

1 Введение

В современном мире программное обеспечение (ПО) стало неотъемлемой частью нашей повседневной жизни, охватывая широкий спектр задач и областей применения. Однако, для того чтобы разработать и использовать ПО эффективно, необходимо четкое понимание требований к нему. Спецификация требований к ПО представляет собой документ, который формализует и описывает функциональные и нефункциональные характеристики, а также ограничения, предъявляемые к создаваемому программному продукту

1.1 Назначение

Для данной спецификации требований к ПО будем рассматривать продукт «reCode» Платформа автоматизации задач с использованием программных макросов. Этот продукт включает в себя как онлайн-платформу (веб-сайт), так и офлайн-приложение для компьютера, предназначенные для автоматизации рутинных задач с помощью программных макросов.

Номер выпуска данной спецификации требований к ПО: 1.0.

Этот документ адресован следующим типам читателей:

1. Разработчикам: Для разработчиков этот документ определяет требования к функциональности и интерфейсу пользовательского опыта веб-платформы и офлайн-приложения, а также описывает технические детали, связанные с интеграцией программных макросов и созданием мастерской макросов.

2. Менеджерам проектов: Для менеджеров проектов этот документ представляет базу для планирования и контроля процесса разработки, устанавливая ожидания относительно функциональности и качества продукта.

3. Маркетологам: Для маркетологов этот документ определяет ключевые особенности и функциональность продукта, которые могут быть использованы в маркетинговых материалах и презентациях.

4. Пользователям: Для конечных пользователей этот документ предоставляет информацию о функциональности и возможностях продукта, а также ожидания относительно его использования и интерфейса.

5. Тестировщикам: Для тестировщиков этот документ служит основой для создания тестовых сценариев и критериев, необходимых для проверки соответствия разрабатываемого продукта требованиям.

6. Составителям документации: Для составителей документации этот документ определяет основные особенности продукта, которые должны быть отражены в документации пользователя и руководствах.

1.2 Соглашения, принятые в документах

Для стандартов и типографических соглашений в спецификации требований к ПО используются следующие:

1. Заголовки и разделы: Заголовки разделов и подразделов следует выделять жирным шрифтом или использовать их в верхнем регистре. Нумерация разделов и подразделов может быть выполнена в формате "1.1", "1.2" и т. д.

2. Требования: Требования могут быть выделены курсивом или жирным шрифтом для лучшей читаемости. Каждое требование обычно начинается с глагола в инфинитиве для ясного определения действия. Нумерация требований может быть автоматической при использовании специализированного программного обеспечения или принятого формата для последующего добавления новых требований.

3. Примеры и код: Примеры кода или фрагменты программного кода могут быть выделены моноширинным шрифтом или блоком для удобства чтения. Это позволяет читателям легко отличать код от обычного текста.

4. Выделение ключевых слов и терминов: Ключевые слова или термины, имеющие особое значение в контексте документа, могут быть выделены жирным шрифтом или курсивом для удобства обращения к ним.

5. Списки: в списке требований, пунктов или иных элементов удобно использовать маркированные или нумерованные списки для более ясного представления информации.

6. Рисунки и таблицы: Рисунки и таблицы должны быть четко пронумерованы и иметь подписи, указывающие их назначение. Рекомендуется использовать специализированные инструменты для создания и вставки рисунков и таблиц.

7. Сноски и пояснения: Пояснения и сноски могут быть выделены маркерами или номерами и помещены в конце страницы или документа для облегчения чтения и понимания текста.

1.3 Границы проекта

Программное обеспечение (ПО) «reCode» предназначено для предоставления средства автоматизации повседневных задач пользователей. Оно позволяет пользователям создавать и использовать программные макросы для выполнения серии действий одной командой или нажатием клавиши, тем самым упрощая и ускоряя рутинные процессы работы.

Этот продукт непосредственно связан с потребностями пользователей в эффективном выполнении повседневных задач. Предоставление средства автоматизации позволяет пользователям сосредоточиться на более важных задачах, увеличивая их производительность и эффективность.

С точки зрения корпоративных целей, данное ПО может быть интегрировано в рабочие процессы компаний различных отраслей, таких как офисные работы, обработка данных, управление операционными системами и другие. Это помогает улучшить рабочие процессы и оптимизировать бизнес-процессы организации.

С точки зрения бизнес-целей и стратегий, данное ПО направлено на увеличение производительности и эффективности рабочих процессов, что в конечном итоге способствует снижению затрат и повышению конкурентоспособности компании. Автоматизация рутинных задач помогает сократить время на выполнение этих задач, что является важным фактором для достижения стратегических целей организации.

Высокоуровневая сводка главной функциональности данного ПО включает в себя возможность создания и использования программных макросов для автоматизации различных задач в офисной среде, обработке данных, управлении операционными системами и других сферах применения, что делает его важным инструментом для улучшения рабочих процессов и повышения производительности пользователей.

