**Спецификация требований.**

**1.1. Назначение**

Документ описывает требования к программному обеспечению, разрабатываемому в рамках проекта, направленного на создание приложения с расширенным функционалом, включающим шифрование, численные методы. Цель документа- формализовать технические и функциональные ожидания от системы для всех участников разработки: программистов, тестировщиков, а также других заинтересованных сторон, включая тех, кто будет заниматься поддержкой и сопровождением ПО.

**1.2. Область применения**

Приложение предназначено для использования в образовательной, исследовательской и прикладной сферах. Оно реализуется в виде клиент-серверного решения, предоставляющего пользователю инструменты для: шифрования и расшифровки текстовых данных; хеширования информации с помощью алгоритма SHA-384; численного решения уравнений методом хорд; внедрения скрытых сообщений в мультимедийные объекты.

Система может применяться как в учебных курсах по криптографии, алгоритмам и программированию, так и в задачах, связанных с защитой информации или анализом данных.

**1.3. Термины и сокращения**

Для удобства понимания ниже приведены определения основных терминов и сокращений, используемых в документе:

**SHA-384** (Secure Hash Algorithm 384) -криптографическая хеш-функция, формирующая уникальное 384-битное значение (48 байт) для заданных данных.

**Метод хорд**- численный алгоритм, используемый для приближённого нахождения корней нелинейных уравнений, основанный на построении секущих (хорд).

**GUI** (Graphical User Interface)- графический пользовательский интерфейс, обеспечивающий визуальное взаимодействие пользователя с программой.

**БД**- база данных, в которой сохраняются пользовательские и системные данные.

**ПО**- программное обеспечение.

**1.4. Связанные документы**

В рамках проекта дополнительно разработаны вспомогательные материалы, содержащие визуальное и техническое описание компонентов системы:

Диаграмма классов, описывающая архитектуру клиентской и серверной части;

Диаграммы прецедентов использования (Use Case), отображающие сценарии взаимодействия пользователей с системой;

Документ, описывающий подходы и стратегию тестирования приложения.

**2. Общее описание**

**2.1. Назначение и архитектура**

Проектируемое приложение реализуется как клиент-серверная система с графическим интерфейсом. Пользователь взаимодействует с программой через клиентскую часть, которая направляет запросы на сервер. Сервер, в свою очередь, обрабатывает эти запросы, управляет базой данных и выполняет вычисления. Подобное распределение обеспечивает гибкость, масштабируемость и удобство сопровождения.

**2.2. Основной функционал**

Программный продукт предоставляет следующий набор функций:

-Регистрация и авторизация пользователей.

-Хеширование текстовых данных с использованием криптографического алгоритма SHA-384.

-Решение уравнений численным методом хорд.

Также система должна обеспечивать:

Сохранение истории действий пользователя (для внутренних целей и отладки).

Интуитивно понятный интерфейс, подходящий для пользователей с минимальной подготовкой.

**2.3. Типы пользователей и их возможности**

В рамках приложения предусмотрены три категории пользователей:

Гость– неавторизованный пользователь, ограниченный в действиях. Может пройти регистрацию или войти в систему.

Обычный пользователь– после авторизации получает доступ к основным функциям, таким как шифрование, вычисления и работа с мультимедиа.

Администратор– обладает дополнительными правами: управление пользовательскими учетными записями, просмотр статистики и техническое обслуживание базы данных.

**2.4. Технические ограничения**

Разработка осуществляется на языке C++ с использованием фреймворка Qt.

Для хранения информации применяется СУБД SQLite.

Поддерживаются операционные системы Windows, Linux.

Все пользовательские пароли должны храниться в зашифрованном виде.

Поддержка Unicode для корректной работы с текстами на разных языках.

**2.5. Зависимости и предположения**

-Предполагается, что конечный пользователь обладает базовыми навыками работы с ПК.

-Для корректной работы требуется наличие установленных библиотек Qt и SQLite.

-Функции, связанные с работой с музыкальными файлами, требуют поддержки соответствующих аудиоформатов.

-Подключение к интернету необходимо для работы клиент-серверной архитектуры (в режиме онлайн).

**3. Требования к интерфейсам**

**3.1. Интерфейс взаимодействия с пользователем**

Приложение предоставляет несколько основных окон, каждое из которых предназначено для выполнения определённых задач.

-Окно входа в систему включает в себя поля ввода логина и пароля, а также две кнопки: для авторизации и перехода к регистрации нового пользователя.

-Окно регистрации позволяет ввести необходимые данные (имя пользователя и пароль) и содержит кнопку подтверждения создания аккаунта.

-Главный интерфейс приложения представляет собой рабочее пространство с вкладками, каждая из которых отвечает за конкретную функцию: например, хеширование с помощью SHA-384 или численное решение уравнений методом хорд. Пользователь вводит данные в соответствующие поля, запускает операции с помощью кнопок и видит результат в специальной зоне вывода.

Дизайн интерфейса ориентирован на простоту и понятность, предполагая, что пользователь сможет интуитивно разобраться в логике навигации.

**3.2. Аппаратные интерфейсы**

Программа не предъявляет особых требований к аппаратной части устройства. Для корректной работы достаточно стандартных периферийных средств- клавиатуры, мыши и экрана. Дополнительное оборудование не требуется.

**3.3. Программные интерфейсы**

Программное обеспечение взаимодействует с рядом внешних компонентов и системных ресурсов. В частности:

-СУБД SQLite используется для хранения и обработки данных пользователей. Операции включают создание и изменение таблиц, выполнение SQL-запросов, добавление и удаление записей.

-Файловая система используется для чтения и сохранения различных файлов, включая изображения и аудиофайлы.

-Сетевое взаимодействие обеспечивается с использованием протокола TCP/IP, что позволяет клиентской и серверной частям приложения обмениваться информацией в реальном времени.

