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 15100051 |
|  | Преддипломной практика зачтена с оценкой  \_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  (цифрой) (прописью) |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя практики от БГУИР)  \_\_\_\_.\_\_\_\_\_.2025 |

**ОТЧЕТ**

**по преддипломной практике**

Место прохождения практики: РУП «Белтаможсервис»

Сроки прохождения практики: с 10.02.2025 по 23.03.2025

|  |  |  |
| --- | --- | --- |
| Руководитель практики от предприятия:  \_\_\_\_\_\_\_\_\_\_\_ Дерюшев В.В.  (подпись руководителя)  М.П. |  | Студент группы 151004  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Король  (подпись студента)  Руководитель практики от БГУИР  Мелких Е.Г. - старший преподаватель, доцент. |

Минск 2025

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc193721953)

[1 Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству 4](#_Toc193721954)

[1.1 Анализ прототипов 4](#_Toc193721955)

[1.2 Сравнение прототипов 10](#_Toc193721956)

[1.3 Формирование требований к проектируемому программному средству 13](#_Toc193721957)

[2 Моделирование предметной области 14](#_Toc193721958)

[2.1 Моделирование программного обеспечения 14](#_Toc193721959)

[2.2 Инфологическая модель базы данных 15](#_Toc193721960)

[2.3 Спецификация функциональных требований 17](#_Toc193721961)

[3 Проектирования программного средства 20](#_Toc193721962)

[3.1 Выбор архитектуры для разработки 20](#_Toc193721963)

[3.2 Логическая модель баз данных 23](#_Toc193721964)

[3.3 Разработка алгоритмов 32](#_Toc193721965)

[Заключение 36](#_Toc193721966)

[Список использованных источников 37](#_Toc193721967)

# Введение

Электронные дневники становятся важным инструментом в современном образовании, обеспечивая эффективное взаимодействие между учениками, учителями и родителями. В условиях стремительного развития информационных технологий необходимость в автоматизации процессов учета успеваемости и организации учебного процесса становится всё более актуальной. Электронные дневники представляют собой цифровые платформы, которые заменяют традиционные бумажные носители, упрощая доступ к информации и повышая прозрачность взаимодействия всех участников образовательного процесса.

Актуальность данного программного обеспечения заключается в том, что сегмент программных решений для ведущих электронных научных журналов в образовательных учреждениях все еще слабо развит. Это создает возможности для разработки специализированных инструментов, которые соответствуют современным требованиям образовательного процесса. Разрабатываемое программное обеспечение направлено на автоматизацию управления образовательным процессом и взаимодействия между участниками образовательного процесса. Внедрение этой системы повысит уровень цифровизации в образовательных учреждениях, сэкономит время для документации учителей и студентов и обеспечит удобный доступ к актуальной информации для учителей, студентов и их родителей.

С помощью электронных дневников учителя могут оперативно выставлять оценки, фиксировать пропуски и назначать домашние задания. Родители, в свою очередь, получают возможность в любое время отслеживать успеваемость своих детей, а ученики — взаимодействовать с учебным материалом и получать обратную связь от преподавателей.

Кроме того, электронные дневники способствуют систематизации информации об успеваемости, что позволяет проводить анализ и выявлять тенденции в обучении.

Таким образом, электронные дневники не только облегчают работу педагогов, но и создают новые возможности для родителей и учеников, способствуя более глубокому вовлечению всех сторон в образовательный процесс и повышая его качество.

# Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству

## Анализ прототипов

Анализ существующих прототипов позволит лучше понять область аналогичных программных средств, тем самым позволяя выявить их слабые и сильные аспекты. На основе данного анализа можно будет выявить требуемый функционал, а также определить возможности для усовершенствования и развития будущего программного средства.

1.1.1 School.by

**Schools.by** — это образовательная платформа, разработанная для учебных заведений Республики Беларусь. [1] Она предоставляет инструменты для автоматизации учебного процесса, управления данными учащихся и учителей, а также для взаимодействия между участниками образовательного процесса.

Электронный журнал на платформе Schools.by — это цифровой инструмент для учителей и администрации школы, который заменяет традиционный бумажный журнал. Он позволяет учителям выставлять оценки, записывать домашние задания, отмечать посещаемость и вести учет успеваемости учащихся. В журнале отображается расписание уроков, а также есть возможность добавлять комментарии к оценкам или пропускам. Учителя могут быстро формировать отчеты по успеваемости и посещаемости, что упрощает анализ учебного процесса. Администрация школы получает доступ к обобщенной статистике по классам и предметам.

Для учеников предоставлен электронный дневник. Электронный дневник на платформе Schools.by — это персональный раздел для учеников и их родителей, где отображается вся информация об учебном процессе. В дневнике можно увидеть текущие оценки, домашние задания, расписание уроков и отметки о посещаемости. Родители получают уведомления о новых оценках или изменениях в расписании, что позволяет им всегда быть в курсе успехов ребенка. Ученики могут самостоятельно отслеживать свои результаты и планировать выполнение заданий. Дневник доступен через сайт или мобильное приложение, что делает его удобным для использования в любое время. Этот инструмент помогает родителям и детям оставаться вовлеченными в учебный процесс и своевременно реагировать на возникающие вопросы.

Учителя могут загружать учебные материалы, такие как презентации, текстовые документы, видеоуроки, ссылки на внешние ресурсы и другие файлы, которые помогают в изучении предмета. Эти материалы прикрепляются к конкретным урокам или домашним заданиям, что делает их легко доступными для учеников. Управление учебными материалами также включает возможность структурировать информацию по темам, разделам или датам, что упрощает навигацию и поиск нужных данных.

Для учеников и родителей раздел с учебными материалами становится удобной цифровой библиотекой, где можно найти всё необходимое для выполнения заданий или самостоятельного изучения темы. Материалы доступны в любое время через сайт или мобильное приложение, что особенно полезно в случае пропуска уроков или необходимости повторить пройденное. Учителя могут обновлять и дополнять материалы, обеспечивая актуальность информации. Эта функция не только упрощает процесс обучения, но и способствует более глубокому пониманию предмета, так как ученики получают доступ к разнообразным ресурсам, которые могут быть адаптированы под их индивидуальные потребности.

Учителя и администрация школы могут создавать, редактировать и управлять расписанием для каждого класса, указывая предметы, время проведения уроков, кабинеты и преподавателей. Расписание автоматически синхронизируется с электронным журналом и дневником, что позволяет избежать путаницы и ошибок. В случае изменений (например, замены учителя или переноса урока) обновления сразу отображаются в системе, и все участники учебного процесса получают уведомления. Это делает процесс планирования учебного дня более гибким и организованным.