2 Общее описание

Программное обеспечение "Платформа автоматизации рутинных задач с использованием программных макросов" разработано для упрощения и ускорения выполнения повседневных задач пользователей путем создания и использования программных макросов. Оно предоставляет средства для создания, редактирования и исполнения макросов, которые могут автоматизировать широкий спектр задач в офисной среде, обработке данных, управлении операционными системами и других сферах применения. Продукт доступен как онлайн-платформа (веб-сайт), так и офлайн-приложение для компьютера, что обеспечивает гибкость использования в различных средах и условиях.

Среда применения:

Продукт может быть применен в различных сферах деятельности, где требуется автоматизация рутинных задач. Он может использоваться в офисных средах для автоматизации процессов работы с текстовыми документами, электронными таблицами и презентациями. Также он может быть полезен в области обработки данных для выполнения серии операций по обработке и анализу данных. Продукт также может быть востребован в сферах управления операционными системами для автоматизации процессов управления и настройки системы.

Предполагаемая пользовательская аудитория:

Пользовательская аудитория включает в себя широкий круг пользователей, включая офисных работников, аналитиков данных, системных администраторов и всех, кто ищет способы увеличения производительности и оптимизации рабочих процессов путем автоматизации рутинных задач.

Известные ограничения, предположения и зависимости:

Ограничения по совместимости: Продукт может иметь ограничения в совместимости с определенными версиями операционных систем или программного обеспечения, на которых он предполагается использовать.

Ограничения по функциональности: Возможны ограничения в функциональности продукта, в зависимости от конкретных задач и сфер применения.

Предположения о технической компетенции пользователей: Предполагается, что пользователи имеют базовые знания в области использования программного обеспечения и понимают основные принципы работы с программными макросами.

Зависимость от доступности интернета (для онлайн-платформы): Для использования онлайн-платформы необходима доступность интернета.

Зависимость от обновлений и поддержки: Продукт может зависеть от регулярных обновлений и поддержки для обеспечения его эффективной работы и соответствия изменяющимся потребностям пользователей.

2.1 Общий взгляд на продукт

Программное обеспечение «reCode» является новым продуктом, созданным в ответ на потребности пользователей в эффективной автоматизации рутинных задач. Оно не является новой версией существующей системы или заменой существующего приложения, а представляет собой совершенно новый продукт, направленный на решение конкретной проблемы.

Программное обеспечение «reCode» может быть частью более крупной системы, включающей другие инструменты и приложения для работы с данными, управления бизнес-процессами и других задач. Основные интерфейсы между этим ПО и другими компонентами системы могут

включать в себя передачу данных, интеграцию с API других приложений и обмен информацией для координации действий.

Важно понимать, что данное ПО предоставляет специализированный функционал по автоматизации рутинных задач с помощью программных макросов, и его роль в системе может быть напрямую связана с обеспечением эффективной работы и взаимодействия с другими компонентами системы.

2.2 Классы и характеристики пользователей

Офисные работники:

Характеристики: Этот класс пользователей включает в себя сотрудников офиса, которые занимаются рутинными задачами, такими как обработка текстовых документов, электронных таблиц, создание презентаций и т. д. Они могут иметь различный уровень технической подготовки, но все они заинтересованы в увеличении производительности и эффективности своей работы.

Требования: Интуитивно понятный пользовательский интерфейс, простота использования, возможность создания и редактирования макросов без глубоких знаний программирования.

Аналитики данных:

Характеристики: Этот класс пользователей состоит из аналитиков данных, которые занимаются обработкой и анализом больших объемов данных. Они могут использовать продукт для автоматизации повторяющихся операций по обработке и анализу данных, ускоряя процесс и повышая точность результатов.

Требования: Возможность работы с различными типами данных и форматами файлов, высокая производительность обработки данных, возможность интеграции с другими инструментами анализа данных.

Системные администраторы:

Характеристики: Этот класс пользователей включает в себя системных администраторов, ответственных за управление и обслуживание операционных систем и сетевой инфраструктуры. Они могут использовать продукт для автоматизации рутинных задач по управлению и настройке системы.

Требования: Гибкие возможности настройки и автоматизации системных операций, возможность работы с различными операционными системами.

Программисты и разработчики:

Характеристики: Этот класс пользователей включает в себя программистов и разработчиков, которые могут использовать продукт для создания сложных макросов и автоматизации специфических задач, требующих программирования.

Требования: Расширенные возможности настройки и программирования макросов, поддержка различных языков программирования.

Привилегированными классами пользователей могут быть программисты и разработчики, которые имеют доступ к расширенным функциональным возможностям продукта и могут создавать более сложные макросы и интеграции.

2.3 Операционная среда

Аппаратная платформа:

Компьютеры с процессорами Intel Core i5 или выше (или их аналоги AMD), с 8 ГБ оперативной памяти и SSD-накопителями для ускорения загрузки и выполнения программ.

