

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра программного обеспечения информационных технологий

Номер зачетной книжки 15100036

Производственная практика зачтена с оценкой

_____ (_____)

(цифрой) (прописью)

(подпись руководителя практики от БГУИР)

_____._____.2024

ОТЧЕТ

по преддипломной практике

Место прохождения практики: ОАО «Беларуськалий»

Сроки прохождения практики: с 10.02.2025 по 23.03.2025

Руководитель практики от
предприятия:
_____ Невмержицкий А.Н.
(подпись руководителя)
М.П.

Студент группы 151004
_____ Дубовский А.В.
(подпись студента)
Руководитель практики от БГУИР
Мелких Е.Г. - старший преподава-
тель, доцент.

Минск 2025

СОДЕРЖАНИЕ

Введение	3
1 Аналитический обзор литературных источников и формирование требований к проектируемому программному средству.....	4
1.1 Анализ прототипов	4
1.2 Сравнение прототипов.....	10
1.3 Формирование требований к проектируемому ПС	14
2 Моделирование предметной области.....	17
2.1 Моделирование программного обеспечения.....	17
2.2 Инфологическая модель базы данных	18
2.3 Спецификация функциональных требований	22
3 Проектирование Программного средства	24
3.1 Выбор архитектуры для разработки.....	24
3.2 Логическая модель базы данных	25
3.3 Физическая модель базы данных	31
3.4 Разработка алгоритма ПС и отдельных модулей.....	31
4 Создание программного средства	37
4.1 Выбор инструментов разработки	37
5 Тестирование, проверка работоспособности и анализ полученных результатов.....	40
6 Руководство по установке и использованию	41
6.1 Развертывание серверной части	41
6.2 Установка клиентского части	43
6.3 Руководство по использованию.....	43
Заключение	44
Список использованных источников	45

ВВЕДЕНИЕ

Электронные дневники играют важную роль в современном образовании, обеспечивая удобное взаимодействие между учениками, учителями и родителями. В условиях быстрого развития информационных технологий автоматизация учета успеваемости и организации учебного процесса становится всё более востребованной. Эти цифровые платформы заменяют традиционные бумажные носители, упрощая доступ к информации и повышая прозрачность образовательного процесса.

Актуальность разработки электронных дневников обусловлена недостаточной развитостью программных решений для образовательных учреждений. Это открывает возможности для создания специализированных инструментов, соответствующих современным требованиям. Разрабатываемое программное обеспечение направлено на автоматизацию управления учебным процессом и взаимодействия между его участниками. Внедрение системы повысит уровень цифровизации образования.

Электронные дневники позволяют учителям оперативно выставять оценки, фиксировать пропуски и назначать задания. Родители получают возможность в любое время отслеживать успеваемость своих детей, а ученикам получать обратную связь от преподавателей.

Кроме того, такие системы способствуют систематизации данных об успеваемости, что помогает анализировать результаты обучения и выявлять тенденции.

Таким образом, электронные дневники не только облегчают работу учителей, но и расширяют возможности родителей и учеников, способствуя более активному вовлечению всех участников образовательного процесса и повышая его качество.

Цель работы – разработка мобильного приложения «Электронный дневник» для образовательных учреждений с использованием технологии .NET MAUI.

Для достижения цели необходимо выполнить следующие задачи:

- Провести анализ существующих решений и определить их преимущества и недостатки.
- Разработать требования к системе и её основные функциональные возможности.
- Спроектировать архитектуру приложения и базы данных.
- Реализовать систему с поддержкой мобильных устройств.
- Оценить эффективность разработанного программного обеспечения.

1 АНАЛИТИЧЕСКИЙ ОБЗОР ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОМУ ПРОГРАММНОМУ СРЕДСТВУ

1.1 Анализ прототипов

Анализ существующих аналогичных решений позволяет не только выявить тенденции на рынке образовательных приложений, но и формирует ясные требования для разработки мобильного приложения для электронного дневника, ориентированного на улучшение пользовательского опыта и эффективное управление учебным процессом.

1.1.1 School.by

School.by — это белорусская платформа, предназначенная для цифровизации образовательного процесса в школах и других учебных заведениях. Она обеспечивает эффективное управление успеваемостью учащихся, улучшает коммуникацию между учителями, учениками и родителями, а также способствует организации учебного процесса в электронном формате. [1][2]

Основные особенности:

- Электронный журнал и дневник: Учителя имеют возможность вводить и обновлять оценки, задания и другую учебную информацию, доступную для учеников и их родителей.
- Аналитика успеваемости: Платформа предоставляет графики и таблицы успеваемости для каждого класса и ученика, что позволяет отслеживать динамику обучения.
- Встроенный чат для коммуникации: Ученики и родители могут общаться с учителями и школьной администрацией через встроенный чат, что способствует оперативному решению возникающих вопросов. □
- SMS-уведомления о посещаемости: Родители получают сообщения о пропусках учеников, что помогает своевременно реагировать на проблемы с посещаемостью.
- Настраиваемый дизайн школьного портала: Возможность редактирования дизайна страниц школы, включая цветовое оформление и графическую структуру, позволяет адаптировать платформу под индивидуальные потребности учебного заведения.

Преимущества:

- Удобство и простота использования: Интуитивно понятный интерфейс облегчает работу для учителей, учеников и родителей.
- Повышенная вовлеченность участников: Оперативная обратная связь и доступ к информации способствуют активному участию всех участников образовательного процесса.

– Интеграция с онлайн-ресурсами: Платформа поддерживает объединение с различными сервисами для онлайн-школ, расширяя функциональные возможности.

– Мобильный доступ: Доступность платформы через мобильные устройства позволяет использовать ее в любое время и в любом месте.

Недостатки:

– Ограниченные аналитические функции: Платформа предоставляет базовые отчеты, но не поддерживает глубокий анализ успеваемости и динамики учащихся.

– Ограниченная настройка отчетности: Система отчетности может не удовлетворять специфические потребности некоторых учебных заведений.

– Ограниченная интеграция с внешними системами: Несмотря на возможность интеграции, некоторые внешние сервисы могут быть несовместимы или требовать дополнительной настройки.

– Ограниченная гибкость настройки интерфейса: Ограниченные возможности настройки интерфейса и персонализации могут не соответствовать ожиданиям всех пользователей.

School.by эффективная платформа для управления образовательным процессом, которая способствует улучшению коммуникации между учителями, учениками и родителями. Несмотря на некоторые ограничения в аналитике и настройках, она предоставляет широкий спектр инструментов для автоматизации учебного процесса и повышения его качества.

1.1.2 MyClassroom

MyClassroom — это образовательная платформа, разработанная для улучшения взаимодействия между учителями и учениками в условиях дистанционного, гибридного и смешанного обучения. Она предоставляет ряд уникальных инструментов, направленных на повышение эффективности учебного процесса.[3][4]

Основные особенности:

– Универсальность и доступность: MyClassroom работает на любом устройстве через веб-браузер, позволяя преподавать до 60 студентам одновременно.

– Интерактивное управление классом: Платформа предлагает инструменты для упрощения управления классом, включая учет посещаемости, совместную работу и удержание внимания учащихся.

– Групповая работа: Функция групп позволяет учащимся общаться в чате, взаимодействовать с доской своей группы и представлять свою работу классу, что способствует развитию сотрудничества и коммуникации.

– Интеграция с Google Classroom: MyClassroom легко интегрируется с

Google Classroom, что обеспечивает эффективное управление занятиями и совместимость с другими образовательными сервисами.

- Интерактивные учебные материалы: Платформа предоставляет доступ к библиотеке готовых интерактивных материалов, планов уроков и обучающего контента, что облегчает подготовку и проведение занятий.

- Образовательный ассистент Companion: MyClassroom включает виртуального помощника Companion, который помогает преподавателям и учащимся в организации учебного процесса, предоставляя полезные рекомендации и поддержку.

Преимущества:

- Упрощение управления классом: Инструменты платформы облегчают организацию учебного процесса, включая управление посещаемостью и стимулирование участия учащихся.

- Стимулирование групповой работы: Функции групповой работы способствуют развитию сотрудничества и коммуникации среди учащихся.

- Интеграция с популярными сервисами: Совместимость с Google Classroom и другими сервисами обеспечивает гибкость и расширенные возможности для преподавателей и учащихся.

- Доступ к интерактивным ресурсам: Библиотека учебных материалов и планов уроков помогает преподавателям эффективно готовить и проводить занятия.

- Поддержка виртуального помощника: Виртуальный ассистент Companion предоставляет рекомендации и поддержку, улучшая организацию учебного процесса.

Недостатки:

- Ограничения по количеству студентов: Платформа поддерживает до 60 студентов одновременно, что может быть недостаточно для крупных учебных заведений.

- Необходимость в обучении персонала: Для эффективного использования всех функций платформы может потребоваться дополнительное обучение преподавателей и технического персонала.

MyClassroom — это мощная и гибкая платформа, предназначенная для улучшения взаимодействия между учителями и учениками в различных форматах обучения. Благодаря широкому спектру инструментов и интеграций, она способствует повышению эффективности учебного процесса. Однако перед внедрением рекомендуется учитывать возможные ограничения и подготовить персонал к работе с новой системой.