Для учеников и родителей расписание уроков доступно в электронном дневнике, где его можно просматривать на день, неделю или месяц. Это позволяет заранее планировать выполнение домашних заданий, подготовку к урокам или внеучебные мероприятия. Расписание также отображается в мобильном приложении, что делает его доступным в любое время и в любом месте. Ученики могут быстро узнать, какие уроки их ждут, а родители — следить за учебной нагрузкой своего ребенка. Благодаря этой функции все участники образовательного процесса остаются в курсе актуального графика, что способствует более эффективной организации учебного времени.

School.by также поддерживает мобильное приложения своего электронного дневника. Приложение доступно для устройств на базе iOS и Android и предоставляет доступ к ключевым функциям платформы, таким как электронный дневник, расписание уроков, оценки, домашние задания и учебные материалы. Интерфейс приложения интуитивно понятен и адаптирован для мобильных устройств, что делает его использование простым и комфортным. Уведомления о новых оценках, изменениях в расписании или домашних заданиях приходят прямо на телефон, что помогает всегда быть в курсе важных событий.

Для учеников мобильное приложение становится личным помощником в учебе: они могут быстро проверить расписание, посмотреть домашние задания или узнать свои оценки. Родители получают возможность следить за успеваемостью ребенка и оперативно реагировать на изменения. Учителя также могут использовать приложение для быстрого доступа к журналу, расписанию или учебным материалам. Мобильное приложение Schools.by значительно упрощает взаимодействие между всеми участниками образовательного процесса, делая его более мобильным, гибким и современным. Это особенно полезно в условиях активного ритма жизни, когда важно иметь доступ к информации в любой момент.

В качестве преимущества сайта Schools.by можно выделить его удобство, прозрачность и функциональность. Платформа объединяет все ключевые аспекты учебного процесса — электронный журнал, дневник, расписание, учебные материалы и коммуникацию между участниками. Это позволяет учителям, ученикам и родителям оперативно получать доступ к необходимой информации, будь то оценки, домашние задания или изменения в расписании. Сайт экономит время, автоматизируя рутинные процессы, такие как заполнение журналов и отчетов, а также обеспечивает прозрачность учебного процесса для родителей. Доступность через мобильное приложение делает платформу еще более удобной для использования в повседневной жизни.

Однако School.by имеет и ряд недостатков они связаны с техническими ограничениями и сложностями в использовании для некоторых пользователей. Например, при высокой нагрузке на серверы возможны замедления в работе платформы. Некоторые пользователи, особенно старшего поколения, могут испытывать трудности с освоением интерфейса или функционала сайта. Несмотря на эти минусы, Schools.by остается одним из самых популярных и полезных инструментов для организации учебного процесса в Беларуси.

1.1.2 Eljur.by

**Eljur.by** — это еще одна популярная образовательная платформа, используемая в школах Беларуси для автоматизации учебного процесса [2]. Она предоставляет широкий набор функций, которые помогают учителям, ученикам и родителям эффективно взаимодействовать и управлять учебной деятельностью.

Как и в предыдущем прототипе Eljur поддерживает журнал для администрации и учителей, а также электронный дневник для учеников школ. В журнале учителя могут выставлять оценки за уроки, контрольные работы и домашние задания, а также отмечать посещаемость учеников. Все данные автоматически синхронизируются с электронным дневником, что позволяет родителям и ученикам видеть актуальную информацию. Журнал также поддерживает возможность добавления комментариев к оценкам или пропускам, что делает процесс более прозрачным и информативным. Это значительно упрощает работу учителей и снижает вероятность ошибок при заполнении документации.

Электронный дневник является персональным разделом для учеников и их родителей, где отображаются оценки, домашние задания и расписание уроков. Родители могут отслеживать успеваемость ребенка в реальном времени, а ученики — планировать выполнение заданий и готовиться к урокам. Дневник также показывает средний балл и итоговые оценки, что помогает оценить прогресс. Уведомления о новых оценках или изменениях в расписании приходят автоматически, что делает процесс обучения более организованным.

Учителя могут добавлять домашние задания в Eljur.by, указывая сроки выполнения и прикрепляя дополнительные материалы, такие как файлы или ссылки. Ученики видят задания в своем дневнике и могут отслеживать их выполнение. Родители также получают доступ к заданиям, что позволяет им контролировать процесс обучения ребенка. Эта функция помогает ученикам лучше организовывать свое время и готовиться к урокам.

Eljur.by предоставляет более развитые возможности для общения между участниками учебного процесса. Eljur.by включает чаты между учителями, учениками и родителями. Это позволяет оперативно решать вопросы, обсуждать успеваемость и обмениваться важной информацией. Такая функция особенно полезна для более тесного взаимодействия между всеми участниками образовательного процесса.

Учителя и администрация могут формировать подробные отчеты по успеваемости, посещаемости и другим параметрам, включая статистику по классам, отдельным ученикам или предметам. Это позволяет выявлять слабые места в учебном процессе и принимать informed решения для улучшения результатов. Такая аналитика особенно полезна для крупных школ с большим количеством данных.

Eljur.by часто предоставляет больше возможностей для интеграции с другими образовательными сервисами и системами. Это может включать синхронизацию с библиотеками, сторонними платформами для тестирования или системами управления школой. Эти аспекты делает Eljur.by более гибким решением для школ, которые используют несколько образовательных инструментов.

В Eljur.by пользователи могут более гибко настраивать уведомления, выбирая, какие события (новые оценки, изменения в расписании, сообщения) и каким способом (email, push-уведомления) они хотят получать. Это делает Eljur.by более удобным для тех, кто хочет персонализировать свой опыт использования платформы.

Eljur.by позволяет не только загружать учебные материалы, но и организовывать их в структурированные каталоги, добавлять теги и категории. Это упрощает поиск и доступ к материалам для учеников и учителей. Eljur.by также предлагает инструменты для создания индивидуальных образовательных траекторий, что позволяет учитывать особенности каждого ученика. Учителя могут назначать персональные задания, отслеживать прогресс и адаптировать учебный процесс под потребности конкретного ученика. В Schools.by такие функции либо отсутствуют, либо реализованы в более базовом формате.

Также нужно упомянуть, что данный сервис включает дополнительные инструменты для дистанционного обучения, такие как интеграция с видеоконференциями, онлайн-тестирование и возможность проведения виртуальных уроков. Это делает платформу более адаптированной к современным условиям, где дистанционное обучение становится все более востребованным.

Eljur.by выделяется среди аналогов своим расширенным функционалом и гибкостью. Платформа предлагает глубокую аналитику, детализированные отчеты, развитые инструменты коммуникации (чаты, уведомления) и поддержку индивидуальных образовательных траекторий. Это делает Eljur.by идеальным выбором для крупных школ или учебных заведений, которым требуется больше возможностей для управления учебным процессом. Кроме того, интеграция с внешними системами и поддержка дистанционного обучения делают платформу более адаптированной к современным образовательным требованиям.

