффективное и масштабируемое решение. Были проанализированы существующие методы разработки и выбраны наиболее подходящие из них.

Тестирование показало, что веб-приложение в полной мере соответствует всем заявленным требованиям, а именно:

- интуитивный интерфейс;
- быстрое действие;
- надёжность.

Данное веб-приложение станет отличной основой для создания более интерактивной платформы, которая станет флагманом и отличительной чертой нашего продукта.

Список используемой литературы

1. Веб-фреймворк Django (Python) - MDN Web Docs [Электронный ресурс]. URL: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django> (дата обращения: 15.09.2023).
2. Грекул В. И., Денищенко Г. Н., Коровкина Н. Л. Проектирование информационных систем [Электронный ресурс]: учебное пособие. Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 299 с. URL: <https://www.iprbookshop.ru/97577.html> (дата обращения: 20.07.2022).
3. Грекул В. И., Коровкина Н. Л., Куприянов Ю. В. Методические основы управления ИТ-проектами : учебник. Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. 467 с. URL: <https://www.iprbookshop.ru/102019.html> (дата обращения: 20.07.2022).
4. Каскад и наследование - Изучение веб-разработки | MDN [Электронный ресурс]. URL: https://developer.mozilla.org/ru/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance (дата обращения: 15.09.2023).
5. Краткое описание нотации BPMN [Электронный ресурс]. URL: <https://habr.com/ru/companies/auriga/articles/667084/> (дата обращения: 16.09.2023).
6. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose [Электронный ресурс] : учебное пособие. М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 317 с. URL: <https://www.iprbookshop.ru/97554.html> (дата обращения: 15.07.2022).
7. Лучшие IDE и редакторы кода для Python [Электронный ресурс]. URL: <https://tproger.ru/translations/python-ide> (дата обращения: 15.09.2023).

8. Механизмы управления: учебное пособие / под ред. Д. А. Новикова. М.: УРСС (Editorial URSS), 2011.

9. Мкртычев С.В., Гущина О.М., Очеповский А.В. Прикладная информатика. Бакалаврская работа [Электронный ресурс] : электрон. учебметод. пособие. Тольятти. ТГУ: Изд-во ТГУ, 2019. URL: <https://dspace.tltsu.ru/handle/123456789/8868> (дата обращения: 10.07.2022)

10. Начало работы с HTML - Изучение веб-разработки | MDN [Электронный ресурс]. URL: https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/Getting_started (дата обращения: 15.09.2023).

11. Пирогов В. Ю. Информационные системы и базы данных. Организация и проектирование / В. Ю. Пирогов. – СПб.: «БХВ-Петербург», 2020. – 254 с.

12. Программа Ramus [Электронный ресурс]. URL: <https://ru.freownloadmanager.org/Windows-PC/Ramus-Educational.html> (дата обращения: 03.09.2022).

13. Требования к системе: классификация FURPS+ [Электронный ресурс]. URL: <https://sysana.wordpress.com/2010/09/16/furps/> (дата обращения: 15.09.2023).

14. Трутнев Д. Р. Архитектуры информационных систем. Основы проектирования: Учебное пособие. СПб.: НИУ ИТМО, 2012. 66 с.

15. Что такое Bootstrap, и как он работает - Hexlet Guides [Электронный ресурс]. URL: <https://guides.hexlet.io/ru/bootstrap/> (дата обращения: 15.09.2023).

16. Django Channels – ответ современному вебу [Электронный ресурс]. URL: <https://habr.com/ru/companies/oleg-bunin/articles/418445/> (дата обращения: 15.09.2023).

17. PyCharm — IDE для веб-разработки на Python [Электронный ресурс]. URL: <https://www.jetbrains>

18. Python - Высокоуровневый язык программирования [Электронный ресурс]. URL: <https://sprweb.ru/technologies/python/> (дата обращения: 15.09.2023).

19. UML Component Diagrams [Электронный ресурс]. URL: <https://www.uml-diagrams.org/component-diagrams.html> (дата обращения: 10.07.2022).

20. WebSocket (WSS): что это и как работают сокеты [Электронный ресурс]. URL: <https://blog.skillfactory.ru/glossary/websocket/> (дата обращения: 15.09.2023).

Приложение А

Основные модели пользователя, учебного курса и урока

```
from django.contrib.auth.models import AbstractUser
from django.conf import settings
from django.db import models
from django.core.validators import FileExtensionValidator

class CustomUser(AbstractUser):
    TEACHER = 'teacher'
    STUDENT = 'student'
    CATEGORY_CHOICES = [
        (TEACHER, 'Teacher'),
        (STUDENT, 'Student'),
    ]
    category = models.CharField(max_length=20, choices=CATEGORY_CHOICES)
class Course(models.Model):
    title = models.CharField(max_length=100)
    description = models.TextField()
    def __str__(self):
        return self.title
class Lesson(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
    course = models.ForeignKey('Course', on_delete=models.CASCADE)
    teacher = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.CASCADE)
    students = models.ManyToManyField(settings.AUTH_USER_MODEL,
related_name='lessons', blank=True)
    lesson_recording = models.FileField(upload_to='lesson_recordings/',
validators=[FileExtensionValidator(['mp4', 'avi', 'mov', 'flv']), blank=True)
    lesson_materials = models.FileField(upload_to='lesson_materials/',
validators=[FileExtensionValidator(['pdf', 'doc', 'docx', 'ppt', 'pptx', 'xls', 'xlsx']), blank=True)

    def __str__(self):
        return self.title
```