1.1.3 Edmodo

Edmodo — это образовательная социальная сеть, предназначенная для взаимодействия между учителями, учениками и их родителями. Платформа предоставляет инструменты для обмена учебным контентом, общения в режиме реального времени и управления учебным процессом.[5][6]

Основные особенности:

- Обмен учебным контентом: Учителя могут публиковать задания, тесты и учебные материалы, доступные для учеников и их родителей.
- Коммуникация в реальном времени: Платформа поддерживает обмен сообщениями, что облегчает оперативное решение учебных вопросов.
- Управление классами: Edmodo позволяет организовывать учебный процесс, отслеживать успеваемость и взаимодействовать с учениками и их родителями.

Преимущества:

- Безопасность: Платформа обеспечивает защищенную среду для общения и обмена информацией между участниками образовательного процесса.
- Удобство использования: Интуитивно понятный интерфейс облегчает доступ к учебным материалам и коммуникацию между пользователями.
- Интеграция с другими сервисами: Edmodo поддерживает совместную работу с различными образовательными инструментами, расширяя возможности обучения.

Недостатки:

- Ограничения функционала: Некоторые пользователи отмечают недостаток расширенных функций для анализа успеваемости и настройки интерфейса.

Edmodo продолжает оставаться популярным инструментом в сфере образования, предлагая широкий спектр функций для эффективного управления учебным процессом. Несмотря на некоторые ограничения, платформа предоставляет ценную поддержку для учителей, учеников и родителей, способствуя улучшению образовательного опыта.

1.1.4 Google Classroom

Google Classroom — это бесплатная образовательная платформа от Google, предназначенная для упрощения процесса создания, распространения и оценки заданий без использования бумаги. [7][8]

Основные особенности:

- Интеграция с продуктами Google: Classroom объединяет приложения, такие как Google Диск, Документы, Таблицы, Презентации, Формы и Почта, обеспечивая удобство работы в одном пространстве.
- Управление заданиями: Учителя могут создавать задания с различными параметрами доступа, устанавливать сроки выполнения и предоставлять обратную связь, что способствует эффективному обучению.
- Коммуникация: Платформа поддерживает обмен сообщениями между учителями и учениками, а также возможность публиковать объявления и вести обсуждения в классе.
- Мобильный доступ: Доступность мобильных приложений для iOS и Android позволяет использовать Classroom на различных устройствах, обеспечивая гибкость в обучении.

Преимущества:

- Удобство использования: Интуитивно понятный интерфейс и интеграция с другими сервисами Google делают платформу доступной для пользователей с разным уровнем подготовки.
- Организация учебного процесса: Classroom помогает систематизировать задания, отслеживать успеваемость и упрощает взаимодействие между всеми участниками образовательного процесса.

Недостатки:

- Ограниченная поддержка стандартов электронного обучения: Отсутствие поддержки SCORM, Tin Can (xAPI) и cmi5 может затруднить использование некоторых интерактивных курсов.
- Отсутствие встроенной вебинарной комнаты: Для проведения онлайн-занятий требуется использование дополнительных сервисов, таких как YouTube или Google Hangouts.
- Ограничения для бесплатных пользователей: В бесплатной версии сервиса количество участников курса ограничено 200 человек, что может быть недостаточно для крупных образовательных учреждений.

Google Classroom является мощным инструментом для организации учебного процесса, предлагая интеграцию с продуктами Google, эффективное управление заданиями и удобные коммуникационные возможности. Однако перед его использованием следует учитывать ограничения, связанные с поддержкой стандартов и функциональными возможностями, чтобы определить его соответствие специфическим потребностям образовательного учреждения.

1.1.5 ClassDojo

ClassDojo — это бесплатная образовательная платформа, предназначенная для содействия сотрудничеству между учителями, учениками и их семьями. Сервис поддерживает социально-эмоциональное обучение, предоставляя инструменты для обратной связи, создания портфолио и коммуникации.[9][10]

Основные особенности:

- Обратная связь через баллы: Учителя могут присуждать ученикам баллы за различные достижения и поведение, что способствует развитию положительных навыков и привычек.
- Портфолио учеников: ClassDojo предоставляет возможность ученикам создавать персональные портфолио, где они могут демонстрировать свои работы и достижения, а также получать обратную связь от учителей и родителей.
- Коммуникация с родителями: Платформа упрощает общение между учителями и родителями, позволяя отправлять уведомления о достижениях и проблемах учеников, а также получать обратную связь.
- Настройка ценностей сообщества: ClassDojo позволяет учителям устанавливать и отслеживать определённые ценности и ожидания в классе, способствуя формированию сплочённой и позитивной учебной среды.

Преимущества:

- Удобство использования: Сервис обладает интуитивно понятным интерфейсом и доступен на нескольких языках, включая русский, что облегчает его использование учителями, учениками и родителями.
- Гибкость и адаптивность: ClassDojo можно настроить в соответствии с потребностями конкретного класса или школы, позволяя учителям индивидуализировать подход к обучению и воспитанию.
- Доступность на мобильных устройствах: Платформа имеет мобильное приложение, что позволяет учителям и родителям оставаться на связи и отслеживать прогресс учеников в любое время и в любом месте.

Недостатки:

- Зависимость от интернет-соединения: Для использования ClassDojo требуется стабильное интернет-соединение, что может быть проблемой в районах с ограниченным доступом к интернету.
- Ограничения функциональности: Некоторые пользователи отмечают, что функциональность мобильного приложения ClassDojo может быть ограничена по сравнению с веб-версией, что может затруднить использование платформы на мобильных устройствах.

ClassDojo является эффективным инструментом для поддержки социально-эмоционального обучения и улучшения коммуникации между учителями, учениками и их семьями. Несмотря на некоторые ограничения, такие как зависимость от интернет-соединения и функциональные ограничения мобильного приложения, ClassDojo предоставляет множество возможностей для создания позитивной и продуктивной учебной среды

1.2 Сравнение прототипов

Таблица 1.2.1 – Сравнительная характеристика прототипов

Характеристика	School.by	MyClassroom	Edmodo	Google Classroom	ClassDojo
Целевая аудитория	Школы и колледжи Беларуси	Учителя и ученики по всему миру	Учителя, ученики и родители	Учебные заведения всех уровней	Учителя, ученики и их семьи
Тип платформы	Веб-сервис и мобильные приложения	Веб-сервис и мобильные приложения	Веб-сервис и мобильные приложения	Веб-сервис и мобильные приложения	Веб-сервис и мобильные приложения
Основной функционал	Электронный дневник, расписание, домашние задания, оценки, уведомления	Виртуальные классы, расписание, тесты, обмен сообщениями	Электронный дневник, задания, тесты, форумы	Расписание, задания, тесты, аналитика	Баллы, портфолио, связь с родителями
Интеграция с другими сервисами	Ограниченная интеграция с внешними сервисами	Интеграция с Google, Microsoft, Zoom	Интеграция с Google, Microsoft, внешними сервисами	Полная интеграция с Google (Docs, Drive и т.д.)	Интеграция с Google и другими сервисами
Поддержка мобильных платформ	iOS, Android	iOS, Android	iOS, Android	iOS, Android	iOS, Android

Продолжение таблицы 1.2.1 – Сравнительная характеристика прототипов

Возможность офлайн-доступа	Нет	Частично (ограниченный функционал)	Нет	Частично (ограниченный функционал)	Частично (ограниченный функционал)
Поддержка мультиязычности	Русский, белорусский	Многоязычная поддержка	Многоязычная поддержка	Многоязычная поддержка	Многоязычная поддержка
Аналитика и отчётность	Базовая аналитика, ограниченные отчёты	Расширенная аналитика, визуализация данных	Ограниченная аналитика, базовые отчёты	Расширенная аналитика, интеграция с Google Sheets	Базовая аналитика, отчёты по поведению
Безопасность данных	Соответствие национальным стандартам	Шифрование данных, соответствие международным стандартам	Шифрование данных, соответствие международным стандартам	Шифрование данных, соответствие международным стандартам	Шифрование данных, соответствие международным стандартам
Стоимость использования	Бесплатно для государственных учреждений	Бесплатная и платная версии	Бесплатная и платная версии	Бесплатно	Бесплатно с дополнительными платными функциями
Поддержка пользователей	Электронная почта, телефон	Онлайн-чат, база знаний	Форумы, база знаний	Форумы, база знаний	Онлайн-чат, база знаний
Уровень персонализации интерфейса	Ограниченная настройка	Расширенная настройка	Ограниченная настройка	Ограниченная настройка	Расширенная настройка

Продолжение таблицы 1.2.1 – Сравнительная характеристика прототипов

Под- держка стандар- тов элек- тронного обучения (SCORM, xAPI)	Нет	Частично	Нет	Нет	Нет
Наличие встроен- ного виде- очата	Нет	Да	Нет	Нет	Нет
Под- держка соци- ально- эмоцио- нального обучения	Нет	Нет	Нет	Нет	Да
Наличие виртуаль- ного ассис- тента	Нет	Да (Companio n)	Нет	Нет	Нет
Под- держка геймифи- кации	Нет	Частично	Нет	Нет	Да
Наличие API для разработ- чиков	Нет	Да	Нет	Да	Нет
Регуляр- ность об- новлений	Нерегу- лярные	Регуляр- ные	Регуляр- ные	Регуляр- ные	Регуляр- ные
Под- держка обратной связи от пользова- телей	Да	Да	Да	Да	Да