Главный недостаток Eljur.by — это его сложность для новичков. Из-за насыщенного интерфейса и большого количества функций пользователям, особенно тем, кто не знаком с подобными платформами, может потребоваться время для освоения. Кроме того, расширенный функционал иногда может быть избыточным для небольших школ или пользователей, которым нужны только базовые возможности. Также, как и у других платформ, возможны технические сбои при высокой нагрузке на серверы.

В целом Eljur.by является мощная и многофункциональная образовательная платформа, которая подходит для школ, стремящихся к автоматизации и глубокому анализу учебного процесса. Ее расширенные возможности, такие как аналитика, интеграция с внешними системами и поддержка дистанционного обучения, делают ее одним из лидеров среди аналогов. Однако сложность интерфейса и избыточность функций для некоторых пользователей могут быть препятствием для быстрого внедрения. В целом, Eljur.by заслуживает высокой оценки за свою универсальность и адаптивность к современным образовательным потребностям, но требует внимательного подхода к обучению пользователей.

1.1.3 Znaij.by

Znaij.by— это одна из популярных образовательных платформ в Беларуси, которая предоставляет доступ к электронному дневнику, расписанию уроков и другим инструментам для управления учебным процессом [3]. Платформа используется многими школами страны и ориентирована на удобство взаимодействия между учителями, учениками и родителями.

Электронный дневник на платформе Znaij.by — это ключевой инструмент для учеников и их родителей. Здесь отображаются текущие оценки, комментарии учителей и информация о домашних заданиях. Родители могут в реальном времени отслеживать успеваемость ребёнка, а ученики — своевременно выполнять задания и готовиться к урокам. Дневник также показывает средний балл и итоговые оценки, что помогает оценить прогресс в учёбе. Уведомления о новых оценках или изменениях в расписании приходят автоматически, что делает процесс обучения более организованным. Это особенно полезно для родителей, которые хотят быть в курсе успехов своего ребёнка.

Расписание уроков на Znaij.by позволяет ученикам и родителям всегда быть в курсе учебного графика. В расписании указываются предметы, время проведения уроков, кабинеты и преподаватели. В случае изменений (например, замены учителя или переноса урока) обновления сразу отображаются в системе. Ученики могут просматривать расписание на день, неделю или месяц, что помогает заранее планировать выполнение домашних заданий и подготовку к урокам. Родители также могут следить за учебной нагрузкой своего ребёнка и быть в курсе любых изменений в расписании.

Учителя могут добавлять домашние задания в **Знай.by**, указывая сроки выполнения и прикрепляя дополнительные материалы, такие как файлы или ссылки. Ученики видят задания в своём дневнике и могут отслеживать их выполнение. Родители также получают доступ к заданиям, что позволяет им контролировать процесс обучения ребёнка. Эта функция помогает ученикам лучше организовывать своё время и готовиться к урокам, а родителям — быть уверенными в том, что их ребёнок успевает выполнять все задания.

Уведомления на Znaij.by — это важный инструмент для оперативного информирования. Родители и ученики получают сообщения о новых оценках, изменениях в расписании или домашних заданиях. Это помогает всегда быть в курсе важных событий. Уведомления можно настроить через сайт или мобильное приложение, что делает их удобными для использования в повседневной жизни. Это особенно полезно для родителей, которые хотят оперативно реагировать на изменения в учебном процессе.

Мобильное приложение Znaij.by доступно для устройств на базе iOS и Android. Оно предоставляет доступ ко всем основным функциям платформы, включая электронный дневник, расписание уроков и домашние задания. Приложение также поддерживает уведомления, что делает его удобным для использования на смартфонах и планшетах. Ученики и родители могут просматривать оценки, задания и расписание в любое время и в любом месте, что делает процесс обучения более мобильным и гибким.

## Сравнение прототипов

В процессе поиска прототипов были выделены следующие критерии для сравнения: функционал, интерфейс, кроссплатформенность, экспорт данных, интеграция, цена.

* + 1. Функционал

Параметр отвечает за основные возможности платформы, которые помогают в организации учебного процесса.

School.by предоставляет базовые функции, такие как ведение расписания, оценок, домашних заданий и уведомления для родителей.

Eljur.by предлагает более расширенный функционал, включая расписание, оценки, домашние задания, SMS-уведомления и чат с учителями.

Znaij.by также включает расписание, оценки и домашние задания, но добавляет аналитику успеваемости, что помогает отслеживать прогресс ученика.

* + 1. Интерфейс

Параметр описывает удобство и простоту использования платформы.

School.by отличается простым и минималистичным интерфейсом, который подходит для быстрого доступа к данным.

Eljur.by предлагает современный и интуитивно понятный интерфейс, удобный для родителей и учителей.

Znaij.by выделяется ярким и красочным интерфейсом с элементами геймификации, что мотивирует учеников.

* + 1. Кроссплатформенность

Параметр показывает, на каких устройствах и операционных системах доступна платформа.

School.by доступен через веб-версию и мобильное приложение для iOS и Android.

Eljur.by также поддерживает веб-версию и мобильные приложения для iOS и Android.

Znaij.by предоставляет аналогичные возможности, включая веб и мобильные приложения для iOS и Android.

* + 1. Экспорт данных

Параметр описывает возможность экспорта данных в различные форматы.

School.by предлагает ограниченные возможности экспорта, такие как расписание в PDF.

Eljur.by поддерживает экспорт оценок и расписания в PDF и Excel, что удобно для анализа и печати.

Znaij.by позволяет экспортировать данные в PDF и другие форматы для удобства пользователей.

* + 1. Интеграции

Параметр показывает, насколько платформа интегрирована с другими сервисами (календари, задачи, облачные хранилища).

School.by интегрируется с образовательными платформами и школьными системами, что упрощает работу учителей.

Eljur.by поддерживает интеграцию с календарями и другими школьными сервисами, делая платформу более универсальной.

Znaij.by также интегрируется с внешними сервисами и образовательными платформами, расширяя возможности использования.

* + 1. Цена

Параметр описывает стоимость использования платформы.

School.by бесплатен для школ, использующих платформу, что делает её доступной для всех.

Eljur.by требует платной подписки для родителей и школ, но предлагает расширенный функционал.

Znaij.by также бесплатен для школ, использующих платформу, что делает её привлекательной для образовательных учреждений.