Шаблон формы создания урока

```
from django import forms
from .models import Lesson
from .models import CustomUser

class LessonForm(forms.ModelForm):
    class Meta:
        model = Lesson
        fields = ['title', 'content', 'course', 'students', 'lesson_recording', 'lesson_materials']
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['students'].queryset = CustomUser.objects.filter(category='student')
```

Приложение Б

Настройка админ-панели

```
from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .models import CustomUser, Course, Lesson

class CustomUserAdmin(UserAdmin):
    fieldsets = (
        (None, {'fields': ('username', 'password')}),
        ('Personal info', {'fields': ('first_name', 'last_name', 'email')}),
        ('Permissions', {'fields': ('is_active', 'is_staff', 'is_superuser', 'groups', 'user_permissions')}),
        ('Important dates', {'fields': ('last_login', 'date_joined')}),
        ('Custom fields', {'fields': ('category',)}), # Добавленное поле для статуса пользователя
    )
    add_fieldsets = (
        (None, {
            'classes': ('wide',),
            'fields': ('username', 'password1', 'password2', 'category'), # Добавленное поле для
статуса пользователя
        }),
    )
    list_display = ('username', 'email', 'first_name', 'last_name', 'is_staff', 'category') #
Отображаем статус пользователя в списке пользователей

admin.site.register(CustomUser, CustomUserAdmin)

admin.site.register(Course)
admin.site.register(Lesson)
```

Настройка путей

```
from django.urls import path
from . import views
from django.contrib.auth import views as auth_views
from . import consumers
urlpatterns = [
    path("", views.home, name='home'),
    path('courses/', views.course_list, name='course_list'),
    path('courses/<int:course_id>/lessons/', views.lesson_list, name='lesson_list'),
    path('lessons/create/', views.create_lesson, name='create_lesson'),
    path('courses/<int:course_id>/', views.course_detail, name='course_detail'),
    path('lessons/<int:lesson_id>/', views.lesson_detail, name='lesson_detail'),
    path('login/', auth_views.LoginView.as_view(), name='login'),
    path('ws/chat/', consumers.ChatConsumer.as_asgi()),
]
```


Приложение В

Настройка обработчика сообщений в чате

```
from channels.generic.websocket import AsyncWebsocketConsumer
import json
```

```
class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope['url_route']['kwargs']['room_name']
        self.room_group_name = 'chat_%s' % self.room_name

        # Join room group
        await self.channel_layer.group_add(
            self.room_group_name,
            self.channel_name
        )

        await self.accept()

    async def disconnect(self, close_code):
        # Leave room group
        await self.channel_layer.group_discard(
            self.room_group_name,
            self.channel_name
        )

    # Receive message from WebSocket
    async def receive(self, text_data):
        text_data_json = json.loads(text_data)
        message = text_data_json['message']

        # Send message to room group
        await self.channel_layer.group_send(
            self.room_group_name,
            {
                'type': 'chat_message',
                'message': message
            }
        )

    # Receive message from room group
    async def chat_message(self, event):
        message = event['message']

        # Send message to WebSocket
        await self.send(text_data=json.dumps({
            'message': message
        })))
```

Приложение Г

Основные представления (функции обработки)

```
from django.shortcuts import render, redirect, get_object_or_404
from .forms import LessonForm
from .models import Course
from .models import Lesson

def create_lesson(request):
    if request.method == 'POST':
        form = LessonForm(request.POST, request.FILES)
        if form.is_valid():
            lesson = form.save(commit=False)
            lesson.teacher = request.user
            lesson.save()
            return redirect('lesson_detail', lesson_id=lesson.pk)
    else:
        form = LessonForm()
    return render(request, 'create_lesson.html', {'form': form})

def course_list(request):
    courses = Course.objects.all()
    return render(request, 'course_list.html', {'courses': courses})

def course_detail(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    lessons = course.lesson_set.all() # Извлекаем все уроки для этого курса
    return render(request, 'course_detail.html', {'course': course, 'lessons': lessons})

def lesson_list(request, course_id):
    course = get_object_or_404(Course, pk=course_id)
    lessons = course.lesson_set.all()
    return render(request, 'lesson_list.html', {'course': course, 'lessons': lessons})

def lesson_detail(request, lesson_id):
    lesson = get_object_or_404(Lesson, pk=lesson_id)
    return render(request, 'lesson_detail.html', {'lesson': lesson})

def home(request):
    current_user = request.user
    if current_user.is_authenticated:
        current_user_role = current_user.category
    else:
        current_user_role = None
    return render(request, 'home.html', {'user': current_user})
```

Приложение Д

Скрипт чата

```
<script>
  const chatSocket = new WebSocket(
    'ws://' + window.location.host + '/ws/chat/'
  );

  // Function to display messages on the page
  function displayMessage(message) {
    const messagesDiv = document.getElementById('chat-messages');
    messagesDiv.innerHTML += '<p>' + message + '</p>';
  }

  // Event handler for receiving messages from the server
  chatSocket.onmessage = function (e) {
    const data = JSON.parse(e.data);
    // Display the received message on the page
    displayMessage(data.message);
  };

  // Function to send a message to the server
  function sendMessage() {
    const messageInput = document.getElementById('chat-input');
    const message = messageInput.value;
    // Send the message to the server in JSON format
    chatSocket.send(JSON.stringify({
      'message': message
    }));
    // Clear the input field after sending the message
    messageInput.value = "";
  }
</script>
```