Продолжение таблицы 1.2.1 – Сравнительная характеристика прототипов

Уровень адаптации под различные образовательные системы	Высокий для Беларуси	Средний	Средний	Высокий	Средний
Возможность настройки уведомлений	Да	Да	Да	Да	Да
Возможность экспорта данных	Ограниченная	Да	Ограниченная		Ограниченная
Поддержка форумов и обсуждений	Нет	Да	Да	Да	Нет
Возможность настройки прав доступа	Ограниченная	Да	Ограниченная	Да	Ограниченная
Поддержка уведомлений в реальном времени	Да	Да	Да	Да	Да
Возможность настройки интерфейса под бренд школы	Ограниченная	Да	Ограниченная	Ограниченная	Да

Продолжение таблицы 1.2.1 – Сравнительная характеристика прототипов

Под- держка интегра- ции с со- циаль- ными се- тями	Нет	Да	Да	Да	Да
---	-----	----	----	----	----

1.3 Формирование требований к проектируемому ПС

1.3.1 Назначение разработки

Проектируемое программное средство — это система «Электронный дневник», предназначенная для автоматизации учебного процесса в образовательных учреждениях. Основной задачей системы является упрощение взаимодействия между учениками, родителями, учителями и администрацией школы. Система предоставляет доступ к информации о расписании, успеваемости, домашних заданиях, а также позволяет управлять учебными планами и школьными данными. Она будет способствовать улучшению контроля за успеваемостью учащихся и повышению прозрачности учебного процесса.

1.3.2 Состав выполняемых функций

Состав выполняемых функций включает:

- Создание журналов для учителей с возможностью внесения оценок, отметок о пропусках и другой важной информации по каждому учебному предмету.
- Формирование электронных дневников для учеников и родителей с доступом к расписанию, учебным предметам, оценкам и домашним заданиям.
- Возможность добавления школ с уникальными параметрами, такими как расписание, преподаватели и учебные планы.
- Назначение администратора для каждой школы с правами редактирования и управления данными школы.
- Автоматический расчет среднего балла ученика за учебную четверть.
- Реализация профилей пользователей с различным доступом в зависимости от их роли (ученик, родитель, учитель, администратор).
- Привязка родителей к ученикам для доступа к электронному дневнику.
- Создание групп и подгрупп для раздельного проведения уроков.
- Отображение общей информации о школах, в том числе новостей и объявлений.
- Возможность публикации школьных новостей, мероприятий и объявлений.

- Реализация функции комментирования новостей для взаимодействия между пользователями.
- Организация личных сообщений между участниками системы для обмена информацией.

1.3.3 Входные данные

Система будет принимать следующие данные:

- Информация о пользователях: имя, роль (ученик/родитель/учитель/администратор), школа, классы, преподаваемые предметы (для учителей), контактные данные, аватар профиля.
- Данные о расписании уроков: дата, время, аудитория, преподаватель.
- Оценки учеников: баллы, дата выставления, ID преподавателя.
- Отметки о пропусках: дата.
- Домашние задания: текст задания, ID класса.
- Новости и объявления: заголовок, текст, дата публикации, ID автора.
- Личные сообщения: текст, ID отправителя и получателя, метка времени.
- Комментарии: текст, ID автора, ID новости/профиля, дата публикации.
- Системные данные: логи аудита (изменения оценок, правки расписания), пароли, логины.

1.3.4 Выходные данные

Система будет генерировать:

- Электронные дневники: оценки, домашние задания, уведомления о событиях (для учеников и родителей).
- Журналы учителей: таблицы с оценками, отметки о пропусках, возможность редактирования данных.
- Отчеты об успеваемости: средний балл ученика, динамика оценок, статистика посещаемости.
- Личные сообщения: история переписок, уведомления о новых сообщениях.
- Комментарии: текстовые сообщения под новостями и профилями пользователей.
- Новости и объявления: заголовок, текст, дата публикации, автор.
- Сводная информация: актуальное расписание, изменения в учебном процессе, активность пользователей.

1.3.5 Требования к составу и параметрам технических и программных средств

Для корректной работы системы потребуется:

- Мобильные приложения для платформ Android и iOS с адаптивным

интерфейсом, а также возможность запуска на Windows и macOS.

- Базы данных для хранения информации о пользователях, школьных данных, расписаниях и оценках.

- Операционные системы: Linux/Windows для серверной части, Android/iOS для мобильных устройств, Windows/macOS для возможного запуска на ПК.

- Подключение к интернету для обеспечения синхронизации данных между различными устройствами и пользователями.

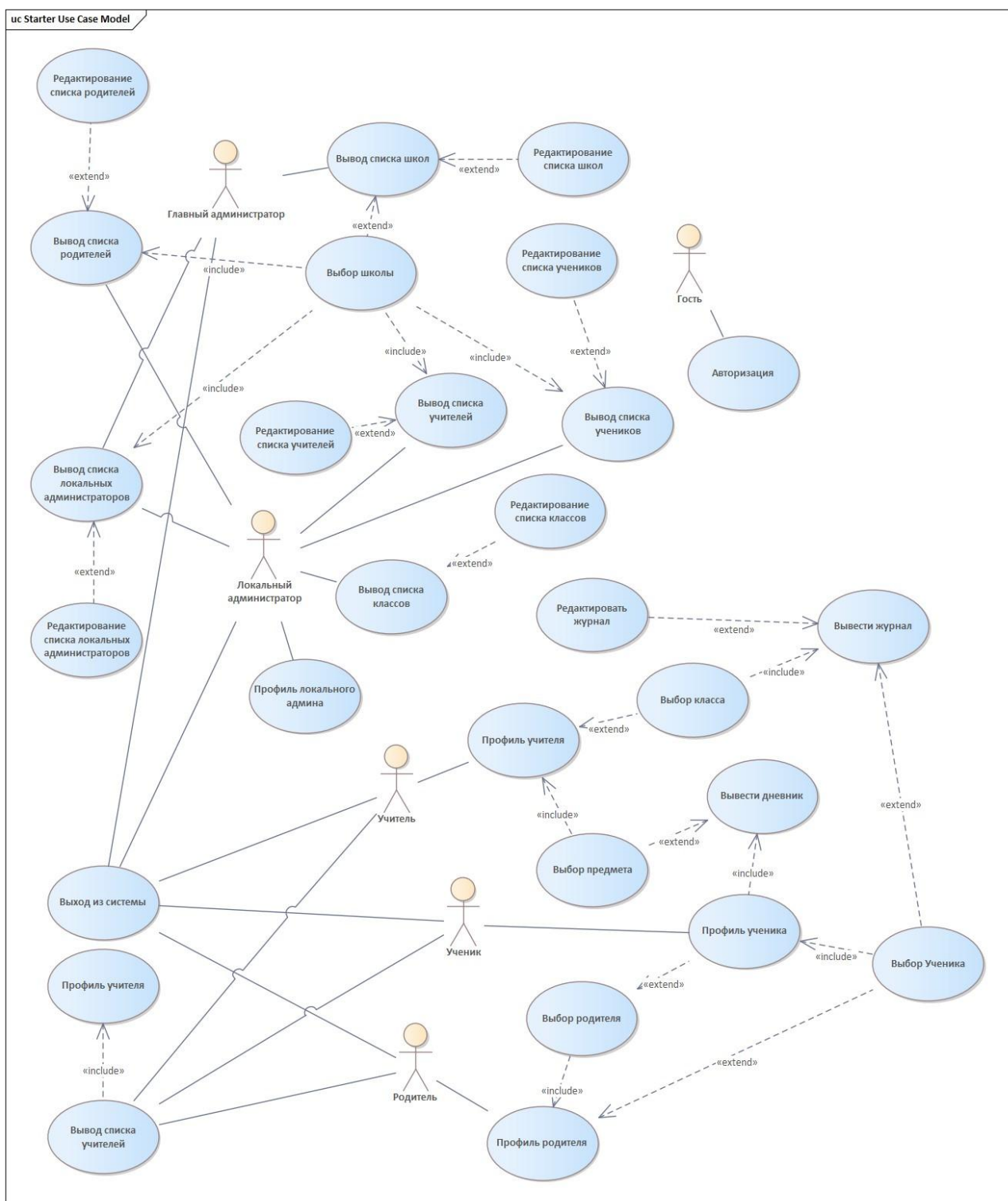
1.3.6 Требования к информационной и программной совместимости

Система должна быть совместима с современными операционными системами и веб-браузерами, включая:

- Операционные системы: Android, iOS, Windows, macOS.

- Обеспечение безопасности данных с использованием хеширования паролей с солью, разграничения доступа по ролям и защиты передаваемой информации.