Операционные системы и их версии:

Windows 10 (версия 1903 и выше), MacOS Catalina (версия 10.15 и выше), Ubuntu Linux (версия 20.04 и выше).

Серверы и базы данных:

Сервера и базы данных будут размещены в облаке для обеспечения доступности и масштабируемости. Провайдеры облачных услуг могут включать Cloud.ru.

Базы данных и сервера могут быть размещены в дата-центрах облачных провайдеров, а веб-сайт будет доступен через интернет.

Другие компоненты ПО или приложения:

1. Браузеры: Google Chrome, Яндекс Firefox, Safari.
2. Пакеты офисных программ: P7, LibreOffice, Мой Оффис.
3. Среды разработки: JetBrains WebStorm.
4. Системы управления базами данных: MySQL, PostgreSQL.
5. Виртуализация: Docker.
6. Системы контроля версий: Git.

Требования к инфраструктуре:

Для обеспечения высокой доступности и отказоустойчивости системы, необходимо настроить резервное копирование данных, репликацию баз данных и механизмы мониторинга для оперативного реагирования на проблемы.

Для обеспечения безопасности данных необходимо реализовать механизмы аутентификации и авторизации пользователей, шифрование данных в покое и в движении, а также регулярное обновление системных компонентов и мониторинг безопасности.

Для обеспечения масштабируемости и производительности необходимо регулярно оптимизировать запросы к базам данных, использовать

кэширование данных и масштабировать инфраструктуру по мере роста нагрузки.

Для обеспечения совместимости с другими системами необходимо следить за обновлениями и патчами, выпускаемыми разработчиками соответствующих программных продуктов.

2.4 Ограничения дизайна и реализации

На данный момент существует код на языке Kotlin для бэкенда, то логично продолжить использовать Kotlin, чтобы не вносить лишнюю сложность переписывания существующего кода. Для фронтенда выбран JavaScript в связке с HTML и CSS для стандартизации технологий.

Требования к производительности и масштабируемости:

Kotlin является языком, обеспечивающим высокую производительность, что подходит для бэкенда. Для фронтенда JavaScript обеспечивает хорошую производительность в веб-приложениях, а использование MySQL и PostgreSQL в качестве баз данных обеспечивает эффективное хранение данных.

Требования к безопасности:

Kotlin известен своей безопасностью, что важно для бэкенда, особенно при обработке чувствительных данных. Для фронтенда следует обеспечивать безопасность через бев-токены, HTTPS и другие стандарты безопасности веб-приложений.

Требования к совместимости и интеграции:

MySQL и PostgreSQL являются широко используемыми реляционными базами данных, обеспечивая хорошую совместимость с другими системами. Выбор языка программирования и фреймворков также должен быть

ориентирован на возможность интеграции с популярными сторонними сервисами.

Ограничения бюджета и ресурсов:

Учитывая возможные ограничения бюджета, важно выбирать технологии с учетом их стоимости лицензирования, поддержки и обучения персонала.

Требования к сообществу и поддержке:

Kotlin и JavaScript обладают активными сообществами разработчиков, что обеспечивает доступ к богатому опыту и ресурсам. Важно удостовериться в том, что используемые технологии поддерживаются и обновляются сообществом.

Технические ограничения среды разработки:

Выбор среды разработки (например, IntelliJ IDEA для Kotlin и Visual Studio Code для JavaScript) может ограничить поддерживаемые технологии и языки программирования. Важно выбирать среды, которые лучше соответствуют требованиям проекта.

2.5 Предположения и зависимости

При создании программного обеспечения для описанного проекта, мы также можем выделить ряд зависимостей от внешних факторов или компонентов:

Языки программирования и технологические стеки: Выбор языков программирования, фреймворков и библиотек зависит от их доступности, актуальности и совместимости с целями проекта. Изменения в языках программирования или технологических трендах могут потребовать адаптации кода и инфраструктуры проекта.

Операционные системы и среды выполнения: Работа программного обеспечения зависит от совместимости с различными операционными системами (Windows, Linux, MacOS) и средами выполнения (например, .NET Framework, Node.js).

Библиотеки и внешние модули: Использование сторонних библиотек и модулей для реализации функциональности проекта. Зависимости от их актуальности, совместимости и обновлений.

Интеграция с внешними сервисами и API: Некоторые функции проекта могут зависеть от доступности и стабильности внешних сервисов и API, таких как сервисы облачных хранилищ, платежные системы, геоданные и т.д.

Базы данных: Работа с данными и их хранение может зависеть от выбора и совместимости с различными типами баз данных (SQL, NoSQL) и их управления.

Безопасность и защита данных: Реализация мер безопасности зависит от актуальности и надёжности используемых технологий и инструментов для обеспечения защиты от внешних угроз.

