Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

**Отчёт**

по предмету «Программирование программного обеспечения»

Лабораторная работа №1

«Электронная школа (интерфейс учителя)»

Студент: Глухова Д.В.

ФИТ 3 курс 2 группа

Преподаватель: Якубенко К.Д.

**Введение**

В данном документе представлены требования к разработке программного обеспечения "Электронная школа (интерфейс учителя)". В условиях современного образовательного процесса использование цифровых технологий становится все более актуальным и необходимым. Электронные школы не только упрощают доступ к учебным материалам, но и значительно повышают эффективность взаимодействия между учениками, преподавателями и родителями. Это создает новые возможности для образовательного процесса, позволяя адаптировать его под индивидуальные потребности каждого учащегося.

Цель проекта заключается в разработке интерфейса для учителей, который будет максимально удобным и функциональным. Мы стремимся создать инструмент, который поможет педагогам эффективно планировать уроки, отслеживать успеваемость учащихся и взаимодействовать с родителями. Это не только упростит рабочие процессы учителей, но и облегчит общение между всеми участниками образовательного процесса.

Кроме того, важной задачей является создание платформы для обмена материалами и идеями между учителями. Это будет способствовать сотрудничеству и поддержке профессионального роста педагогов. Обмен опытом и ресурсами поможет создать сообщество, где учителя смогут вдохновляться идеями друг друга и внедрять их в свою практику.

Современные требования к образованию требуют от педагогов использования различных подходов и технологий. Учителя должны быть готовы адаптироваться к новым условиям и использовать цифровые инструменты для повышения качества образования. Интерфейс, который мы разрабатываем, будет удобным и интуитивно понятным, что позволит учителям сосредоточиться на обучении, а не на технических деталях.

Мы понимаем, что нагрузка на учителей часто очень велика, и поэтому наша задача — сделать интерфейс максимально простым в использовании. Это позволит педагогам быстрее осваивать новые технологии и сосредотачиваться на своих основных обязанностях — обучении и развитии учащихся.

В следующих разделах документа будет подробно рассмотрены требования к разработке, функциональные возможности интерфейса и ожидаемые результаты его внедрения.

1. **Основания для разработки**

В условиях стремительных изменений в образовательной среде, вызванных технологическим прогрессом и пандемией, актуальность внедрения цифровых решений становится очевидной. Современные образовательные учреждения все чаще переходят на дистанционные и смешанные форматы обучения, что создает необходимость в эффективных инструментах для управления учебным процессом. Учителя сталкиваются с множеством рутинных задач, таких как планирование уроков, выставление оценок, ведение журналов и составление отчетов. Эти процессы требуют значительных временных и трудовых затрат. Автоматизация и упрощение этих задач позволяют учителям сосредоточиться на обучении и взаимодействии с учениками, что в итоге улучшает качество образования.

Приложение должно включать ряд ключевых функциональных возможностей. Во-первых, управление расписанием позволит учителям легко создавать, редактировать и адаптировать расписание уроков в зависимости от изменений. Во-вторых, система ведения успеваемости обеспечит возможность выставления оценок, комментирования результатов и отслеживания прогресса каждого ученика. Важным аспектом является возможность коммуникации с учениками и их родителями, что позволит поддерживать обратную связь и оперативно решать возникающие вопросы. Кроме того, наличие инструментов для генерации отчетов и аналитики по успеваемости и активности учащихся поможет учителям принимать обоснованные решения и корректировать образовательные стратегии.

Внедрение приложения "Электронная школа" отвечает современным требованиям к образованию и способствует улучшению взаимодействия между учителями, учениками и родителями. Учителя получают доступ к важной информации в любое время и с любого устройства, что делает процесс управления учебной деятельностью более гибким. Использование цифровых инструментов позволяет более эффективно отслеживать прогресс учеников, выявлять слабые места в обучении и адаптировать подходы к каждому ученику. Это, в свою очередь, способствует более глубокому вовлечению учеников и повышению их мотивации.