2.1 Моделирование программного обеспечения



2.2 Инфологическая модель базы данных

2.2.1 Табличное представление

Таблица 2.2.1.1 – Сущности и связи

Сущность	Атрибуты	Связанные сущности Тип связи
Базовые данные пользователей (ELD_USERS)	Логин, Хеш, Соль	
Типы учебных заведений (ELD_EDUCATIONAL_INSTITUTIONS_TYPES)	Название типа	Учебные заведения (ELD_EDUCATIONAL_INSTITUTIONS)
Регионы (ELD_REGIONS)	Название региона	Населенные пункты (ELD_SETTLEMENTS)
Населенные пункты (ELD_SETTLEMENTS)	Название	Учебные заведения (ELD_EDUCATIONAL_INSTITUTIONS)
Учебные заведения (ELD_EDUCATIONAL_INSTITUTIONS)	Адрес, email, название, путь к изображению, телефон	Типы учебных заведений (ELD_EDUCATIONAL_INSTITUTIONS_TYPES), Населенные пункты (ELD_SETTLEMENTS)
Администраторы (ELD_ADMINISTRATORS)	email, имя, фамилия, путь к изображению, отчество, телефон	Учебные заведения (ELD_EDUCATIONAL_INSTITUTIONS), Типы пользователей (ELD_USERS_TYPES)
Учителя (ELD_TEACHERS)	email, имя, фамилия, путь к изображению, отчество, телефон	Учебные заведения (ELD_EDUCATIONAL_INSTITUTIONS), Назначения (ELD_TEACHER_ASSIGNMENTS)
Классы (ELD_CLASSES)	Название	Учителя (ELD_TEACHERS)
Школьные предметы (ELD_SCHOOL_SUBJECTS)	Название предмета	Назначения учителей (ELD_TEACHER_ASSIGNMENTS)

Продолжение таблицы 2.2.1.1 – Сущности и связи

Назначения учителей (ELD_TEACHER_ASSIGNMENTS)		Учителя (ELD_TEACHERS), Классы (ELD_CLASSES), Школьные предметы (ELD_SCHOOL_SUBJECTS), Расписание (ELD_SCHEDULE)
Группы (ELD_GROUPS)		Назначения учителей (ELD_TEACHER_ASSIGNMENTS), Ученики (ELD_SCHOOL_STUDENTS)
Дни журнала (ELD_GRADEBOOK_DAYS)	Дата, описание	Группы (ELD_GROUPS)
Ученики (ELD_SCHOOL_STUDENTS)	email, имя, фамилия, путь к изображению, отчество, телефон	Классы (ELD_CLASSES), Родители (ELD_PARENTS)
Посещаемость (ELD_GRADEBOOK_ATTENDANCES)	Статус (присутствие/отсутствие)	Дни журнала (ELD_GRADEBOOK_DAYS), Ученики (ELD_SCHOOL_STUDENTS)
Оценки (ELD_GRADEBOOK_SCORES)	Оценка	Дни журнала (ELD_GRADEBOOK_DAYS), Ученики (ELD_SCHOOL_STUDENTS)
Участники групп (ELD_GROUP_MEMBERS)		Группы (ELD_GROUPS), Ученики (ELD_SCHOOL_STUDENTS)
Изображения (ELD_IMAGES)	Данные изображения, дата загрузки	Учебные заведения (ELD_EDUCATIONAL_INSTITUTIONS)
Сообщения (ELD_MESSAGES)	Текст, время отправки	Пользователи: адресаты (ELD_USERS), отправители (ELD_USERS)
Новости (ELD_NEWS)	Заголовок, содержание, дата публикации	Пользователи-владельцы (ELD_USERS)

Продолжение таблицы 2.2.1.1 – Сущности и связи

Комментарии к новостям (ELD_NEW_COMMENTS)	Текст, время	Новости (ELD_NEWS), Пользователи (ELD_USERS)
Родители (ELD_PARENTS)	email, имя, фамилия, путь к изображению, отчество, телефон	Ученики (ELD_SCHOOL_STUDENTS), Типы родственных связей (ELD_PARENTS_TYPES)
Типы родственных связей (ELD_PARENTS_TYPES)	Название типа (отец, мать и т.д.)	Родители (ELD_PARENTS)
Оценки за четверть (ELD_QUARTER_SCORES)	Оценка	Предметы (ELD_SCHOOL_SUBJECTS), Ученики (ELD_SCHOOL_STUDENTS)
Связи учеников и родителей (ELD_SCHOOL_STUDENTS_AND_PARENTS)		Ученики (ELD_SCHOOL_STUDENTS), Родители (ELD_PARENTS), Типы родственных связей (ELD_PARENTS_TYPES)
Расписание (ELD_SCHEDULE)	День недели	Уроки (ELD_SCHEDULE_LESSONS)
Уроки в расписании (ELD_SCHEDULE_LESSONS)	Номер урока, предмет урока	Расписание (ELD_SCHEDULE), Назначения учителей (ELD_TEACHER_ASSIGNMENTS)
Комментарии пользователей (ELD_USER_COMMENTS)	Текст, время	Пользователи-отправители (ELD_USERS), пользователи-получатели (ELD_USERS)

Продолжение таблицы 2.2.1.1 – Сущности и связи

Типы пользователей (ELD_USERS_TYPES)	Название (админ, учитель и т.д.)	Администраторы (ELD_ADMINISTRATORS), Учителя (ELD_TEACHERS), Ученики (ELD_SCHOOL_STUDENTS), Родители (ELD_PARENTS)
---	----------------------------------	--

2.2.2 Схематичное представление

Концептуальная инфологическая модель базы данных отражает основные сущности системы и связи между ними. В модели выделены ключевые объекты, такие как пользователи, учебные заведения, журналы успеваемости и учебные дисциплины. Каждая сущность обладает уникальными атрибутами, определяющими ее свойства и взаимоотношения в системе.

Инфологическая модель базы данных представлена на рисунке 2.2.2.1

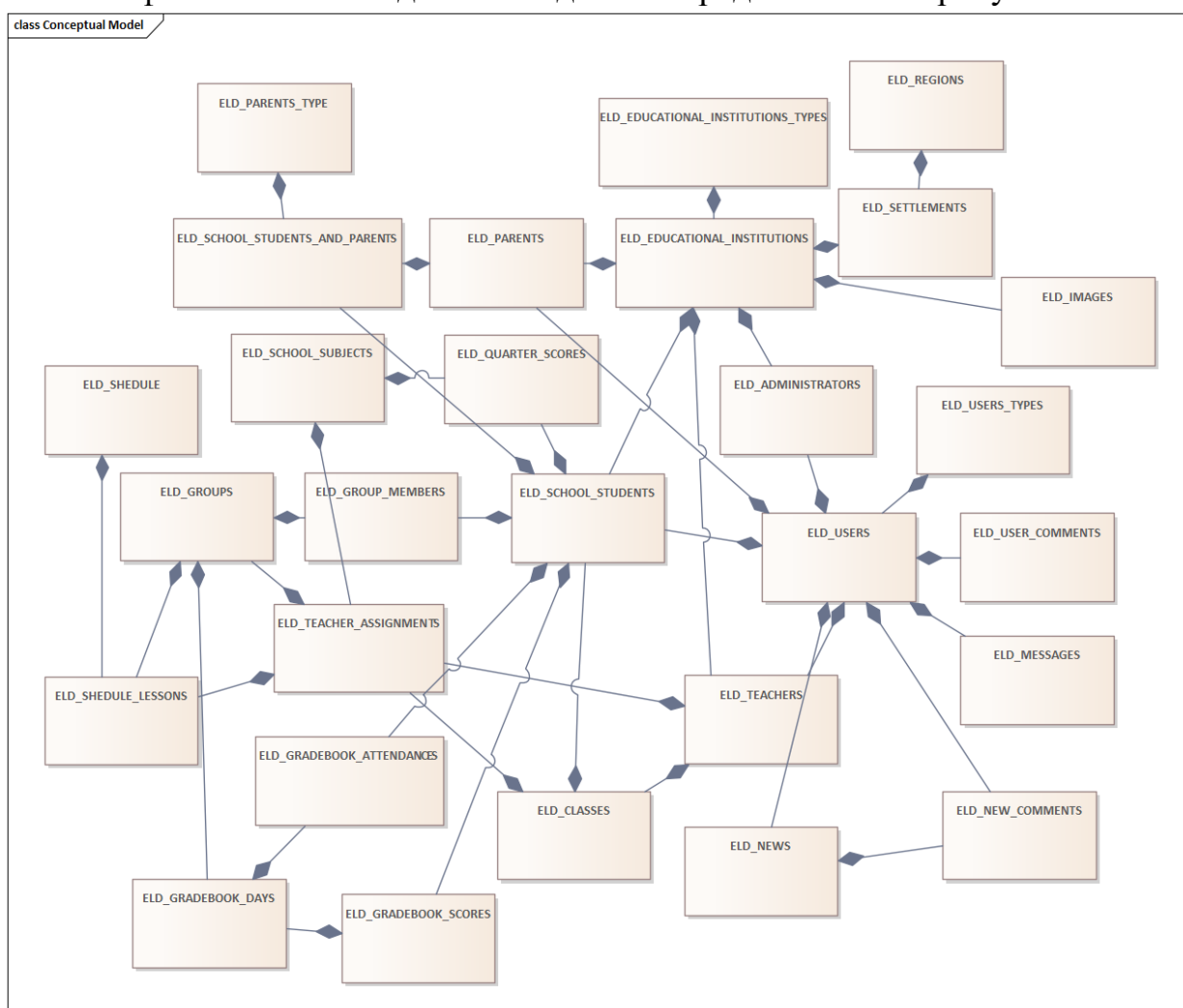


Рисунок 2.2.2.1 — Инфологическая модель базы данных

2.3 Спецификация функциональных требований

2.3.1 Авторизация и ролевой доступ

Спецификация:

- Пользователь входит в систему по логину и паролю.
- Пароль хранится в зашифрованном виде (алгоритм SHA-256 с солью).
- Администратор может сбросить пароль пользователя через панель управления.
- Ролевая модель доступа.