Требования к аппаратному обеспечению: Работоспособность программного обеспечения может зависеть от требований к аппаратному обеспечению, таким как процессор, оперативная память, дисковое пространство и т.д.

Лицензии и авторские права: В зависимости от используемых библиотек и инструментов, а также от специфики проекта, может требоваться учёт лицензионных ограничений и авторских прав.

Обновления и поддержка: Необходимость обновлений и поддержки программного обеспечения для обеспечения его актуальности и работоспособности в долгосрочной перспективе

3 Функции системы

Управление макросами:

1. Создание макросов на различных языках программирования (VBA, JS и другие).
2. Редактирование и тестирование макросов.
3. Импорт и экспорт макросов.
4. Объединение нескольких макросов в один макрос.
5. Мастерская макросов:
6. Магазин готовых макрос решений.
7. Простой конструктор макросов для создания простых решений без программирования.
8. Возможность заказа индивидуальных макросов.
9. Поддержка различных областей применения:
10. Интеграция с офисными программами (Р7-офис, МойОфис, OnlyOffice, AlterOffice, LibreOffice, OpenOffice).
11. Поддержка программ обработки данных, таких как Matlab.
12. Автоматизация управления операционными системами (Linux, macOS, Windows).
13. Интеграция с графическими редакторами (Photoshop, Lightroom, Encaps).
14. Поддержка умного дома для автоматизации задач в бытовых условиях.

Функции управления ресурсами:

1. Распределение и сортировка файлов и данных в различных системах и форматах.
2. Автоматическая обработка и объединение блоков данных в АСУ.
3. Управление умным домом для выполнения различных команд (включение света, открытие ворот и т.д.).
4. Платформы и режимы работы:

5. Онлайн-платформа (веб-сайт) и офлайн-приложение для компьютера.
6. Поддержка использования в условиях с плохим доступом к интернету (например, крайний север).

3.1 Функция системы X

3.1 Поддержка различных языков макросов

3.1.1 Перевод макросов на VBA

3.1.2 Перевод макросов на JS

3.1.3 Перевод макросов на другие популярные языки

3.2 Мастерская макросов

3.2.1 Магазин готовых макрос решений

3.2.2 Простой конструктор макросов

3.2.3 Заказ индивидуальных макросов

3.3 Интеграция с различными областями применения

3.3.1 Интеграция с офисными программами

3.3.2 Поддержка программ обработки данных

3.3.3 Автоматизация управления операционными системами

3.4 Управление ресурсами

3.4.1 Распределение файлов и данных

3.4.2 Объединение блоков данных в АСУ

3.1 Описание

Поддержка различных языков макросов

Функция системы: Обеспечение возможности перевода макросов на различные языки программирования, включая VBA, JS и другие популярные языки.

Приоритет: Высокий

Обоснование приоритета: Эта функция является ключевой для пользователей, так как она обеспечивает универсальность системы и поддержку разнообразных языков программирования, что повышает ее ценность и гибкость.

Мастерская макросов

Функция системы: Создание магазина готовых макрос решений с простым конструктором макросов и возможностью заказа индивидуальных макросов.

Приоритет: Средний

Обоснование приоритета: Эта функция обеспечивает удобство использования для широкого круга пользователей, но ее важность несколько ниже, чем у функции поддержки различных языков макросов.

Интеграция с различными областями применения

Функция системы: Интеграция с офисными программами, программами обработки данных, операционными системами и графическими редакторами.

Приоритет: Высокий

Обоснование приоритета: Эта функция придает системе универсальность и расширяет ее область применения, что делает ее более привлекательной для широкого круга пользователей.

Управление ресурсами

Функция системы: Распределение и управление файлами и данными.

Приоритет: Средний

Обоснование приоритета: Эта функция важна, но ее влияние на основные возможности системы несколько меньше, чем у функций, описанных выше.

3.2 Функциональные требования

Функциональные требования для функции "Поддержка различных языков макросов":

FT1: Перевод макросов на VBA

ПО должно обеспечивать возможность конвертации макросов из других языков программирования в VBA.

При ошибке конвертации, ПО должно выдавать сообщение об ошибке с описанием проблемы и предложением решения.

FT2: Перевод макросов на JS

ПО должно поддерживать перевод макросов из других языков программирования в JavaScript.

При обнаружении синтаксических ошибок или несовместимости с JS, ПО должно предупреждать пользователя и предоставлять возможность исправить ошибки.

FT3: Перевод макросов на другие популярные языки

ПО должно иметь функционал для перевода макросов на другие популярные языки программирования.

При неподдерживаемом языке, ПО должно информировать пользователя о невозможности перевода и предложить альтернативные варианты.

Реакция на ожидаемые ошибки, неправильный ввод информации или неверные действия:

При возникновении ошибок в процессе перевода макросов, ПО должно информировать пользователя о возможной причине и предоставить инструкции по исправлению.

