**Руководство системного программиста электронный дневник студента**

**Введение**

Данное руководство системного программиста предназначено для разработчиков, создающих программный модуль для учета заявок на ремонт бытовой техники. Программа, написанная на языке C#, должна обеспечивать эффективное управление процессами приема, обработки и выполнения заявок. В условиях современного сервиса, где важна скорость и качество обслуживания, такой модуль способен значительно упростить работу как для пользователей, так и для технического персонала. Основная цель — создать систему, которая будет интуитивно понятной, надежной и безопасной.

**1. Архитектура системы**

**1.1 Общая структура**

Аутентификация:

Система должна поддерживать надежную аутентификацию пользователей. Это может включать использование логина и пароля, а также возможность внедрения многофакторной аутентификации для повышения уровня безопасности. Пользователи должны иметь возможность восстанавливать доступ к своим учетным записям через электронную почту или SMS. Это не только повысит безопасность, но и улучшит пользовательский опыт.

Система доступа:

Разный уровень доступа. Таким образом преподаватели смогут отслеживать оценки своих учащихся, ставить новые, исправлять старые.

У учащихся доступ к просмотру своих оценок по предметам, а так-же возможность просматривать расписание.

Хранение данных:

Данные о заявках, пользователях и их взаимодействиях должны храниться в реляционной базе данных. Это позволит организовать данные в структурированном виде и обеспечит возможность выполнения сложных запросов для анализа и отчетности. База данных должна быть нормализована для минимизации избыточности данных и повышения производительности. Рекомендуется использовать такие системы управления базами данных (СУБД), как PostgreSQL или MySQL, которые обеспечивают надежность и масштабируемость.

**1.2 Технологии**

Для разработки модуля используются следующие технологии:

Backend:

Язык C# и Visual Studio Community 2022 также являются отличным выбором для разработки программного модуля для разработки программного модуля для электронного дневника C# обладает высокой производительностью, хорошей поддержкой объектно-ориентированной парадигмы, а также широким набором библиотек и фреймворков, что позволит разработать функциональный и эффективный модуль.  
Visual Studio Community 2022, в свою очередь, обеспечивает удобную среду разработки с множеством инструментов для ускорения процесса создания и отладки кода. Отличный отладчик, поддержка Git, автоматические рефакторинги и другие возможности помогут сделать процесс разработки проекта более удобным и эффективным.  
Сочетание C# и Visual Studio Community 2022 позволит создать модуль, который будет удобным в поддержке, расширении и развитии, что важно для программного обеспечения, предназначенного для мониторинга и анализа работы охранной службы.

Frontend:

Если требуется создание пользовательского интерфейса, можно использовать такие технологии, как HTML/CSS/JavaScript в сочетании с фреймворками для создания динамичных веб-приложений. Это позволит создать современный интерфейс с высокой отзывчивостью.

База данных:

SQL для создания и управления базами данных, использования MySQL или PostgreSQL для хранения и обработки данных.

Выбор конкретной СУБД должен основываться на требованиях проекта и предполагаемой нагрузке.

3. Интеграция и разработка

3.1 Интеграция с другими сервисами

Модуль должен поддерживать интеграцию с другими системами управления сервисом (например, CRM-системами) для автоматизации процессов передачи данных между системами. Это позволит сократить время обработки заявок и повысить общую эффективность работы.

Например возможность масштабируемости проекта в целях использования данного готового решения для разных учебных заведений.

3.2 Разработка приложений

При разработке приложений, можно придерживаться готового решения, которое можно будет внедрить в любой другой сервис с помощью его адаптации с возможность масштабирования.

Разработка должна включать этапы проектирования архитектуры приложения, написания кода, тестирования и развертывания системы. Важно также предусмотреть возможность масштабирования системы в будущем.

3.3 Использование SDK

Предоставление SDK (Software Development Kit) для разработчиков позволит им интегрировать модуль в свои приложения без необходимости глубокого изучения внутренней структуры системы. SDK должен содержать библиотеки и документацию для упрощения работы с API модуля.

4. Разработка и тестирование

4.1 Среда разработки

Рекомендуется использовать современные IDE (интегрированные среды разработки), такие как Code::Blocks или Visual Studio Code, которые поддерживают язык C и предлагают множество инструментов для упрощения разработки.

В процессе разработки важно следовать стандартам кодирования и использовать инструменты статического анализа кода для выявления потенциальных ошибок на ранних этапах.

4.2 Тестирование кода

Юнит-тесты: Проверка отдельных компонентов системы на корректность работы.

Интеграционные тесты: Проверка взаимодействия между различными модулями системы.

Функциональные тесты: Проверка соответствия системы требованиям пользователей.

Автоматизация тестирования поможет сократить время на проверку работоспособности системы при внесении изменений.

5. Безопасность

5.1 Аутентификация и авторизация

Необходимо реализовать многоуровневую аутентификацию пользователей, включая возможность двухфакторной аутентификации (например, через SMS или приложения для генерации кодов). Это значительно повысит уровень безопасности учетных записей пользователей.

Также стоит внедрить механизмы ограничения доступа к различным функциям системы в зависимости от ролей пользователей (например, администраторы должны иметь больше прав по сравнению с обычными пользователями).

5.2 Защита данных

Все данные должны передаваться по защищенным протоколам (например, HTTPS) для предотвращения перехвата информации во время передачи. Хранение конфиденциальных данных (например, паролей) должно осуществляться в зашифрованном виде с использованием современных алгоритмов шифрования.

Важно также предусмотреть регулярные резервные копии базы данных для защиты от потери информации.

5.3 Журналы и мониторинг

Система должна вести журналы всех действий пользователей и администраторов для отслеживания изменений в системе и выявления возможных нарушений безопасности. Мониторинг активности пользователей поможет быстро реагировать на подозрительные действия.

Рекомендуется использовать специализированные инструменты мониторинга для анализа журналов событий и выявления аномалий в работе системы.

Заключение

Разработка электронного дневника учащегося поможет учебным заведениям, таким как школа, колледж, университет. С возможностью масштабируемости и использования его, как готового решения для использования его разным учебным заведениям. В том числе использование относительного нового семейства C позволит получить быструю работу сервиса.