Разработка приложения "Электронная школа (интерфейс учителя)" отвечает современным требованиям образовательного процесса и способствует улучшению качества обучения. Это приложение не только повышает эффективность работы учителей, но и создает условия для более активного вовлечения учеников и их родителей в образовательный процесс. Внедрение таких цифровых решений становится важным шагом в направлении модернизации образовательной среды и повышения ее конкурентоспособности.

1. **Назначение разработки**

Функциональным назначением нашей социальной сети является создание платформы, которая будет удовлетворять потребности педагогов и способствовать эффективному обучению в современном образовательном процессе. В условиях стремительного развития технологий и изменений, происходящих в образовательной среде, важно, чтобы учителя имели доступ к инструментам, которые облегчают их работу и улучшают качество образования.

Интерфейс нашей платформы предоставляет учителям возможность организовывать и управлять учебным процессом, отслеживать успеваемость учеников и взаимодействовать с ними в реальном времени. Это позволяет педагогам не только эффективно планировать уроки, но и быстро реагировать на потребности своих учеников. Интерактивные функции платформы помогают создать динамичную и вовлекающую образовательную среду.

Учителя могут публиковать и делиться различными учебными ресурсами, включая текстовые задания, презентации и мультимедийные материалы. Это разнообразие ресурсов способствует более интерактивному обучению, позволяя учащимся осваивать материал с разных ракурсов. Возможность использования мультимедийных элементов помогает сделать уроки более наглядными и интересными, что, в свою очередь, стимулирует познавательную активность учеников.

Платформа также предоставляет возможность создавать группы для работы над проектами и совместными заданиями. Это не только облегчает организацию групповой работы, но и способствует развитию командных навыков у учеников. Совместная деятельность помогает укрепить взаимодействие между учителями и учениками, создавая атмосферу сотрудничества и поддержки. В таких группах учащиеся могут обмениваться идеями, обсуждать задания и совместно находить решения, что имеет важное значение для формирования их личностных и социальных навыков.

Платформа позволяет создавать группы для работы над проектами и совместными заданиями, что способствует развитию командных навыков у учеников и укреплению взаимодействия между учителями.

Цель нашего интерфейса — обеспечить педагогам эффективные и удобные инструменты для организации образовательного процесса, взаимодействия с учениками и обмена опытом с коллегами в цифровой среде.

**2.1 Функциональное назначение**

Разработка включает следующие ключевые функции:

* Регистрация и аутентификация пользователей: создание надежного механизма, который позволяет учителям регистрироваться и аутентифицироваться для доступа к образовательным ресурсам и функциям платформы.
* Управление классами: функционал для создания и управления классами, включая возможность добавления учеников, назначения уроков и отслеживания успеваемости.
* Коммуникация и обратная связь: инструменты для взаимодействия между учителями и учениками, включая чаты, обсуждения и систему уведомлений о заданиях и мероприятиях.
* Анализ успеваемости: функционал для отслеживания и анализа успеваемости учеников, позволяющий учителям получать отчеты и статистику по результатам обучения.

Группы и проекты: Возможность создания групп для совместной работы над проектами, что способствует развитию командных навыков и взаимодействия между учениками.

**2.2 Эксплуатационное назначение**

Приложение "Электронная школа (интерфейс учителя)" предоставляет учителям доступ к широкому спектру образовательных ресурсов через веб-интерфейс и мобильное приложение. Пользователи могут управлять классами, создавать и обмениваться учебными материалами, а также отслеживать успеваемость учеников.

Платформа поддерживает различные образовательные подходы и методы, позволяя учителям адаптировать обучение под нужды своих учеников. Кроме того, приложение может включать платные функции, такие как доступ к дополнительным учебным материалам, онлайн-курсам или специализированным инструментам, что является основной моделью монетизации.

Таким образом, "Электронная школа" обеспечивает эффективные и удобные инструменты для организации учебного процесса и взаимодействия между всеми участниками образовательного процесса.