Таблица 1.2.1 – Сравнительная характеристика аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | School.by | Eljur.by | Znaij.by |
| Функционал | Ведение расписания, оценок, домашних заданий, уведомления для родителей. | Расписание, оценки, домашние задания, SMS-уведомления, чат с учителями. | Расписание, оценки, домашние задания, уведомления, аналитика успеваемости. |
| Интерфейс | Простой и минималистичный интерфейс. | Современный и интуитивно понятный интерфейс. | Удобный и красочный интерфейс с элементами геймификации. |
| Кроссплатформенность | Веб-версия и мобильное приложение. | Веб-версия и мобильное приложение. | Веб-версия и мобильное приложение |
| Экспорт данных | Ограниченные возможности экспорта в PDF. | Экспорт оценок и расписания в PDF и Excel. | Экспорт данных в PDF и другие форматы. |
| Интеграции | Интеграция с образовательными платформами и школьными системами. | Интеграция с календарями и другими школьными сервисами. | Интеграция с внешними сервисами и образовательными платформами. |
| Цена | Бесплатно для школ, использующих платформу. | Платная подписка для родителей и школ. | Бесплатно для школ, использующих платформу. |

Подраздел «Анализ прототипов» предоставляет детальный обзор существующих прототипов. Основная цель этого анализа — исследовать текущую рыночную ситуацию в области электронных дневников и определить их ключевые характеристики, сильные и слабые стороны.

Таким образом, этот раздел содержит важные сведения, которые помогут в создании нового сервиса, ориентированного на запросы пользователей и способного эффективно конкурировать с уже имеющимися на рынке решениями.

## Формирование требований к проектируемому программному средству

Исходя из анализа прототипов был сформирован список требований к проектируемому программному средству:

- Журнал для каждого учителя и его предмета, что позволит учителю вносить оценки и пропуски для учеников.

- Электронный дневник для учеников и родитель, чтобы получать информацию о расписании и предметах.

- Возможность добавления школ с их уникальным содержимым.

- Для каждой школы иметь возможность добавления администратора, имеющего доступ к редактированию собственной школы.

- Реализовать подсчёт среднего бала для каждого ученика в четверти.

- Отображения профиля для каждого пользователя в зависимости от его роли в веб-приложении.

- Поддержка добавления родителей для учеников, чтобы они имели доступ к электронному дневнику соответсвенного ученика.

- Возможность создавать группы и подгруппы в случаях раздельных уроков.

- Отображение общей информации о школах.

Подраздел «Формирование требований к проектируемому программному средству» представляет собой перечень ключевых требований, которые должны быть реализованы в новом программном средстве “Электронный дневник”. Основная цель этого раздела — определить функциональные и технические характеристики будущего программного средства, которые обеспечат его конкурентоспособность, удобство использования и соответствие потребностям целевой аудитории.

Что сделано:

1. На основе анализа прототипов программных средств был сформирован список требований к проектируемому продукту.
2. Требования охватывают широкий спектр функциональных возможностей.
3. Учтены необходимости по созданию различных ролей с различным доступом в программном средстве для обеспечения безопасности данных.

Таким образом, данный раздел служит основой для разработки программного средства, которое будет не только функциональным и удобным, но и конкурентоспособным на рынке, удовлетворяя потребности пользователей.

# Моделирование предметной области

## Моделирование программного обеспечения

Для описания функциональной модели разрабатываемого программного продукта была использована диаграмма вариантов использования на языке UML [4]. Данная диаграмма визуализирует функциональные требования системы, демонстрируя способы взаимодействия пользователя с программным средством и получение значимых для него результатов.

UML — это универсальный язык, предназначенный для визуального моделирования процессов разработки программного обеспечения. С его помощью разработчики создают модели, которые упрощают понимание, проектирование, документирование и визуализацию системы. Этот язык включает набор графических элементов, позволяющих строить различные типы диаграмм для описания структуры, поведения и взаимодействия компонентов системы.

На основе проведённого анализа и исходя из требований к программному средству была разработана и создана UML диаграмма, описывающая аспекты разрабатываемого программного средства. Диаграмма разрабатываемого программного средства представлена на рисунке 2.1.1.

Изображение выглядит как зарисовка, рисунок, диаграмма, Штриховая графика

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2.1.1 — UML диаграмма

Раздел «Моделирование программного обеспечения» посвящён описанию UML (Unified Modeling Language) диаграммы, предназначенной для визуализации и моделирования функциональных возможностей разрабатываемого программного обеспечения. Основная цель этого раздела — предоставить наглядное представление о взаимодействии пользователей с системой, доступных функциях и их взаимосвязях.

## Инфологическая модель базы данных

Выбор инфологической модели в качестве основного инструмента для описания структуры базы данных обусловлен тем, что она позволяет сосредоточиться на концептуальных аспектах данных и их взаимосвязях, что помогает лучше понять бизнес-логику системы. Инфологическая модель предоставляет возможность визуализировать основные сущности, их атрибуты и связи, что облегчает выявление требований и способствует эффективному обсуждению архитектуры базы данных с заинтересованными сторонами. Кроме того, использование инфологической модели способствует более структурированному подходу к проектированию, что в дальнейшем упрощает переход к логическим и физическим моделям базы данных.

Таблица 2.2.1 – Инфологическая модель базы данных

|  |  |
| --- | --- |
| Сущность | Атрибуты |
| Главный администратор | - Логин  - Хеш пароля  - ФИО  - Почта  - Номер телефона |
| Администратор школы | - Логин  - Хеш пароля  - ФИО  - Почта  - Номер телефона |
| Учитель | - Логин  - Хеш пароля  - ФИО  - Почта  - Номер телефона  - Руководимый класс |

Продолжение таблицы 2.2.1 – Инфологическая модель базы данных

|  |  |
| --- | --- |
| Ученик | - Логин  - Хеш пароля  - ФИО  - Почта  - Номер телефона  - Класс |
| Родитель | - Логин  - Хеш пароля  - ФИО  - Почта  - Номер телефона  - Тип родителя |
| Класс | - Название  - Классный руководитель |

Подраздел «Инфологическая модель базы данных» предоставляет текстовое описание структуры базы данных проектируемого программного средства. Основная цель этого подраздела — представить логическую модель базы данных, которая описывает основные сущности и их атрибуты, необходимые для функционирования системы. Это позволяет чётко определить, какие данные будут храниться в системе, как они будут организованы и связаны между собой.

Что сделано:

1. Разработана инфологическая модель базы данных, представленная в виде многоуровневого списка.
2. Определены основные сущности базы данных.
3. Для каждой сущности перечислены ключевые атрибуты, которые описывают её свойства и характеристики.

Преимущества данного решения:

1. Чёткая структура данных. Инфологическая модель позволяет чётко определить, какие данные будут храниться в системе и как они будут организованы, что упрощает процесс проектирования и реализации базы данных.
2. Упрощение разработки. Наличие чёткой модели базы данных помогает разработчикам лучше понять структуру системы и ускоряет процесс создания базы данных, так как все сущности и их атрибуты уже определены.
3. Масштабируемость. Модель позволяет легко добавлять новые сущности или изменять существующие, что делает систему гибкой и адаптируемой к будущим изменениям.
4. Документирование. Инфологическая модель служит важным элементом документации, который помогает зафиксировать структуру данных на ранних этапах разработки, что снижает риск ошибок и недопонимания в дальнейшем.
5. Улучшение взаимодействия между командами. Чёткая модель базы данных облегчает взаимодействие между разработчиками, тестировщиками и другими участниками проекта, так как все имеют единое представление о структуре данных.