Спецификация ролевой модели доступа:

- Общий функционал: доступ к сообщениям\новостям\комментариям.
- Ученик: просмотр оценок, расписания, домашних заданий.
- Родитель: доступ к данным ребенка.
- Учитель: внесение оценок, управление домашними заданиями.
- Локальный администратор: доступ к определенной школе, управление пользователями, аудит действий.
- Глобальный админ: расширение локального админа с доступом ко всем школам.

2.3.2 Просмотр электронного журнала учителем

Спецификация:

- Учитель должен иметь доступ к журналу с возможностью просмотра всех учеников своего класса.
- Журнал содержит столбцы с датами, выставленными оценками, темами урока, домашними заданиями.
- Ученик и родитель могут просматривать только оценки, выставленные ученику.
- Журнал автоматически рассчитывает средний балл ученика за четверть.
- Данные обновляются в реальном времени при наличии интернет-соединения.

2.3.3 Просмотр электронного дневника

Спецификация:

- Дневник содержит расписание учебных занятий по дням недели.
- В информацию об уроке входят: номер, название предмета, ФИО учителя, кабинет.

2.3.4 Управление пользователями

Спецификация:

- Администратор добавляет пользователей через панель администратора, задавая информацию о пользователе, логин и пароль.
- Администратор изменяет\удаляет пользователей.
- Родительский аккаунт не может существовать без аккаунта ученика.
- При удалении удаляются все связанные сущности.

2.3.5 Управление школами

Спецификация:

- Главный администратор осуществляет добавление\изменение\удаление.
- При удалении удаляются все связанные сущности.
- Администраторы добавляют пользователей с привязкой к школе.

2.3.6 Управление новостями\сообщениями\комментариями

Спецификация:

- Администраторы\учителя создают\редактируют\удаляют новости.
- Любой авторизированный пользователь создаёт комментарий\сообщение.
- Администратор редактирует удаляет сообщения\комментарии.

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

3.1 Выбор архитектуры для разработки

Архитектура системы «Электронный дневник» основана на многоуровневой клиент-серверной модели, обеспечивающей разделение ответственности между компонентами и поддержку кроссплатформенности.

Клиентский уровень:

- Мобильные приложения: интерфейсы для учеников, родителей, учителей и администраторов.
- Функционал: отображение расписания, оценок, домашних заданий; взаимодействие через чаты и уведомления.

Серверный уровень:

- Использование серверной части стороннего разработчика

Уровень данных:

- База данных: реляционная СУБД для хранения структурированной информации (пользователи, школы, оценки, расписания).
- Медиа сервер: получение\хранение\удаление файлов.
- Резервное копирование: обеспечение сохранности данных.

Выбор клиент-серверной архитектуры обоснован ключевыми требованиями проекта:

- Кроссплатформенность — необходимость поддержки мобильных устройств (Android/iOS).
- Централизованное управление данными — единый источник истины для учебных заведений, учеников и родителей.
- Безопасность и масштабируемость — защита персональных данных и возможность роста числа пользователей без деградации производительности.
- Разделение ответственности. Клиент отвечает за взаимодействие с пользователем и конвертацию его действий в единый формат. Сервер отвечает за централизованную обработку данных единого формата независимо от платформы пользователя.

Выбор реляционной СУБД обоснован следующими факторами:

- Целостность данных: Внешние ключи, транзакции и нормализация предотвращают дублирование и противоречия в данных.
- Безопасность: Встроенные механизмы шифрования и разграничения прав доступа соответствуют требованиям GDPR и локальным с

3.2 Логическая модель базы данных

3.2.1 Табличное представление

Таблица 3.2.1.1 – Сущности и связи

Сущности	Атрибуты	Ключи
ELD_USERS	HASH(BYTE[]), ID(INTEGER), LOGIN(String), SALT(BYTE[]), U_UT_ID(INTEGER),	ID первичный ключ U_UT_ID внешний ключ к таблице ELD_USER_TYPES
ELD_REGIONS	ID(INTEGER), NAME(String)	ID первичный ключ S_R_ID внешний ключ к таблице ELD_REGIONS
ELD_SETTLEMENTS	ID(INTEGER), NAME(String), S_R_ID(INTEGER)	ID первичный ключ
ELD_EDUCATIONAL_INSTITUTIONS_TYPES	ID(INTEGER), NAME(String)	ID первичный ключ
ELD_EDUCATIONAL_INSTITUTIONS	ID(INTEGER), ADDRESS(String), EMAIL(String), NAME(String), PATH_IMAGE(BYTE[]), PHONE_NUMBER(String), EI_EIT_ID(INTEGER), EI_S_ID(INTEGER)	ID первичный ключ EI_EIT_ID внешний ключ к таблице ELD_EDUCATIONAL_INSTITUTIONS_TYPES EI_S_ID внешний ключ к таблице ELD_SETTLEMENTS
ELD_CLASSES	ID(INTEGER), NAME(String), C_T_ID(INTEGER)	ID первичный ключ C_T_ID внешний ключ к таблице ELD_TEACHERS
ELD_SCHOOL_SUBJECTS	ID(INTEGER), NAME(String)	ID первичный ключ

Продолжение таблицы 3.2.1.1 – Сущности и связи

ELD_ADMINISTRATORS	ID(INTEGER), EMAIL(STRING), FIRST_NAME(STRING), LAST_NAME(STRING), PATH_IMAGE(BYTE[]), PATRONYMIC(STRING), PHONE_NUMBER(STRING), A_EI_ID(INTEGER), A_U_ID(INTEGER)	ID первичный ключ T_EI_ID внешний ключ к таблице ELD_EDUCATIONAL_INSTITUTIONS T_U_ID внешний ключ к таблице ELD_USERS
ELD_TEACHERS	ID(INTEGER), EMAIL(STRING), FIRST_NAME(STRING), LAST_NAME(STRING), PATH_IMAGE(BYTE[]), PATRONYMIC(STRING), PHONE_NUMBER(STRING), T_EI_ID(INTEGER), T_U_ID(INTEGER)	ID первичный ключ T_EI_ID внешний ключ к таблице ELD_EDUCATIONAL_INSTITUTIONS T_U_ID внешний ключ к таблице ELD_USERS
ELD_TEACHER_ASSIGNMENTS	TA_C_ID(INTEGER), TA_SS_ID(INTEGER), TA_T_ID(INTEGER)	ID первичный ключ TA_C_ID внешний ключ к таблице ELD_CLASSES TA_SS_ID внешний ключ к таблице ELD_SCHOOL_SUBJECTS TA_T_ID внешний ключ к таблице ELD_TEACHERS
ELD_GROUPS	ID(INTEGER), G_TA_ID(INTEGER)	ID первичный ключ G_TA_ID внешний ключ к таблице ELD_TEACHER_ASSIGNMENTS

Продолжение таблицы 3.2.1.1 – Сущности и связи

ELD_GRADEBOOK_DAYS	ID(INTEGER), DATE(DATE) GD_G_ID(INTEGER),	ID первичный ключ GD_G_ID внешний ключ к таблице ELD_GROUPS
ELD_SCHOOL_STUDENTS	ID(INTEGER), EMAIL(String), FIRST_NAME(String), LAST_NAME(String), PATH_IMAGE(BYTE[]), PATRONYMIC(String), PHONE_NUMBER(String), SST_C_ID(INTEGER), SST_EI_ID(INTEGER), SST_U_ID(INTEGER)	ID первичный ключ SST_C_ID внешний ключ к таблице ELD_CLASSES SST_EI_ID внешний ключ к таблице ELD_EDUCATIONAL_INSTITUTIONS SST_U_ID внешний ключ к таблице ELD_USERS
ELD_GRADEBOOK_ATTENDANCES	GA_GD_ID(INTEGER), GA_SST_ID(INTEGER),	ID первичный ключ GA_GD_ID внешний ключ к таблице ELD_GRADEBOOK_DAYS GA_SST_ID внешний ключ к таблице ELD_SCHOOL_STUDENTS
ELD_GRADEBOOK_SCORES	SCORE(INTEGER) GS_GD_ID(INTEGER), GS_SST_ID(INTEGER),	ID первичный ключ GS_GD_ID внешний ключ к таблице ELD_GRADEBOOK_DAYS GS_SST_ID внешний ключ к таблице ELD_SCHOOL_STUDENTS_

Продолжение таблицы 3.2.1.1 – Сущности и связи

ELD_GROUP_MEMBERS	GM_G_ID(INTEGER), GM_SST_ID(INTEGER)	ID первичный ключ GM_G_ID внешний ключ к таблице ELD_GROUPS GM_SST_ID внешний ключ к таблице ELD_SCHOOL_STU- DENTS
ELD_IMAGES	ID(INTEGER), PATH_IM- AGE(String), I_EI_ID(INTEGER)	ID первичный ключ I_EI_ID внешний ключ к таблице ELD_EDU- CATIONAL_INSTITU- TIONS
ELD_MESSAGES	ID(INTEGER), MESSAGE(String), DATE(DATETIME) M_U_GETTER_ID(IN- TEGER), M_U_SENDER_ID(IN- TEGER),	ID первичный ключ M_U_GETTER_ID внешний ключ к таб- лице ELD_USERS M_U_SENDER_ID внешний ключ к таб- лице ELD_USERS
ELD_NEWS	ID(INTEGER), TITLE(String), CONTENT(String), DATE(DATE) N_U_OWNER_ID(INTE- GER),	ID первичный ключ N_U_OWNER_ID внешний ключ к таб- лице ELD_USERS
ELD_NEW_COMMENTS	ID(INTEGER), TEXT(String), DATE(DATETIME) NC_N_GETTER_ID(IN- TEGER), NC_U_GETTER_ID(IN- TEGER),	ID первичный ключ NC_N_GETTER_ID внешний ключ к таб- лице ELD_NEWS NC_U_SENDER_ID внешний ключ к таб- лице ELD_USERS
ELD_PARENTS_TYPES	ID(INTEGER), NAME(String)	ID первичный ключ