1. **Требования к программному средству**
   1. **Требования к функциональным характеристикам**

Функционал, доступный пользователям, отличается в зависимости от роли пользователя. В приложении доступны следующие роли:

* гость;
* учитель;
* координатор(контроллер-супервайзер);
* администратор.
  + 1. **Требования к клиентской части**

Клиентская часть должна поддерживать онлайн режим работы. Обеспечивать работу в любом размере окна веб-браузера с адаптацией пользовательского интерфейса. Интерфейс клиентского приложения должен быть информативным и незагруженным. Также обеспечивать выполнение перечисленных ниже функций:

* регистрация пользователя (гость);
* авторизация пользователя (гость);
* ввод и изменение личных данных (учитель, координатор);
* просмотр информации о всех пользователях (администратор);
* блокировка пользователей (администратор);
* удаление нежелательного контента пользователей (администратор);
* создание страницы профиля;
* реагирование на посты пользователей;
* комментирование записей с помощью текста, эмодзи, а также с помощью картинок;
* добавление в подписки, отмена запроса, принятие запроса, удаление запроса, удаление из подписок;
* поиск других пользователей, с последующим сохранением в истории поиска;
* создание постов, используя текст, эмодзи, фон, изображения;
* редактирование данных пользователя.

Клиентская часть должна быть реализована на ReactJs.

* + 1. **Требования к серверной части**

Сервер должен быть создан на базе платформы Node.js с использованием языка программирования JavaScript.

Серверная часть должна обеспечивать следующую функциональность:

* проверка прав пользователя;
* аутентификация и авторизация;
* доступ к базе данных MS SQL;
* защита персональных данных;
* обработка возникающих ошибок;
* предоставление API для работы с клиентской частью приложения.

Перечисленные требования обеспечивают необходимую работоспособность и безопасность серверной части приложения. Это позволяет обеспечить надежное и эффективное функционирование системы.

**3.1.3 Требования к временным характеристикам**

После внесения изменений в данные, находящиеся в базе данных, новая информация должна быть отображена пользователям немедленно, без задержек. Система должна обеспечивать мгновенное обновление данных для пользователей. Это гарантирует плавный и комфортный пользовательский опыт без лишних задержек при взаимодействии с сервисом.

* 1. **Требования к надёжности**

Программное средство должно:

* вероятность безотказной работы;
* обеспечивать проверку корректности данных авторизации пользователя;
* передачу зашифрованных данных от клиента к серверу и наоборот;
* хранение пароля пользователя в базе данных в зашифрованном виде;
* обеспечивать защиту личных данных пользователя от внешнего вмешательства;
* ограничить возможность создания нескольких учетных записей одному и тому же пользователю.
  1. **Условия эксплуатации**

Эксплуатация приложения "Электронная школа" рассчитана на пользователей всех возрастных категорий, что делает его доступным для широкого круга пользователей, независимо от их возраста или пола. Важно, чтобы приложение соответствовало потребностям разнообразной аудитории, включая учащихся, учителей и родителей.

Интерфейс приложения должен быть интуитивно понятным, что позволит пользователям без особых усилий осваивать все функции. Мы стремимся создать дизайн, который будет минималистичным и не перегруженным, что важно для комфортного восприятия информации. Визуальная иерархия элементов интерфейса будет четкой, чтобы пользователи могли быстро находить нужные функции и материалы. Информативность интерфейса также играет ключевую роль: пользователи должны получать необходимую информацию без излишнего усложнения.

Особое внимание будет уделено условиям эксплуатации приложения. Мы понимаем, что не всегда пользователи имеют доступ к стабильному интернет-соединению. Поэтому приложение должно быть спроектировано так, чтобы оно функционировало даже в условиях плохого соединения. Это может быть реализовано через возможность кэширования данных, что позволит пользователям продолжать работу, даже если интернет-соединение временно прерывается.