Таким образом, данный подраздел предоставляет важную основу для проектирования базы данных, которая обеспечивает чёткую структуру данных, упрощает разработку и способствует созданию масштабируемой и гибкой системы.

## Спецификация функциональных требований

Описание спецификации функциональных требований поможет четко определить, как должна функционировать система, какие элементы интерфейса необходимо реализовать и как они будут взаимодействовать с пользователем. Это позволит структурировать процесс разработки, минимизировать риски недопонимания между заказчиком и разработчиком, а также обеспечит прозрачность и контроль на всех этапах создания продукта. Кроме того, такая спецификация станет основой для тестирования, позволяя убедиться, что реализованная функциональность соответствует ожиданиям и требованиям.

* + 1. Вход в систему

Гость при попытке захода в систему сразу будет отправлен в окно авторизации, где ему будет необходимо ввести свой логин и пароль. После успешной авторизации пользователь будет переведён в соответствующий раздел.

* + 1. Профиль пользователя

Для каждого пользователя создан свой уникальный профиль в соответствии с его ролью.

Администрация школы:

- ФИО пользователя.

- Базовая информация о пользователе.

Учитель:

- ФИО пользователя.

- Базовая информация о пользователе.

- Список предметов.

- Руководимый класс.

Ученик:

- ФИО пользователя.

- Базовая информация о пользователе.

- Класс.

- Список родителей.

- Дневник

Родитель:

- ФИО пользователя.

- Базовая информация о пользователе.

- Список детей.

* + 1. Панель админа

Доступ к панели админа имеют 2 роли, а именно главный админ и админ школы. На данной панели главный админ имеет доступ к списку школы, списку админов школы, списку учителей и списку классов. Так же админ имеет доступ к добавлению и удалению этих сущностей.

Админ школы имеет тот же доступ, что и главный админ, однако он не может получить список школ, а лишь собственную школу, а также не может её удалить.

* + 1. Страница школы

На странице школы выводится общая информация о школе, выводятся панели для быстрого перехода к другим разделам, а также фотографии этой школы.

* + 1. Страница класса

На странице класса должна быть выведена информация о классном руководителе, а также список учеников, относящихся к этому классу. Админ имеет доступ к добавлению и удалению учеников.

* + 1. Список родителей

На данной странице пользователь получает информацию о родителях определённого ученика, а также о их типе. Админ имеет доступ к добавлению и удалению родителей ученика.

* + 1. Список детей

На этой странице выводится список детей определённого родителя.

* + 1. Журнал

Журнал предоставляет доступ учителю к списку класса, где учитель может выставить им оценки или пропуски.

* + 1. Дневник

К дневнику ученика имеют доступ ученик и его родители. В дневнике можно просмотреть расписания класса, его оценки и домашние задачи.

Что сделано:

1. Описаны ключевые функциональные требования к системе.
2. Для каждой функции указаны конкретные требования.
3. Определены сценарии взаимодействия пользователей с системой.

Преимущества спецификации функциональных требований:

1. Спецификация функциональных требований обеспечивает ясное понимание работы системы, что упрощает процессы разработки и тестирования.
2. Все функциональные требования описаны в едином формате, что способствует согласованности представлений о системе у разработчиков, тестировщиков и других участников проекта.
3. Спецификация функциональных требований делает все этапы разработки более прозрачными, что улучшает коммуникацию между командами и заинтересованными сторонами.
4. Детализированные требования помогают выявить потенциальные проблемы на ранних этапах, что способствует созданию более качественного и надёжного продукта.
5. Чёткое понимание требований позволяет оптимально распределять ресурсы, избегая перерасхода времени и бюджета на ненужные задачи.

# Проектирования программного средства

## Выбор архитектуры для разработки

Выбор архитектуры для разработки веб-приложения является критически важным этапом, так как он определяет структуру, масштабируемость, производительность и удобство поддержки приложения в долгосрочной перспективе. Правильно выбранная архитектура позволяет эффективно распределять задачи между компонентами системы, обеспечивает гибкость при внесении изменений и упрощает интеграцию новых функций.

* + 1. Язык программирования Java

*Java* – это многоплатформенный, объектно-ориентированный и сетецентрический язык, который сам по себе может использоваться как платформа. Это быстрый, безопасный и надежный язык программирования для всего: от мобильных приложений и корпоративного ПО до приложений для работы с большими данными и серверных технологий [5].

*Java* используется для разработки обычных прикладных программ, которые запускаются пользователем и имеют доступ ко всем ресурсам компьютера наравне с любыми другими программами. Он также используется для операций с *Big Data* и создания разнообразных научно-технических программ.

Программы на *Java* транслируются в байт-код *Java*, выполняемый виртуальной машиной *Java* (*JVM*). Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять *Java*-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии *Java* является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной.

* + 1. Среда разработки *IntelliJ IDEA*

*IntelliJ IDEA* – это среда разработки на языке *Java*. *IntelliJ IDEA* предоставляет интегрированный инструментарий для разработки графического пользовательского интерфейса. Среди прочих возможностей среда хорошо совместима со многими популярными свободными инструментами разработчиков, такими как *CVS*, *Subversion*, *Apache Ant*, *Maven* и *JUnit*. В *IntelliJ* *IDEA* те же горячие клавиши и интерфейс, что и в других продуктах *JetBrains*.

Визуальные подсказки помогают немедленно устранить проблемы, не дожидаясь появления ошибок во время сборки или выполнения. Если навести указатель мыши на волнистую линию, на экран будут выведены дополнительные сведения об ошибке. Имеется возможность одним нажатием кнопки отформатировать код и применить к нему исправления. Рефакторинг включает в себя такие операции, как интеллектуальное переименование переменных, извлечение одной или нескольких строк кода в новый метод и изменение порядка параметров методов. Предоставляет возможности совместного редактирования и отладки в реальном времени независимо от типа приложения или языка.

Кроме того, *IntelliJ IDEA* поможет в разработке веб приложений: она предлагает эффективные встроенные инструменты, поддержку *Java* и связанных с ней технологий. А бесплатные плагины, позволяют дополнительно расширить возможности *IntelliJ IDEA* и использовать ее для работы с другими языками программирования.