Для неправильного ввода информации (например, неверное указание языка программирования), ПО должно выдавать соответствующее уведомление и предложить исправить информацию.

При неверных действиях (например, попытке перевода макроса, несовместимого с выбранным языком), ПО должно предупредить

пользователя о возможных последствиях и предложить альтернативные варианты действий.

4 Требования к данным

1. Входная информация:

a. Макросы на различных языках программирования: Система будет принимать макросы, написанные на различных языках программирования, таких как VBA, JavaScript, Python, Ruby и другие.

b. Информация о задачах и проектах: Если система интегрируется с управлением задачами или проектами, то входной информацией могут быть данные о задачах, их статусах, сроках выполнения и т.д.

c. Файлы и данные: Входные данные могут включать в себя файлы различных форматов (текстовые, таблицы, изображения) и структурированные данные, которые необходимо обработать или преобразовать с помощью макросов.

2. Обработка информации:

a. Конвертация макросов: Система должна обрабатывать входные макросы, конвертируя их из одного языка программирования в другой в соответствии с запросами пользователя.

b. Анализ и исполнение: После конвертации макросы должны быть анализированы и исполнены согласно их функциональности, что может включать в себя автоматизацию повторяющихся задач в офисных приложениях, обработку данных или управление системами.

c. Обработка ошибок: Система должна обрабатывать возможные ошибки при выполнении макросов, предоставляя пользователю информацию о проблемах и возможных путях их решения.

3. Выходная информация:

а. Результаты выполнения макросов: Выходная информация включает в себя результаты выполнения макросов, такие как созданные файлы, обработанные данные, измененные параметры системы и т.д.

б. Сообщения об успешном выполнении или ошибках: Система должна возвращать сообщения пользователю об успешном выполнении макросов или о возникших ошибках в процессе их выполнения.

с. Статус задач и проектов: Если система интегрируется с управлением задачами или проектами, то она может возвращать информацию о статусе задач и проектов после выполнения соответствующих макросов.

д. Журнал действий: Важным аспектом выходной информации может быть составление журнала действий, включающего информацию о выполненных макросах, времени выполнения, возникших ошибках и т.д.

4.1 Логическая модель данных

Требования к структуре БД:

БД должна быть реляционной и иметь нормализованную структуру данных.

Должны быть созданы таблицы для каждой сущности из модели данных, а также связи между ними.

Требования к хранению данных:

Должна быть обеспечена целостность данных с помощью ограничений (например, уникальные ключи, внешние ключи) и правил целостности.

Необходимо учитывать требования к производительности при проектировании структуры БД и выборе индексов.

Требования к безопасности данных:

Должны быть реализованы меры безопасности для защиты данных от несанкционированного доступа и изменения.

Для этого могут использоваться авторизация и аутентификация пользователей, ролевые права доступа, шифрование данных и другие методы.

Требования к производительности:

БД должна обеспечивать высокую производительность при выполнении запросов на чтение и запись данных.

Должны быть оптимизированы запросы к БД, используемые индексы и другие средства для повышения эффективности работы системы.

Требования к резервному копированию и восстановлению:

Необходимо регулярно создавать резервные копии данных для обеспечения их сохранности в случае сбоев или аварийных ситуаций.

Должны быть разработаны и документированы процедуры восстановления данных из резервных копий.

Требования к масштабируемости:

Структура БД должна быть спроектирована с учетом возможности масштабирования по объему данных и количеству пользователей.

Должны быть использованы средства для горизонтального и вертикального масштабирования БД.

Требования к резервированию и обновлению БД:

Необходимо разработать процедуры и средства для обновления и поддержания структуры БД в актуальном состоянии.

Должны быть учтены требования к совместимости данных при обновлении версий приложения или изменении модели данных.

4.3 Отчеты

Отчет о выполненных макросах:

Характеристики: Этот отчет будет содержать информацию о всех выполненных макросах за определенный период времени, например, за день, неделю или месяц. Он может включать в себя дату и время выполнения макроса, его название, результат выполнения (успешно/неуспешно), а также любые ошибки или предупреждения, которые возникли в процессе выполнения.

Отчет о частоте использования макросов:

Характеристики: Этот отчет позволит пользователям увидеть, как часто определенные макросы используются в их системе. Он может содержать информацию о количестве выполнений для каждого макроса за определенный период времени, а также процентное соотношение использования макросов относительно общего числа выполнений.

Отчет о расходе времени на выполнение макросов:

Характеристики: Этот отчет будет показывать, сколько времени занимает выполнение каждого макроса. Он может включать в себя среднее время выполнения для каждого макроса, общее время выполнения всех макросов за определенный период времени, а также максимальное и минимальное время выполнения.