Приложение также должно быть доступным для людей с нарушением зрения. Это включает в себя использование контрастных цветовых схем и возможность настройки размера шрифта, чтобы текст был читаемым для всех пользователей. Дополнительно, поддержка экранных читалок и других вспомогательных технологий обеспечит удобство для пользователей с различными типами нарушений.

Таким образом, проектируя приложение, мы учитываем разнообразие нашей аудитории и создаем условия, которые способствуют комфортной и эффективной эксплуатации.

* 1. **Требования к составу и параметрам технических и программных средств**

Минимальные рекомендованные аппаратные требования к пользовательской машине:

* процессор архитектуры x86 или arm;
* минимальное количество ядер процессора – 4;
* минимальный объем оперативной памяти – 8 ГБ;
* минимальный доступный объем дискового пространства – 20 ГБ.

Рекомендованные аппаратные требования для серверной машины:

* процессор архитектуры x86 с поддержкой виртуализации;
* количество ядер от 8 до 16;
* минимальный объем оперативной памяти – 16 ГБ;
* минимальный доступный объем дискового пространства – 512 ГБ.

Программные средства требующиеся для работы приложения:

* клиент – операционная система Windows, Linux, MacOS (ПК);
* клиент – операционная система Android, iOS (Мобильный телефон);
* сервер – операционная система Windows, Linux.
  1. **Требования к информационной и программной совместимости**

Приложение клиент и сервер общаются с использованием сети интернет по протоколу HTTP. Приложение сервер общается с СУБД локально (если они расположены на одной машине) или по сети интернет по протоколу RDA.

Приложения для учителей и учеников должны быть разработаны с учетом совместимости, поддерживая одни и те же версии протоколов и стандартов для обмена данными. Это обеспечивает корректное взаимодействие между ними.

Рекомендуется применять стандартизированные протоколы, такие как HTTP, для обмена данными между приложениями и базой данных. Это упрощает разработку, поддержку и обеспечивает совместимость системы.

Для предотвращения появления несанкционированных устройств в сети необходимо использовать средства защиты, такие как фаерволы, контроль доступа и мониторинг сетевой активности. Это помогает избежать несанкционированного доступа и утечек данных.

При обновлении приложений и базы данных важно учитывать совместимость версий. Все компоненты системы должны быть согласованы, чтобы обеспечить совместимость данных и интерфейсов.

**3.6 Требование к маркировке и упаковке**

Программное обеспечение для "Электронной школы" (интерфейс учителя), доступное через интернет, не требует традиционной маркировки и упаковки. Оно загружается с официального сайта разработчика в виде архива или исполняемых файлов.

Однако существуют следующие рекомендации и требования:

* Загрузка с официального сайта: рекомендуется загружать программное обеспечение только с официального сайта разработчика. Это помогает избежать установки поддельных или вредоносных версий программы.
* Проверка контрольных сумм: для подтверждения подлинности программного обеспечения следует предоставлять контрольные суммы (hash-суммы) загруженных файлов на официальном сайте. Пользователи могут сравнивать контрольные суммы с указанными значениями, чтобы удостовериться в целостности и подлинности файлов.

1. **Требования к программной документации**

Программная документация является важным компонентом разработки и эксплуатации программного обеспечения "Электронная школа". Она должна быть представлена в виде отдельных конечных точек приложения как для клиентской, так и для серверной части. Наличие четкой и понятной документации способствует более эффективному взаимодействию между разработчиками, пользователями и другими заинтересованными сторонами.

Внутри кода программы также рекомендуется использовать комментарии. Они помогут другим разработчикам быстро понять логику и структуру кода, а также облегчат процесс его сопровождения и модификации в будущем.

Комментарии должны быть лаконичными и информативными, описывая ключевые функции и алгоритмы, что сделает код более читаемым и поддерживаемым.

Документация для серверной части приложения должна быть представлена в виде детального описания конечных точек API.