* + 1. Фреймворк *Java Spring*

*Spring* — это фреймворк для разработки современных *Java*-приложений на любой платформе. Он предоставляет комплексную модель программирования и конфигурации, а также поддержку различных технологий и стандартов *JavaEE.*

Фреймворк *Java* *Spring* предоставляет разработчикам инструменты для создания сложных систем, например многопользовательских корпоративных веб-приложений со множеством функций для бизнеса. Он позволяет быстро создавать приложения, которые умеют работать с базами данных и облаками, состоят из разных модулей, обмениваются данными с пользователями через интернет по защищённым каналам. Теоретически всё это можно реализовать в *Java* и вручную, но *Spring* даёт разработчикам уже готовые инструменты, которые позволяют писать код гораздо быстрее и концентрироваться не на формальностях, а на уникальных функциях программы [6].

У фреймворка достаточно много модулей, например, такие как: *Spring* *Core*, Spring Security, Spring MVC, Spring Data JPA, модуль авторизации, модуль интеграции, Spring Boot.

*Spring Core* – это основной модуль *Spring*-фреймворка, который предоставляет основные функции, такие как управление зависимостями, инверсия управления, внедрение зависимостей и аспектноориентированное программирование. *Spring* *Core* также определяет общие абстракции и интерфейсы для других модулей Spring.

*Spring* *Security* – это модуль *Spring*-фреймворка, который предоставляет мощные и гибкие механизмы для защиты веб-приложений от различных видов атак, таких как аутентификация, авторизация, *CSRF*, *XSS* и т. д. *Spring* *Security* также интегрируется с другими технологиями и стандартами безопасности, такими как *OAuth2*, *JWT*, *LDAP* и т.д.

*Spring* *MVC* – это модуль Spring-фреймворка, который предоставляет модель-вид-контроллер (*MVC*) для разработки веб-приложений на основе *Servlet* *API*. *Spring* *MVC* поддерживает аннотации, *RESTful* веб-сервисы, валидацию данных, преобразование данных, интернационализацию и локализацию, тестирование и т.д.

*Spring* *Data* *JPA* – это модуль Spring-фреймворка, который предоставляет удобный доступ к данным в реляционных базах данных с помощью *Java* *Persistence* *API* (*JPA*). *Spring* *Data* *JPA* упрощает реализацию слоя доступа к данным путем предоставления готовых интерфейсов репозиториев и поддержки запросов на основе методов и аннотаций.

Модуль авторизации – этот модуль позволяет настраивать авторизацию и аутентификацию в системе. Он обеспечивает разделение прав пользователей и защиту паролей.

Модуль интеграции – предназначен для того, чтобы интегрировать корпоративные приложения в единую систему: обмениваться сообщениями, активировать функции и строить единую архитектуру.

*Spring Boot –* это отдельный модуль, который упрощает настройку фреймворка Spring и ускоряет запуск проектов. Он может автоматически сконфигурировать приложение и создать веб-сервер для его запуска. Большинство новых приложений Spring создаётся с помощью Spring Boot [5].

* + 1. Система управления базами данных *Oracle*

Oracle – это мощная объектно-реляционная система управления базами данных (ОРСУБД), которая поддерживает широкий спектр возможностей стандарта SQL, включая современные стандарты, такие как SQL:2011. Она доступна для множества платформ, включая UNIX-подобные системы (например, Linux и macOS), а также для Microsoft Windows. Oracle является одной из самых популярных СУБД в корпоративной среде благодаря своей надежности, производительности и функциональности [7].

Oracle поддерживает транзакции, которые позволяют объединять несколько операций в одну логическую единицу работы. Если какая-либо из операций в транзакции завершается с ошибкой, все изменения откатываются, что обеспечивает целостность данных. Среди ключевых особенностей Oracle можно выделить поддержку JSON-типа данных, расширенные возможности для работы с геоданными, встроенные инструменты для анализа данных и машинного обучения, а также мощные механизмы для обеспечения безопасности и резервного копирования.

К преимуществам Oracle относятся высокая производительность, масштабируемость, поддержка больших объемов данных и возможность интеграции с другими корпоративными системами.

## Логическая модель баз данных

Логическая модель баз данных поможет структурировать представление данных, которое позволяет четко определить сущности, их атрибуты и взаимосвязи между ними [8].

* + 1. Таблица Users

Таблица Users предназначена для хранения скрытой информации о пользователе, такой как логин и хеш.

Таблица 3.2.1.1 - Таблица Users

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Login | Логин | Varchar(255) |  |
| Hash | Хеш пароля | Raw(64) |  |
| Salt | Соль генерации Хеша | Raw(64) |  |
| UserType\_Id | Идентификатор Роли пользователя | Bigint | FK к User\_Types |

* + 1. Таблица User\_Type

Таблица User\_Type предназначена для хранения списка ролей.

Таблица 3.2.2.1 - Таблица User\_Type

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Name | Название роли | Varchar(255) |  |

* + 1. Таблица Settlements

Таблица Settlements предназначена для хранения списка областей, доступных для добавления к адресу школы.

Таблица 3.2.3.1 - Таблица Settlements

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Name | Название области | Varchar(255) |  |
| Region\_id | Идентификатор регион области | Bigint | FK к Regions |

* + 1. Таблица Regions

Таблица Regions предназначена для хранения списка регионов, доступных для добавления к адресу школы.

Таблица 3.2.4.1 - Таблица Regions

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Name | Название региона | Varchar(255) |  |

* + 1. Таблица Educational\_Institutions

Таблица Educational\_Institutions предназначена для хранения информации о школе.

Таблица 3.2.5.1 - Таблица Educational\_Institutions

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Name | Название школы | Varchar(255) |  |
| Address | Адрес школы | Varchar(255) |  |
| PathImage | Путь к фотографии | Varchar(255) |  |
| Email | Почта | Varchar(255) |  |
| PhoneNumber | Номер телефона | Varchar(255) |  |
| Settlement\_Id | Идентификатор области | Bigint | FK к Settlements |

* + 1. Таблица Image

Таблица Image предназначена для хранения путей к фотографиям галереи школы.

Таблица 3.2.6.1 - Таблица Image

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| PathImage | Путь к фотографии | Varchar(255) |  |
| EducationalInstitution  \_Id | Идентификатор школы | Bigint | FK к Educational  \_Institutions |

* + 1. Таблица School\_Students

Таблица School\_Students предназначена для хранения информации об ученике школы.

Таблица 3.2.7.1 - Таблица School\_Students

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| User\_Id | Идентификатор пользователя | Bigint | FK к User |
| ClassRoom\_Id | Идентификатор класса | Bigint | FK к Classes |
| EducationalInstitution  \_Id | Идентификатор школы | Bigint | FK к Educational  \_Institutions |
| FirstName | Имя ученика | Varchar(255) |  |
| LastName | Фамилия ученика | Varchar(255) |  |
| Patronymic | Отчество ученика | Varchar(255) |  |
| PathImage | Путь к фотографии | Varchar(255) |  |
| Email | Почта | Varchar(255) |  |
| PhoneNumber | Номер телефона | Varchar(255) |  |

* + 1. Таблица Parents

Таблица Parents предназначена для хранения информации об ученике школы.