Отчет о результатах макросов:

Характеристики: Этот отчет будет содержать подробную информацию о результатах выполнения каждого макроса. Например, если макрос выполняет какие-то вычисления или обработку данных, отчет может содержать полученные результаты в виде таблицы или графика.

Отчет о статусе макросов:

Характеристики: Этот отчет позволит пользователям быстро оценить текущий статус всех макросов в их системе. Он может содержать информацию о том, какие макросы активны, отключены или требуют обновления.

4.4 Получение, целостность, хранение и утилизация данных

Инициализация данных:

При открытии канала передачи данных может потребоваться выполнить начальный дамп всех данных в принимающую систему. Это позволит установить базовое состояние данных и обеспечить целостность при передаче только изменений в дальнейшем.

Обновление данных:

После инициализации система должна использовать каналы для передачи только изменений данных. Это позволит сократить объем передаваемой информации и снизить нагрузку на сеть.

Защита целостности данных:

Все передаваемые данные должны быть защищены от несанкционированного доступа и модификаций. Для этого могут использоваться методы шифрования данных и аутентификации пользователей.

Для обнаружения и восстановления поврежденных данных могут применяться методы контрольных сумм и цифровых подписей.

Процедуры обслуживания данных:

Регулярное резервное копирование данных должно проводиться с учетом их важности и частоты изменений.

Создание контрольных точек (checkpointing) может быть использовано для сохранения состояния данных на определенных этапах обработки или передачи.

Зеркальное отображение данных на резервного сервера может обеспечить отказоустойчивость и повысить надежность системы.

Политики хранения и утилизации данных:

Должны быть определены политики хранения и утилизации различных типов данных, включая временные данные, метаданные, удаленные записи и данные в кеше.

Удаленные записи могут быть сохранены для аудита или восстановления данных в случае необходимости, но они также должны быть утилизированы в соответствии с правилами безопасности и конфиденциальности.

Данные в кеше и локальные копии должны быть регулярно синхронизированы с основным источником данных, чтобы избежать утери обновленной информации.

Архивация данных должна проводиться с учетом требований к хранению и возможности восстановления информации в будущем.

5.1 Пользовательские интерфейсы

Ссылки на стандарты графического интерфейса:

Интерфейс должен соответствовать стандартам графического дизайна, таким как Material Design для веб-приложений или графический интерфейс операционных систем для настольных приложений.

Стандарты шрифтов, значков, названий кнопок и изображений:

Использование стандартных шрифтов, значков и изображений в соответствии с рекомендациями по UI/UX дизайну для обеспечения единообразия и узнаваемости интерфейса.

Размер и конфигурация экрана:

Интерфейс должен быть адаптирован к различным размерам и разрешениям экранов, включая мобильные устройства, планшеты и настольные компьютеры.

Стандартные кнопки, функции и ссылки перемещения:

Присутствие стандартных кнопок, функций и ссылок перемещения на всех экранах приложения для обеспечения единообразия и удобства навигации.

Сочетания клавиш:

Предоставление сочетаний клавиш для выполнения операций, что упрощает и ускоряет работу пользователя с системой.

Стандарты отображения и текста сообщений:

Отображение сообщений и уведомлений должно соответствовать стандартам, быть понятным и информативным для пользователей.

Стандарты проверки данных:

Настройка стандартов проверки данных, включая ограничения на вводимые значения и проверку содержимого полей, чтобы обеспечить корректность и целостность данных.

Специальные возможности для пользователей с ограничениями:

Разработка специальных возможностей для пользователей с проблемами со зрением, различением цвета и другими ограничениями, таких как возможность увеличения размеров шрифтов или использование специальных цветовых схем.

5.2 Интерфейсы ПО

Связь с базой данных:

Продукт будет взаимодействовать с базой данных (например, MySQL версии 8.0) для хранения и извлечения информации. Форматы данных могут включать структурированные таблицы с различными типами данных (текст, числа, даты и т. д.). Сообщения могут содержать SQL-запросы для извлечения, обновления или вставки данных.

Связь с внешними веб-сервисами:

Продукт может обмениваться данными с внешними веб-сервисами, например, с использованием REST API других приложений (например, версии 2.0). Форматы данных могут быть JSON или XML. Сообщения могут содержать HTTP-запросы и ответы с данными в указанных форматах.

Связь с другими приложениями:

Продукт может интегрироваться с другими приложениями, например, с помощью протоколов обмена данными, таких как SOAP или MQTT. Форматы данных и сообщений будут зависеть от спецификаций этих протоколов и используемых API.

Связь с операционной системой:

Продукт будет работать на операционных системах (например, Windows 10 или Linux Ubuntu). Взаимодействие может включать работу с файловой системой, управление процессами и т. д.

Преобразование данных:

При перемещении данных между системами могут потребоваться преобразования форматов (например, из JSON в XML) или преобразования типов данных (например, из строки в число).