Продолжение таблицы 3.2.1.1 – Сущности и связи

ELD_PARENTS	ID(INTEGER), EMAIL(STRING), FIRST_NAME(STRING), LAST_NAME(STRING), PATH_IMAGE(BYTE[]), PATRONYMIC(STRING), PHONE_NUMBER(STRING), P_EI_ID(INTEGER), P_U_ID(INTEGER)	ID первичный ключ P_EI_ID внешний ключ к таблице ELD_EDUCATIONAL_INSTITUTIONS P_U_ID внешний ключ к таблице ELD_USERS
ELD_QUARTER_SCORES	SCORE(INTEGER) QS_SS_ID(INTEGER), QS_SST_ID(INTEGER),	ID первичный ключ QS_SS_ID внешний ключ к таблице ELD_SCHOOL_SUBJECTS QS_SST_ID внешний ключ к таблице ELD_SCHOOL_STUDENTS
ELD_SCHEDULE	ID(INTEGER),	ID первичный ключ
ELD_SCHOOL_STUDENTS_AND_PARENTS	ID(INTEGER), SSTAP_P_ID(INTEGER), SSTAP_PT_ID(INTEGER), SSTAP_SST_ID(INTEGER)	ID первичный ключ SSTAP_P_ID внешний ключ к таблице ELD_PARENTS SSTAP_PT_ID внешний ключ к таблице ELD_PARENT_TYPES SSTAP_SST_ID внешний ключ к таблице ELD_SCHOOL_STUDENTS
ELD_SCHEDULE_LESSONS	TIME(DATETIME) SL_G_ID(INTEGER), SL_SH_D(INTEGER), SL_TA_ID(INTEGER),	ID первичный ключ SL_SH_ID внешний ключ к таблице ELD_SCHEDULE SL_TA_ID внешний ключ к таблице ELD_TEACHER_ASSIGNMENTS

Продолжение таблицы 3.2.1.1 – Сущности и связи

ELD_USER_COMMENTS	ID(INTEGER), TEXT(STRING), DATE(DATETIME) UC_U_GETTER_ID(INTEGER), UC_U_SENDER_ID(INTEGER),	ID первичный ключ UC_U_GETTER_ID внешний ключ к таблице ELD_USERS UC_U_SENDER_ID внешний ключ к таблице ELD_USERS
ELD_USERS_TYPES	ID(INTEGER), NAME(STRING)	ID первичный ключ

3.2.2 Схематичное представление

Логическая модель базы данных представлена на рисунке 3.2.2.1

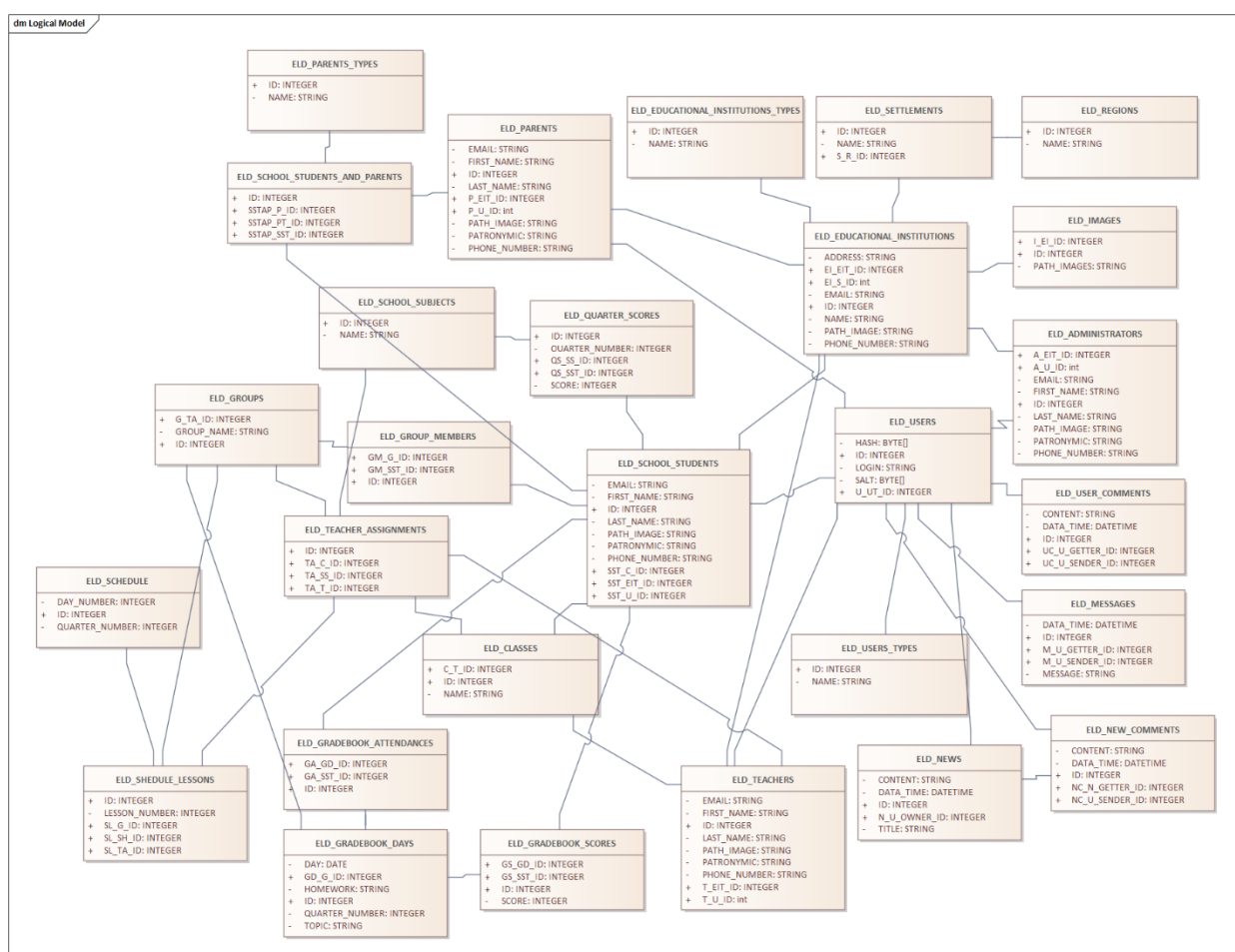


Рисунок 3.2.2.1 — Логическая модель базы данных

3.3 Физическая модель базы данных

3.3.1 Схематичное представление

Физическая модель базы данных представлена на рисунке 3.3.2.1



Рисунок 3.3.2.1 — Физическая модель базы данных

3.4 Разработка алгоритма ПС и отдельных модулей

3.4.1 Алгоритм ПС

Приложение начинает работу с чтения данных из файлов настроек и профиля пользователя, включая параметры интерфейса, учетные данные. После загрузки данных выполняется авторизация: система проверяет логин и пароль, сверяя их с сохраненными хешами. При успешной проверке открывается главное окно, интерфейс которого адаптируется под роль пользователя (администратор, учитель, ученик). Пользователь взаимодействует с программой через элементы управления. Каждое действие обрабатывается последовательно —

валидируются входные параметры, выполняются операции с базой данных, а результаты выводятся на экран. После завершения работы все изменения сохраняются, соединения с базой закрываются, а временные ресурсы очищаются. Основная логика сосредоточена в обработке событий, инициируемых пользователем, что обеспечивает интерактивность и гибкость системы.

Алгоритм ПС представлен на рисунке 3.4.1

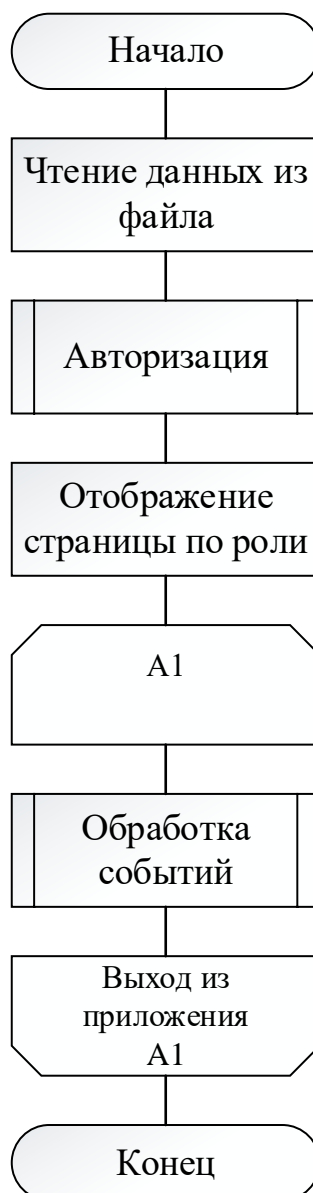


Рисунок 3.4.1 — Алгоритм ПС

3.4.2 Алгоритм общения с сервером

Алгоритм предназначен для стандартизации взаимодействия с сервером, обеспечивая единый подход к обработке HTTP-запросов и ошибок. Процесс начинается с определения типа запроса (GET, POST, PUT, DELETE), после

чего отправляется соответствующий запрос к серверу. При успешном ответе (статус 200-299) данные возвращаются для дальнейшей обработки.