Таблица 3.2.8.1 - Таблица Parents

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| User\_Id | Идентификатор пользователя | Bigint | FK к User |
| EducationalInstitution  \_Id | Идентификатор школы | Bigint | FK к Educational  \_Institutions |
| FirstName | Имя ученика | Varchar(255) |  |
| LastName | Фамилия ученика | Varchar(255) |  |
| Patronymic | Отчество ученика | Varchar(255) |  |
| PathImage | Путь к фотографии | Varchar(255) |  |
| Email | Почта | Varchar(255) |  |
| PhoneNumber | Номер телефона | Varchar(255) |  |

* + 1. Таблица Parents\_Types

Таблица Parents\_Types предназначена для хранения списка типов родителей: мать, отец и т. п.

Таблица 3.2.9.1 - Таблица Parents\_Types

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Name | Название типа | Varchar(255) |  |

* + 1. Таблица School\_Students\_And\_Parents

Таблица School\_Students\_And\_Parents предназначена для хранения информации о связи родителя и ребёнка.

Таблица 3.2.10.1 - Таблица School\_Students\_And\_Parents

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| SchoolStudent\_Id | Идентификатор ученика | Bigint | FK к School\_Students |
| Parent\_Id | Идентификатор родителя | Bigint | FK к Parents |
| ParentType\_Id | Идентификатор типа родителя | Bigint | FK r Parents\_Types |

* + 1. Таблица Teacher\_Assignments

Таблица Teacher\_Assignments предназначена для хранения информации о связи учителя предмета и класса, где этот предмет преподаётся.

Таблица 3.2.11.1 - Таблица Teacher\_Assignments

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Teacher\_Id | Идентификатор учителя | Bigint | FK к Teachers |
| SchoolSubject\_Id | Идентификатор предмета | Bigint | FK к SchoolSubjects |
| ClassRoom\_Id | Идентификатор класса | Bigint | FK к Classes |

* + 1. Таблица Administrators

Таблица Administrators предназначена для хранения информации об администраторе школы.

Таблица 3.2.12.1 - Таблица Administrators

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| User\_Id | Идентификатор пользователя | Bigint | FK к User |
| EducationalInstitution  \_Id | Идентификатор школы | Bigint | FK к Educational  \_Institutions |
| FirstName | Имя ученика | Varchar(255) |  |
| LastName | Фамилия ученика | Varchar(255) |  |
| Patronymic | Отчество ученика | Varchar(255) |  |
| PathImage | Путь к фотографии | Varchar(255) |  |
| Email | Почта | Varchar(255) |  |
| PhoneNumber | Номер телефона | Varchar(255) |  |

* + 1. Таблица Teachers

Таблица Teachers предназначена для хранения информации об учителе школы.

Таблица 3.2.13.1 - Таблица Teachers

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| User\_Id | Идентификатор пользователя | Bigint | FK к User |
| EducationalInstitution  \_Id | Идентификатор школы | Bigint | FK к Educational  \_Institutions |
| FirstName | Имя ученика | Varchar(255) |  |
| LastName | Фамилия ученика | Varchar(255) |  |
| Patronymic | Отчество ученика | Varchar(255) |  |
| PathImage | Путь к фотографии | Varchar(255) |  |
| Email | Почта | Varchar(255) |  |
| PhoneNumber | Номер телефона | Varchar(255) |  |

* + 1. Таблица Classes

Таблица Classes предназначена для хранения информации о классе.

Таблица 3.2.14.1 - Таблица Classes

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Name | Название класса | Varchar(255) |  |
| Teacher\_Id | Идентификатор учителя | Bigint | FK к Teachers |

* + 1. Таблица School\_Subjects

Таблица School\_Subjects предназначена для хранения списка предметов, преподаваемых в школе.

Таблица 3.2.15.1 - Таблица School\_Subjects

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Name | Название предмета | Varchar(255) |  |

* + 1. Таблица Groups

Таблица Groups предназначена для хранения информации о группе класса.

Таблица 3.2.16.1 - Таблица Groups

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| TeacherAssignment  \_Id | Идентификатор назначения учителя | Bigint | FK к Teachers\_  Asignments |
| Group\_Name | Название группы | Varchar(255) |  |

* + 1. Таблица Groups\_Members

Таблица Groups\_Members предназначена для хранения информации об учениках группы.

Таблица 3.2.17.1 - Таблица Groups\_Members

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Group\_Id | Идентификатор назначения учителя | Bigint | FK к Teachers\_  Groups |
| School\_Student\_Id | Идентификатор ученика | Bigint | FK к Teachers\_  School\_Students |

* + 1. Таблица News

Таблица News предназначена для хранения информации о новости школы.

Таблица 3.2.18.1 - Таблица News

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| User\_Id | Идентификатор автора новости | Bigint | FK к Users |
| Title | Заголовок новости | String |  |
| Content | Содержание новости | String |  |
| DateTime | Дата публикации | String |  |

* + 1. Таблица New\_Comments

Таблица New\_Comments предназначена для хранения информации о комментарии к определённой новости.

Таблица 3.2.19.1 - Таблица New\_Comments

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| User\_Id | Идентификатор автора комментария | Bigint | FK к Users |
| News\_Id | Идентификатор новости | Bigint | FK к News |
| Content | Содержание комментария | String |  |
| DateTime | Дата публикации | String |  |

* + 1. Таблица Schedule\_Lesson

Таблица Schedule\_Lesson предназначена для хранения информации о предмете на неделе.

Таблица 3.2.20.1 - Таблица Schedule\_Lesson

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Group\_Id | Идентификатор группы | Bigint | FK к Groups |
| Shedule\_Id | Идентификатор дня недели в четверти | Bigint | FK к Schedule |
| TeacherAssignment  \_Id | Идентификатор предмета | Bigint | FK к Teacher\_  Assignment |
| LessonNumber | Номер урока | Bigint |  |

* + 1. Таблица Schedule

Таблица Schedule предназначена для хранения информации о дне и четверти урока.

Таблица 3.2.21.1 - Таблица Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| QuarterNumber | Номер четверти | Bigint |  |
| DayNumber | День недели | Bigint |  |

* + 1. Таблица Messages

Таблица Messages предназначена для хранения информации об отправленных сообщениях.