Нефункциональные требования:

Для обеспечения безопасности, могут использоваться меры аутентификации и авторизации при обмене данными с внешними системами.

Для обеспечения уровней обслуживания времени и частоты отклика могут быть установлены SLA (Service Level Agreements).

Общие данные:

Общий доступ к данным может потребоваться для синхронизации информации между различными компонентами или для обеспечения целостности данных.

5.4 Коммуникационные интерфейсы

Для уточнения требований функций взаимодействия, включая электронную почту, веб-браузер, сетевые протоколы и электронные формы, а также определения соответствующих форматов сообщений и особенностей безопасности, скорости передачи данных и механизмов согласования и синхронизации, следует учесть следующие аспекты:

Электронная почта:

Формат сообщений: Стандартный формат электронных писем (например, MIME).

Особенности безопасности: Шифрование писем с использованием протоколов TLS или PGP.

Ограничения: Допустимые типы вложений и их максимальный размер.

Веб-браузер:

Формат сообщений: HTTP/HTTPS запросы и ответы.

Особенности безопасности: Использование HTTPS для защищенной передачи данных.

Ограничения: Поддерживаемые браузером форматы контента (HTML, CSS, JavaScript и т.д.).

Сетевые протоколы:

Формат сообщений: Зависит от протокола (например, TCP/IP, UDP, HTTP, MQTT и т.д.).

Особенности безопасности: Использование протоколов с шифрованием данных (например, SSH для TCP/IP).

Ограничения: Максимальный размер передаваемых пакетов, поддерживаемые протоколом операции (например, GET, POST для HTTP).

Электронные формы:

Формат сообщений: Данные формы могут быть отправлены в различных форматах (например, JSON, XML).

Особенности безопасности: Валидация введенных данных для предотвращения атак вроде SQL-инъекций или XSS.

Ограничения: Максимальные размеры полей формы, типы разрешенных данных (текст, числа, файлы и т.д.).

Особенности безопасности:

Шифрование данных: Использование SSL/TLS для защищенной передачи данных по сети.

Аутентификация: Механизмы аутентификации пользователей и авторизации доступа к ресурсам.

Ограничение доступа: Управление правами доступа к функциям и данным в зависимости от ролей пользователей.

Скорость передачи данных:

Определение минимальных требований к скорости передачи данных в зависимости от функциональных потребностей продукта.

Механизмы согласования и синхронизации:

Использование технологий синхронизации данных, таких как WebSockets или AJAX для обновления данных в реальном времени.

Реализация механизмов согласования изменений данных при работе с распределенными системами.

6. Атрибуты качества

6.1 Удобство использования

Легкость изучения:

Интуитивно понятный интерфейс, который не требует длительного обучения новым пользователям.

Понятные и информативные подсказки и инструкции для помощи пользователям в изучении функций продукта.

Возможность доступа к ресурсам обучения, таким как видеоуроки или документация, для новых пользователей.

Простота использования:

Минималистичный дизайн интерфейса с четко выделенными функциональными элементами.

Использование понятных и логически организованных меню и навигации.

Минимизация количества шагов для выполнения задач пользователя.

Предотвращение ошибок и восстановление:

Использование предупреждений и подсказок для предотвращения потенциальных ошибок при вводе данных или выполнении операций.

Возможность отмены действий и восстановления предыдущих состояний продукта.

Четкое информирование пользователя о возможных последствиях действий.

Эффективность взаимодействия:

Оптимизированный рабочий процесс с использованием контекстных меню, горячих клавиш и быстрых действий.

Предоставление возможности настройки интерфейса в соответствии с потребностями конкретного пользователя.

Использование механизмов автоматизации для ускорения выполнения рутинных задач.

Специальные возможности:

Поддержка различных режимов работы, таких как режим для людей с ограниченными возможностями или режим для экспертов.

Возможность настройки интерфейса и функционала в зависимости от потребностей конкретного пользователя.

Предоставление дополнительных функций и возможностей, которые повышают эффективность использования продукта.

6.2 Производительность

Для обеспечения производительности продукта следует учитывать следующие конкретные требования к производительности для различных системных операций:

Загрузка и запуск приложения:

Время загрузки приложения не должно превышать 30 секунд.

Время запуска приложения на различных операционных системах (Windows, macOS, Linux) должно быть сопоставимым.

Отклик интерфейса:

Задержка при открывании окон, выпадающих списков, контекстных меню и других интерфейсных элементов не должна превышать 97 миллисекунд.

Время отклика при взаимодействии с интерфейсом (например, щелчок по кнопке, переключение вкладок) не должно превышать 15 миллисекунд.

Обработка данных:

Время выполнения основных операций обработки данных (например, поиск, фильтрация, сортировка) не должно превышать 17 секунд для набора данных определенного размера.

При выполнении сложных вычислений или операций с большим объемом данных необходимо обеспечить оптимальное использование ресурсов системы, чтобы избежать перегрузок и зависаний приложения.