Если возникает ошибка, система анализирует её тип. В случае проблем с авторизацией (например, истекший токен) пользователь перенаправляется на экран ввода учетных данных

Метод завершается либо возвратом данных, либо уведомлением о проблеме

Алгоритм общения с сервером представлен на рисунке 3.4.2



Рисунок 3.4.2 — Алгоритм общения с сервером

3.4.3 Алгоритм адаптивной вёрстки



Рисунок 3.4.3 — Алгоритм адаптивной вёрстки

Алгоритм предназначен для динамической адаптации интерфейса при изменении размеров окна или других условий отображения. Он обеспечивает единый подход к перераспределению и отображению столбцов на экране.

Процесс начинается с регистрации события изменения окна, что может повлиять на количество отображаемых столбцов. Затем система инициирует

перерасчет — определяется максимально возможное количество столбцов и соответствующая им ширина с учетом текущих параметров окна.

Если количество имеющихся столбцов превышает допустимое — лишние элементы временно скрываются. После этого формируется набор столбцов для отображения: выбираются последние доступные, соответствующие рассчитанному количеству.

На финальном этапе производится визуализация — добавленные столбцы отображаются в интерфейсе с заданной шириной. Таким образом, достигается адаптивное отображение данных без перегрузки пользовательского интерфейса.

Алгоритм адаптивной вёрстки представлен на рисунке 3.4.3

3.4.4 Алгоритм расчёта размера столбцов

Алгоритм предназначен для вычисления оптимального количества отображаемых столбцов и их ширины на основе плотности пикселей и текущей ширины окна приложения.

На первом этапе осуществляется получение плотности пикселей экрана устройства. Затем определяется текущая ширина окна приложения.

Если полученное количество столбцов превышает верхний предел (3) — используется максимальное значение. Если результат меньше допустимого минимума (1) — устанавливается минимальное значение.

После определения числа столбцов происходит расчет ширины экрана в пикселях, и на его основе вычисляется ширина одного столбца с применением дополнительного коэффициента.

На выходе алгоритм возвращает рассчитанное количество столбцов и соответствующую им ширину, обеспечивая адаптивное отображение данных в зависимости от устройства и текущих условий.

Алгоритм расчета размера столбцов представлен на рисунке 3.4.4

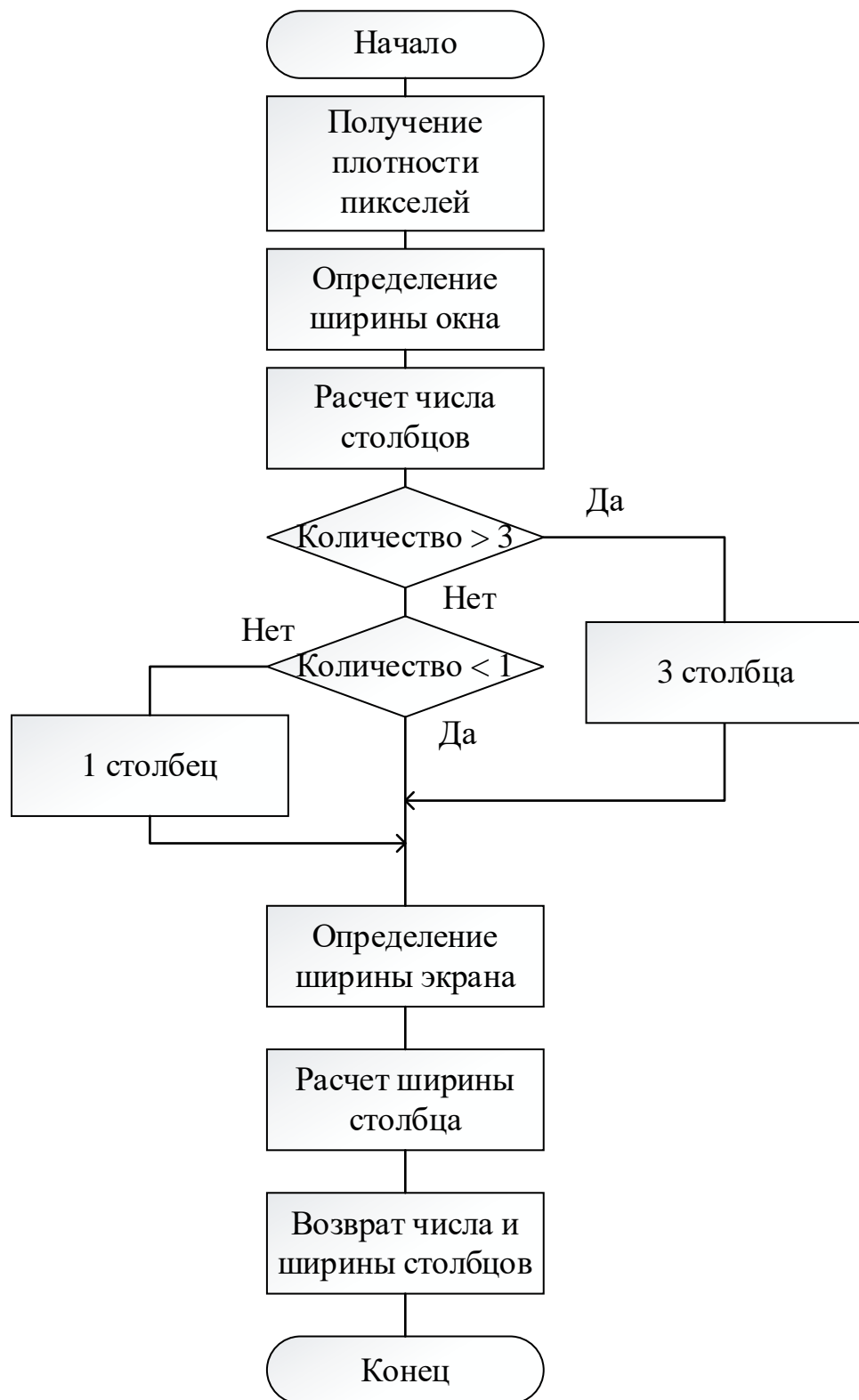


Рисунок 3.4.4 — Алгоритм адаптивной вёрстки

4 СОЗДАНИЕ ПРОГРАММНОГО СРЕДСТВА

4.1 Выбор инструментов разработки

4.1.1 Язык программирования C#

C# — это объектно-ориентированный язык программирования, разработанный корпорацией Microsoft в рамках платформы .NET. Он был создан в начале 2000-х годов и с тех пор постоянно развивается, становясь одним из самых популярных языков для создания программного обеспечения. C# используется для разработки веб-приложений, десктопных приложений, мобильных решений, а также игр.[11]

Особенности C#:

- Объектно-ориентированность — поддержка классов, интерфейсов, наследования и полиморфизма.
- Безопасность типов — предотвращение ошибок, связанных с некорректными операциями с памятью.
- Совместимость с .NET — возможность работы с библиотеками и фреймворками .NET.
- Автоматическое управление памятью — система сборки мусора освобождает память автоматически.
- Асинхронность — поддержка асинхронного программирования с использованием `async` и `await`.
- C# активно используется в корпоративной среде благодаря своей надежности, безопасности и удобству работы.

4.1.2 Фреймворк .NET MAUI

.NET Multi-platform App UI (.NET MAUI) — это кроссплатформенный фреймворк от Microsoft, предназначенный для создания мобильных и десктопных приложений с использованием единой кодовой базы. Он является развитием Xamarin.Forms и позволяет разрабатывать приложения для Windows, macOS, iOS и Android.[12]

Преимущества .NET MAUI:

- Единая кодовая база — один проект для всех платформ.
- Гибкость интерфейса — поддержка MVU (Model-View-Update) и традиционного MVVM (Model-View-ViewModel).
- Поддержка C# и XAML — разработка UI с использованием знакомых инструментов.
- Кроссплатформенные API — возможность работы с файлами, сетью, сенсорами и другими функциями устройств.
- Оптимизированная производительность — улучшенная работа с памятью и графическими интерфейсами.

.NET MAUI является перспективным решением для создания современных бизнес-приложений, обеспечивая простоту разработки и поддержку множества платформ.

4.1.3 Community Toolkit для .NET MAUI

.NET MAUI Community Toolkit — это библиотека, созданная для расширения возможностей .NET MAUI.[13]

Она включает в себя:

- Дополнительные элементы управления (Expander, Popup и другие).
- Конвертеры данных (например, преобразование bool в Visibility).
- Расширенные анимации и стилизация UI.
- Использование Community Toolkit значительно упрощает разработку, помогая создавать удобные и функциональные интерфейсы.