Таблица 3.2.22.1 - Таблица Messages

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| User\_Sender\_Id | Идентификатор пользователя отправителя | Bigint | FK к Users |
| User\_Getter\_Id | Идентификатор пользователя получателя | Bigint | FK к Users |
| DateTime | Дата отправки | Date |  |
| Message | Содержание | String |  |

* + 1. Таблица Quarter\_Scores

Таблица Quarter\_Scores предназначена для хранения информации об оценках за четверть по предмету.

Таблица 3.2.23.1 - Таблица Quarter\_Scores

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| School\_Subject\_Id | Идентификатор предмета | Bigint | FK к School\_Subjects |
| School\_Student\_Id | Идентификатор пользователя | Bigint | FK к School\_Students |
| QuarterNumber | Номер четверти | Date |  |
| Message | Содержание | String |  |

* + 1. Таблица User\_Comments

Таблица User\_Comments предназначена для хранения информации о комментарии к пользователю.

Таблица 3.2.24.1 - Таблица User\_Comments

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| User\_Sender\_Id | Идентификатор автора комментария | Bigint | FK к Users |
| User\_Getter\_Id | Идентификатор профиля | Bigint | FK к Users |
| Content | Содержание комментария | String |  |
| DateTime | Дата публикации | Date |  |

* + 1. Таблица Gradebook\_Score

Таблица Gradebook\_Score предназначена для хранения информации об оценке за урок.

Таблица 3.2.25.1 - Таблица Gradebook\_Score

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| GradebookDay\_Id | Идентификатор дня урока | Bigint | FK к Gradebook\_Days |
| School\_Student\_Id | Идентификатор пользователя | Bigint | FK к School\_Students |
| Score | Оценка | Bigint |  |

* + 1. Таблица Gradebook\_Attendance

Таблица Gradebook\_Attendance предназначена для хранения информации о посещаемости ученика.

Таблица 3.2.26.1 - Таблица Gradebook\_Attendance

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| GradebookDay\_Id | Идентификатор дня урока | Bigint | FK к Gradebook\_Days |
| School\_Student\_Id | Идентификатор пользователя | Bigint | FK к School\_Students |

* + 1. Таблица Gradebook\_Day

Таблица Gradebook\_Day предназначена для хранения информации о дне урока.

Таблица 3.2.27.1 - Таблица Gradebook\_Day

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Назначение атрибута | Тип данных | Примечание |
| Id | Идентификатор | Bigint | PK |
| Group\_Id | Идентификатор группы | Bigint | FK к Group |
| DateTime | Дата урока | Date |  |
| Topic | Комментарий к уроку | String |  |
| Homework | Содержание домашнего задания | String |  |
| QuarterNumber | Номер четверти | Bigint |  |

Подраздел «Логическая модель баз данных» содержит подробное описание структуры баз данных, используемых в различных микросервисах разрабатываемого приложения. Основная задача этого подраздела — определить принципы организации, хранения и взаимодействия данных между различными компонентами системы. Это позволяет чётко структурировать информацию, обеспечить её целостность и эффективное управление, что является важным условием для стабильной и корректной работы приложения.

## Разработка алгоритмов

Разработка алгоритмов необходима для создания чётких и эффективных инструкций, которые позволяют решать конкретные задачи или проблемы в программных системах. Алгоритмы определяют последовательность действий, необходимых для обработки данных, выполнения вычислений или автоматизации процессов, что способствует оптимизации производительности, снижению ошибок и экономии ресурсов.

3.3.1 Алгоритм добавления пользователя

Основной алгоритм для сервиса, который будет добавлять пользователя на сервер. Для это сперва будет необходимо проверить валидность данных, если данные валидны, то они будут отправлены на сервер для добавления, после чего будет получено сообщение о статусе добавления.

Изображение выглядит как текст, диаграмма, Технический чертеж, План

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.3.1.1 – Алгоритм добавления пользователя

* + 1. Алгоритм добавления комментария

К существующим новостям каждой школы имеется возможность добавления комментариев от каждого пользователя. Для это пользователю нужно будет перейти к соответствующей записи и написать текст комментария, после чего отправить его на сервер, если данные будут валидны, комментарий будет добавлен.

Изображение выглядит как текст, диаграмма, Технический чертеж, План

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.3.2.1 – Алгоритм добавления пользователя

* + 1. Алгоритм добавления класса

При попытке добавления класса в базу данных поступает запрос на получение учителей, у которых ещё нет руководимого класса, после чего формируется модуль, где администратор может заполнить данные о классе и выбрать одно из полученных учителей, после чего отправить данные на сервер.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3.3.3.1 – Алгоритм добавления класса

# Заключение

В ходе выполнения преддипломной практики была проведена работа по анализу, проектированию и разработке электронного дневника, объединяющего функциональность таких популярных платформ, как School.by, Eljur.by и Znaij.by.

Во введении была определена актуальность разработки универсальной платформы, которая объединяет лучшие черты существующих решений и устраняет их недостатки. Была поставлена цель — создание веб-приложения электронный дневник, ориентированного на удобство пользователей и предоставление широких возможностей для введения образовательного процесса. Для достижения этой цели были сформулированы задачи, включающие анализ существующих платформ, определение требований, разработку архитектуры, реализацию основных функций и тестирование. В результате было достигнуто чёткое понимание направления разработки и обоснование необходимости создания нового веб-приложения.

Во втором разделе была разработана UML-диаграмма вариантов использования, а также инфологическая модель базы данных, которая описывает основные сущности и их атрибуты. Для этого были использованы инструменты UML для визуализации функциональных требований. Это дало чёткое представление о том, как пользователи будут взаимодействовать с системой, и как данные будут организованы и связаны между собой.

В третьем разделе была разработана архитектура программного средства, были выбраны технологи архитектуры для разработки, а также разработана логическая модель базы данных. Таким образом в этом разделе была разработана архитектура программного средства, которая обладает высокой производительностью, масштабируемостью и безопасность системы за счет использования современных подходов к проектированию веб-приложений и современных технологий.

# Список использованных источников

[1] SchoolBy [Электронный ресурс]. – Режим доступа : https://schools.by.

[2] EljurBy [Электронный ресурс]. – Режим доступа : https://eljur.by.

[3] ZnaijBy [Электронный ресурс]. – Режим доступа : *https://* znaij.by.

[4] Lucidchart UML [Электронный ресурс]. – Режим доступа : https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language.

[5] Блинов, И. *Java*. Методы программирования / И. Блинов, В. Романчик. – Минск : Четыре четверти, 2013. – 896 с.

[6] *Spring Framework* [Электронный ресурс]. – Режим доступа : *https://javarush.com*.

[7] Куликов, C. Работа с *MySQl, MS SQL Server* и *Oracle* в примерах / C. Куликов – *EPAM Systems*, *RD Dep*, 2021. – 590 с.

[8] Куликов, C. Реляционные базы данных в примерах / C. Куликов – *EPAM Systems*, *RD Dep*, 2021. – 420 с.