Это описание должно включать следующие элементы:

* URL конечной точки: это адрес, по которому разработчики могут обращаться к определенному ресурсу, например, для получения списка пользователей это может быть /api/users;
* название используемого HTTP метода: каждый API-метод должен иметь уникальный URL, который будет использоваться для его вызова. Этот URL должен быть четко задокументирован, чтобы разработчики могли легко понять, как обращаться к различным ресурсам;
* пример данных, передаваемых на сервер по указанной конечной точке: для каждой конечной точки необходимо указать, какой HTTP метод используется (GET, POST, PUT, DELETE и т.д.). Это поможет понять, какой тип операции выполняется: получение данных, создание новых ресурсов, обновление существующих или удаление;
* пример данных, возвращаемых с сервера: важно также предоставить примеры данных, которые будут возвращены с сервера в ответ на запрос.

Для достижения этих целей может быть использована спецификация описания API под названием Open API.

1. **Технико-экономические показатели**

Капитальные затраты включают в себя различные расходы, связанные с разработкой приложения. Оценка затрат на разработку программного обеспечения включает зарплаты разработчиков, дизайнеров и тестировщиков, а также расходы на создание документации и тестирование. Кроме того, необходимо учитывать затраты на аппаратное обеспечение, включая серверы и компьютеры, которые понадобятся для функционирования приложения. Также важными являются расходы на приобретение лицензий для используемых технологий и платформ, что может значительно повлиять на общую стоимость проекта.

Эксплуатационные расходы представляют собой текущие затраты, связанные с поддержанием и улучшением приложения. Эти расходы включают техническую поддержку, что подразумевает зарплаты технического персонала и расходы на обучение. Регулярные обновления программного обеспечения и внедрение новых функций требуют дополнительных финансовых вложений. Хостинг и инфраструктура также являются важными компонентами эксплуатационных расходов, так как они обеспечивают доступность приложения для пользователей.

Экономическая эффективность разработки приложения проявляется в снижении временных затрат учителей на рутинные задачи. Это может быть измерено в часах, сэкономленных благодаря автоматизации процессов, таких как планирование уроков и выставление оценок. Увеличение продуктивности будет означать, что учителя смогут проводить больше уроков за то же время. Также важно учитывать снижение ошибок, связанных с ручным вводом данных, что напрямую влияет на качество обучения и восприятие информации учениками.

Социальная эффективность включает в себя уровень удовлетворенности пользователей, который можно оценить через опросы и исследования. Вовлеченность учеников в процесс обучения также является важным показателем; процент учащихся, активно использующих приложение, может дать представление о его реальной ценности. Обратная связь от пользователей позволит выявить сильные и слабые стороны приложения и определить направления для его дальнейшего улучшения.

Качество обслуживания можно оценить по времени обработки запросов пользователей и уровню поддержки. Среднее время, необходимое для обработки запросов, и процент успешно решенных обращений помогут определить, насколько эффективно приложение справляется с запросами пользователей и насколько удовлетворены они получаемыми услугами.

1. **Стадии и этапы разработки**

Разработка программного обеспечения "Электронная школа" будет проводиться в три ключевых этапа: техническое задание, технический проект и внедрение. Каждый из этих этапов играет важную роль в создании качественного и функционального продукта, соответствующего потребностям пользователей и требованиям заказчика.

На стадии технического задания будет выполнен процесс разработки, согласования и утверждения самого задания. Этот этап включает в себя:

Сбор требований: необходимо провести встречи с заинтересованными сторонами — учителями, администрацией школ и родителями, чтобы определить их потребности и ожидания от приложения. Это поможет сформулировать четкие и конкретные требования к функционалу.

Анализ и документирование: Все собранные требования будут систематизированы и задокументированы в виде технического задания. Этот документ будет содержать описание целей проекта, функциональных возможностей, интерфейса и других критически важных аспектов.

Согласование: после подготовки технического задания оно будет представлено на рассмотрение заинтересованным сторонам для получения обратной связи и внесения необходимых корректировок. Только после окончательного согласования документ будет утвержден.

Разработка должна быть проведена в 3 этапа:

* техническое задание;
* технический проект;
* внедрение.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

* разработка программы;
* разработка программной документации;
* испытание программы.