4.1.4 Интегрированная среда разработки Visual Studio

Visual Studio — это мощная интегрированная среда разработки (IDE) от Microsoft. Она поддерживает множество языков программирования, включая C#, и предоставляет удобные инструменты для написания, отладки и тестирования кода.[14]

Ключевые возможности Visual Studio:

- Редактор кода с подсветкой синтаксиса и автодополнением.
- Инструменты отладки — возможность установки точек останова, просмотра переменных в реальном времени.
- Интеграция с Git и GitHub — удобное управление версиями.
- Поддержка контейнеров и облачных сервисов — развертывание приложений в Azure.
- Благодаря Visual Studio разработка становится более удобной и эффективной.

4.1.5 Oracle Database

Oracle Database — это одна из самых мощных и надежных систем управления базами данных (СУБД). Она широко используется в корпоративных решениях благодаря своей высокой производительности, безопасности и поддержке работы с большими объемами данных.[15]

Преимущества Oracle Database:

- Высокая масштабируемость — поддержка кластерных решений.
- Надежность и безопасность — расширенные механизмы резервного копирования и восстановления данных.
- Поддержка SQL и PL/SQL — удобный язык программирования для работы с БД.
- Интеграция с облачными сервисами — возможность работы с Oracle

Cloud.

– Oracle Database является основой для множества корпоративных приложений, требующих высокой надежности.

4.1.6 PL/SQL Developer

PL/SQL Developer — это специализированная среда разработки для работы с языком программирования PL/SQL в Oracle Database. Этот инструмент позволяет писать, отлаживать и тестировать SQL-скрипты и хранимые процедуры.[16]

Ключевые возможности PL/SQL Developer:

- Редактор кода с автодополнением.
- Инструменты отладки и профилирования.
- Поддержка работы с объектами базы данных.
- Гибкая система отчётов и анализа данных.
- Использование PL/SQL Developer значительно облегчает работу разработчиков баз данных, ускоряя процесс написания кода и его тестирования.

5 ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

6 РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ

6.1 Развертывание серверной части

6.1.1 Установка зависимостей в Ubuntu.

Обновление системы.

```
sudo apt update && sudo apt upgrade -y
```

Установка докера.

```
sudo apt install docker.io docker-compose -y  
sudo systemctl enable --now docker
```

Установка Java.

```
sudo apt install openjdk-17-jdk -y
```

Установка Nginx;

```
sudo apt install nginx -y
```

6.1.2 Запуск Oracle Database в Docker.

Проверка наличия образа.

```
docker images | grep gvenzl/oracle-free
```

При отсутствии образа выполнить загрузку.

```
docker pull gvenzl/oracle-free:23-slim
```

Создание контейнера.

```
docker run -d \  
  --name oracle \  
  -p 1521:1521 \  
  -e ORACLE_PASSWORD=<ваш_пароль> \  
  gvenzl/oracle-free:23-slim
```

Замените <ваш_пароль> на надежный пароль для администратора БД.

Подключитесь к контейнеру и выполните SQL-скрипты для создания пользователя и схемы.

```
docker exec -it oracle sqlplus sys/<ваш_пароль>@//localhost:1521/FREEPDB1 as sysdba
```

6.1.3 Запуск и настройка Nginx.

Создание файла конфигурации.

```
sudo nano /etc/nginx/sites-available/meeting-rooms.conf
```

Поместить в файл.

```
server {
    listen 80;
    server_name <ip>;

    # Включение MIME-типов
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    # Одна директория для загрузки и скачивания
    location /files/ {
        alias /var/www/files/;

        # Разрешение PUT-запросов (загрузка)
        dav_methods PUT;
        create_full_put_path on;
        dav_access user:rw group:rw all:r;
        client_max_body_size 100M;

        # Включение листинга (если нужно)
        autoindex on;
        autoindex_exact_size off;
        autoindex_localtime on;

        # Оптимизация кеширования
        expires 30d;
        add_header Cache-Control "public, no-trans-
form";

        # Запрет выполнения PHP
        location ~ /\.php$ {
            deny all;
        }
    }
}
```

Заменить <ip> на ip адресс сервера в глобальной сети.

Активация конфигурации.

```
sudo ln -s /etc/nginx/sites-available/meeting-
rooms.conf /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl reload nginx
```

6.1.4 Запуск Java сервер приложения.

Разместить MeetingRooms-0.0.1-SNAPSHOT.jar в директории на сервере.

Запуск сервера. Сервер автоматически создаст таблицы в базе данных если были выполнены предыдущие шаги.

```
nohup java -jar MeetingRooms-0.0.1-SNAPSHOT.jar \  
    --spring.datasource.url=jdbc:oracle:thin:@lo-  
calhost:1521/FREEPDB1 \  
    --spring.datasource.username=meeting_rooms \  
    --spring.datasource.password=<пароль_пользователя_БД>  
\  
    > app.log 2>&1 &
```

6.2 Установка клиентского части

Скачать установочный файл для выбранной платформы.

Установить.

Принять разрешения.

6.3 Руководство по использованию

ЗАКЛЮЧЕНИЕ

В ходе выполнения преддипломной практики была проведена комплексная работа по проектированию и разработке системы «Электронный дневник», направленной на автоматизацию учебного процесса и повышение эффективности взаимодействия между участниками образовательной среды.

Во введении обоснована актуальность разработки, связанная с необходимостью создания универсальной платформы, объединяющей преимущества существующих решений (School.by, MyClassroom, Edmodo, Google Classroom, ClassDojo) и устраняющей их недостатки. Целью работы стало создание программного средства, обеспечивающего удобный доступ к информации об успеваемости, расписании, домашних заданиях, а также поддержку коммуникации между учениками, родителями, учителями и администрацией.

В разделе аналитического обзора проведён детальный анализ современных образовательных платформ, выявлены их сильные и слабые стороны. Это позволило сформировать требования к проектируемой системе, включая функциональные возможности, входные/выходные данные, а также критерии совместимости и безопасности.

Моделирование предметной области обеспечило чёткое структурирование системы: UML-диаграмма вариантов использования визуализировала взаимодействие пользователей с приложением, а инфологическая модель базы данных определила ключевые сущности и их взаимосвязи. Это стало основой для дальнейшего проектирования архитектуры.

В разделе проектирования обоснован выбор технологий: язык C# и фреймворк .NET MAUI для кроссплатформенной разработки, Oracle Database для надёжного хранения данных, а также инструменты PL/SQL Developer и Visual Studio. Логическая и физическая модели базы данных, алгоритмы работы приложения и взаимодействия с сервером обеспечили высокую производительность, масштабируемость и безопасность системы.

Результатом работы стала архитектура программного средства, сочетающая гибкость интерфейса, ролевой доступ, интеграцию с внешними сервисами и поддержку мобильных устройств. Разработанные модели и алгоритмы позволяют эффективно реализовать такие функции, как ведение электронного журнала, автоматический расчёт среднего балла, управление расписанием и публикация новостей.

Проведённая работа подтвердила возможность создания универсальной платформы, которая не только соответствует современным требованиям к цифровизации образования, но и предоставляет расширенные возможности для анализа успеваемости и повышения вовлечённости всех участников учебного процесса. Дальнейшее развитие проекта может включать внедрение искусственного интеллекта для прогнозирования успеваемости и интеграцию с государственными образовательными порталами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] schools.by [Электронный ресурс]. – Режим доступа: <https://schools.by>.
- [2] Обзор schools.by [Электронный ресурс]. – Режим доступа: <https://picktech.ru/product/schools-by>.
- [3] myClassroom: Class Tools [Электронный ресурс]. – Режим доступа: <https://minga.io/solutions/classroom-management-tools-teachers>.
- [4] Обзор myClassroom [Электронный ресурс]. – Режим доступа: <https://webcatalog.io/ru/apps/myclassroom>.
- [5] Edmodo [Электронный ресурс]. – Режим доступа: <https://bea-edmodo-instruction-rus.tilda.ws>.
- [6] Обзор Edmodo [Электронный ресурс]. – Режим доступа: <https://soware.ru/products/edmodo>.
- [7] Google Classroom [Электронный ресурс]. – Режим доступа: <https://edu.google.co.ug/workspace-for-education/products/classroom>.
- [8] Минобрнауки Татарстана. Методические рекомендации по использованию платформы Google classroom в процессе обучения [Электронный ресурс]. – Режим доступа: https://mon.tatarstan.ru/rus/file/pub/pub_2277963.pdf.
- [9] ClassDojo [Электронный ресурс]. – Режим доступа: <https://www.classdojo.com/ru-ru/points>.
- [10] Гродненский областной институт развития образования. Знакомство с ClassDojo [Электронный ресурс]. – Режим доступа: <https://groiro.by/об-институте/сервисы/новости/p-68273.html>.
- [11] Документация C# [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp>.
- [12] Документация MAUI [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/maui/?view=net-maui-9.0>.
- [13] Документация Community Toolkit для .NET MAUI [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/communitytoolkit/maui>.
- [14] Документация Visual Studio 2022 [Электронный ресурс]. – Режим доступа: <https://visualstudio.microsoft.com/ru>.
- [15] Документация Oracle DataBase [Электронный ресурс]. – Режим доступа: <https://www.oracle.com/database>.
- [16] Документация PL/SQL Developer [Электронный ресурс]. – Режим доступа: <https://www.allroundautomations.com/products/pl-sql-developer>.