Передача данных по сети:

Скорость передачи данных по сети (например, загрузка файлов, синхронизация данных с сервером) должна соответствовать стандартам производительности сети и не создавать задержек в работе приложения.

Обновление и обслуживание приложения:

Время выполнения процессов обновления и обслуживания приложения (например, установка обновлений, очистка кэша) должно быть минимальным и не создавать прерываний в работе пользователя.

6.4 Техника безопасности

Аутентификация и авторизация:

Продукт должен обеспечивать механизмы аутентификации пользователей перед доступом к конфиденциальным данным или функциям.

Должны существовать строгие политики управления доступом, определяющие права доступа пользователей к различным ресурсам продукта в зависимости от их ролей и обязанностей.

Шифрование данных:

Все конфиденциальные данные, передаваемые между клиентом и сервером или хранящиеся на сервере, должны быть зашифрованы с использованием современных криптографических методов.

Должны существовать механизмы защиты данных в покое и в движении для предотвращения несанкционированного доступа к ним.

Физическая безопасность:

Для серверных систем, хранящих конфиденциальные данные, должны применяться меры физической безопасности, такие как ограниченный доступ к серверным помещениям, видеонаблюдение и биометрическая идентификация.

Защита от вредоносного ПО:

Продукт должен обеспечивать механизмы защиты от вредоносного программного обеспечения, такие как антивирусные и антишпионские программы, для защиты от угроз безопасности.

Аудит и мониторинг:

Должны существовать механизмы аудита и мониторинга, позволяющие отслеживать и регистрировать все действия пользователей, имеющих доступ к конфиденциальным данным или функциям продукта.

Обеспечение соответствия законодательству о защите данных:

Продукт должен соответствовать всем применимым законодательным требованиям о защите данных, таким как GDPR (Общий регламент по защите данных).

Обновления и патчи:

Продукт должен регулярно обновляться и патчироваться для устранения уязвимостей безопасности и обеспечения актуальной защиты от известных угроз.

8. [Остальные требования]

Юридические и законодательные требования:

Продукт должен соответствовать всем применимым юридическим и законодательным требованиям, включая авторские права, лицензирование и защиту данных пользователей.

Продукт должен соблюдать международные стандарты безопасности информации, такие как GDPR, HIPAA, PCI DSS и другие

Финансовые требования:

Продукт должен быть доступен по разумной цене для целевой аудитории, обеспечивая оптимальное соотношение цены и качества.

Продукт должен предоставлять механизмы для оплаты и выставления счетов в соответствии с финансовыми политиками компании.

Требования к установке, конфигурированию и обслуживанию:

Продукт должен иметь простой процесс установки и настройки для конечных пользователей.

Должна быть предусмотрена возможность обновления и патчирования продукта через автоматические механизмы обновления.

Должны быть предоставлены инструкции по установке, конфигурированию и обслуживанию продукта для администраторов системы.

Журналирование, мониторинг и контроль следа:

Продукт должен обеспечивать механизмы для журналирования событий, ошибок и действий пользователей для последующего анализа, и мониторинга.

Должны быть предоставлены средства для мониторинга производительности и доступности продукта, а также уведомления об аварийных ситуациях.

Требования к миграции данных:

Если применимо, продукт должен обеспечивать средства для миграции данных с предыдущей системы на новую.

Должны быть предоставлены инструменты для преобразования и импорта данных из различных источников.

Приложение А. Словарь терминов

ПО - программное обеспечение. Это набор программных инструкций, которые выполняют определенные функции на компьютере или другом устройстве.

Макрос - набор заранее записанных команд или действий в программе, которые могут быть выполнены одной командой или нажатием клавиши.

VBA - Visual Basic for Applications. Это программный язык и интегрированная среда разработки от Microsoft, используемая для создания макросов в приложениях Microsoft Office.

JS - JavaScript. Это высокоуровневый язык программирования, который часто используется для создания интерактивных веб-страниц.

ASU - Автоматизированная система управления. Это система, которая автоматизирует управленческие процессы в организации, обычно с использованием программного обеспечения и баз данных.

FBO - Функциональный блок оператора. Это элемент или модуль в автоматизированной системе управления, который выполняет определенные функции или операции.

API - Application Programming Interface. Это набор соглашений и инструментов, который позволяет различным программам взаимодействовать друг с другом.

GDPR - General Data Protection Regulation. Общий регламент по защите данных, нормативный акт Европейского союза, регулирующий сбор и обработку персональных данных.

HIPAA - Health Insurance Portability and Accountability Act. Закон о портативности и ответственности медицинских записей, закон США, регулирующий конфиденциальность медицинских данных.

PCI DSS - Payment Card Industry Data Security Standard. Стандарт безопасности данных платежной индустрии, устанавливающий требования к защите данных платежных карт.