На этапе разработки программы должна быть выполнена работа по программированию и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

1. **Порядок контроля и приемки**

Этап планирования контроля и приемки начинается с разработки четкого плана, который определяет порядок проведения контрольных мероприятий и критерии приемки приложения. На этом этапе важно установить сроки для каждого из этапов тестирования и приемки, а также определить ответственных лиц. План должен включать этапы, такие как предварительное тестирование, функциональное тестирование, нагрузочное тестирование и приемочное тестирование, что позволит систематически подходить к оценке качества приложения.

Предварительное тестирование проводится на ранних стадиях разработки, чтобы выявить и устранить критические ошибки до начала основного тестирования. Этот этап включает в себя проверку функционала, простоты использования и соответствия первоначальным требованиям. На данном этапе также важно провести тестирование интерфейса, чтобы убедиться, что он интуитивно понятен и удобен для пользователей. Результаты предварительного тестирования фиксируются и анализируются, что позволяет команде разработчиков внести необходимые изменения до следующего этапа.

Функциональное тестирование осуществляется после завершения разработки ключевых функций приложения. На этом этапе проверяется, насколько приложение соответствует заявленным функциональным требованиям, указанным в техническом задании. Тестировщики проводят серию тестов, чтобы убедиться, что все функции работают корректно и без ошибок. Важно проверить взаимодействие различных компонентов приложения, а также его совместимость с различными устройствами и операционными системами. Все найденные ошибки документируются и передаются на доработку.

Нагрузочное тестирование важно для оценки производительности приложения при различных условиях эксплуатации. На этом этапе тестируется способность системы обрабатывать большое количество запросов одновременно, что позволяет выявить потенциальные узкие места и проблемы с производительностью. Результаты нагрузочного тестирования помогают определить, насколько приложение готово к реальной эксплуатации в условиях высокой нагрузки и сколько пользователей оно может обслуживать без потери качества работы.

Контроль после внедрения также играет важную роль в обеспечении качественной работы приложения. После запуска необходимо продолжать мониторинг его работы, собирать отзывы пользователей и выявлять возможные проблемы.

**Заключение**

Разработка программного обеспечения "Электронная школа" представляет собой комплексный и многоступенчатый процесс, который включает в себя три ключевых этапа: формирование технического задания, разработка технического проекта и внедрение. Эти этапы являются основой для создания эффективного инструмента, который будет служить интересам педагогов, учащихся и их родителей. Каждому из этапов отводится своя важная роль, и их качественное выполнение напрямую влияет на успех всего проекта.

На этапе формирования технического задания особое внимание уделяется сбору и анализу требований пользователей. Это критически важный процесс, поскольку именно на этом этапе выявляются нужды и ожидания всех заинтересованных сторон. Участие педагогов, учащихся и родителей в обсуждении функциональных возможностей приложения позволяет создать более точное представление о том, каким должно быть окончательное решение. Проведение опросов, интервью и обсуждений помогает собрать ценную информацию, которая служит основой для дальнейшей работы. Это позволяет избежать недоразумений на более поздних этапах разработки и гарантирует, что конечный продукт будет соответствовать ожиданиям пользователей.

В ходе реализации технического проекта осуществляется разработка самого программного обеспечения, сопровождаемая созданием необходимой документации и проведением тестирования. Эти действия гарантируют высокое качество и стабильность работы приложения.

Этап внедрения включает в себя обучение пользователей и пилотное тестирование, что способствует smooth переходу к новому инструменту в образовательном процессе. Такой подход позволяет не только минимизировать возможные проблемы, но и обеспечить максимальную удовлетворенность пользователей.

В заключение, системный подход к разработке, включающий этапы планирования, реализации и внедрения, позволяет создать надежное и функциональное программное решение, способствующее повышению эффективности образовательного процесса и улучшению взаимодействия между всеми участниками. "Электронная школа" станет важным шагом в интеграции цифровых технологий в образование, открывая новые возможности для обучения и развития.