**3.4. Коммуникационные интерфейсы**

Передача данных между компонентами системы осуществляется по стандартным сетевым протоколам. В основе архитектуры лежит TCP/IP-соединение, обеспечивающее стабильную и безопасную коммуникацию между клиентом и сервером независимо от операционной системы.

**4. Функциональные требования**

**4.1. Система регистрации и входа**

Пользователи, не обладающие учетной записью, могут зарегистрироваться в системе, указав уникальный логин и пароль. Программа обязана проверять, чтобы выбранное имя пользователя ещё не существовало в базе данных. При этом важно, чтобы информация о пароле сохранялась в зашифрованном виде с соблюдением современных стандартов безопасности.

После регистрации пользователь получает возможность входа в систему. При авторизации осуществляется проверка корректности введённых данных. Если логин и пароль соответствуют сохранённым в базе, пользователь получает доступ к функционалу приложения, согласно своей роли (обычный пользователь или администратор).

**4.2. Работа с хеш-функцией SHA-384**

Система предоставляет пользователям возможность хешировать текстовые данные с использованием алгоритма SHA-384.

Функции в этом разделе включают:

-Ввод произвольного текста через интерфейс приложения.

-Генерация хеш-суммы по введённому тексту с использованием алгоритма SHA-384.

-Вывод результата пользователю в удобочитаемой форме.

Кроме того, реализована возможность проверки хеша. Пользователь вводит текст и предполагаемую хеш-сумму, после чего система сравнивает вычисленный хеш с введённым и сообщает, совпадают ли они. Это может использоваться, например, для проверки целостности данных.

**4.3. Численное решение уравнений методом хорд**

В рамках математического модуля пользователь может вводить произвольное нелинейное уравнение, а также необходимые параметры: начальные приближения и требуемую точность результата. Интерфейс предусматривает понятный формат ввода, а система- автоматическую обработку введённых данных.

При запуске решения метод хорд применяется к заданной функции. Если удаётся найти корень, система отобразит результат. В случае, если метод не сходится, приложение должно отреагировать соответствующим сообщением, не вызывая сбоев в работе.

**5. Нефункциональные требования**

**5.1. Производительность**

Программа должна обеспечивать плавный отклик на действия пользователя- в идеале, не превышающий двух секунд при стандартной нагрузке. Система не ориентирована на масштабное многопользовательское использование, однако должна корректно работать в условиях параллельного доступа нескольких клиентов. Обрабатываемые объёмы данных предполагаются относительно небольшими, что упрощает требования к ресурсам. Следует учитывать, что общая производительность приложения зависит от характеристик компьютера пользователя и стабильности сетевого соединения, особенно в контексте клиент-серверного взаимодействия.

**5.2. Безопасность**

В целях базовой защиты данные пользователей сохраняются в базе данных в виде хешей. Во время входа система сверяет введённые логин и пароль с соответствующими хешированными значениями. Хотя приложение не реализует продвинутые меры защиты, такие как предотвращение SQL-инъекций или защита от сетевых атак, это объясняется его ориентацией на использование в учебных и локальных средах. При необходимости расширения или внедрения в более открытые среды, архитектура может быть доработана для включения соответствующих механизмов безопасности.

**5.3. Надежность**

В нормальных условиях работы система обеспечивает сохранение пользовательской информации. Однако при неожиданном завершении работы могут возникнуть потери данных, поскольку отказоустойчивость на уровне транзакций и резервного копирования не реализована. Тем не менее, приложение должно быть устойчиво к большинству типичных ошибок- сбои не должны приводить к его аварийному завершению. Функции логирования действий и ошибок пока не предусмотрены, но их добавление возможно в рамках будущих улучшений.

**5.4. Удобство использования**

Интерфейс приложения спроектирован таким образом, чтобы быть интуитивно понятным для пользователей с базовыми навыками владения ПК. Управление осуществляется с помощью привычных устройств ввода- клавиатуры и мыши, без необходимости дополнительного обучения или изучения инструкций.

**5.5. Сопровождаемость**

Структура проекта предельно проста, что позволяет вносить изменения в отдельные модули без полной переработки системы. Документация включает краткое описание ключевых компонентов и их функционала. Обновление приложения предполагается вручную, путём замены файлов, при этом сохранность данных при апдейтах не гарантируется. При наличии доступа к исходному коду возможна модернизация и расширение функциональности в соответствии с новыми требованиями.

**5.6. Переносимость**

Система разрабатывается с учётом кроссплатформенности и должна стабильно функционировать в популярных операционных системах, таких как Windows и Linux. Также предполагается возможность переноса данных между разными экземплярами программы, включая перенос между системами на различных ОС. Это обеспечивает гибкость и расширяет потенциальную область применения приложения.

**6. Прочие требования**

**6.1. Документационное сопровождение**

Для обеспечения доступности и понятности работы с программой, проект должен сопровождаться двумя основными видами документации. Во-первых, необходимо наличие руководства пользователя, которое поможет конечным пользователям освоить функционал приложения. Во-вторых, требуется техническая документация, описывающая архитектуру, основные модули и принципы работы системы, что особенно важно для сопровождения и возможной доработки проекта в будущем.

**6.2. Тестирование**

Перед выпуском система должна пройти серию проверок, охватывающих как функциональную, так и нефункциональную часть требований. Это включает в себя: тестирование основных функций на корректность работы; проверку соответствия нефункциональным характеристикам, таким как производительность, устойчивость и удобство использования; анализ безопасности, включая проверку на возможные уязвимости. Этап тестирования должен быть документирован и, при необходимости, повторяем для новых версий системы.

**6.3. Развёртывание**

Разработка должна предусматривать удобную процедуру установки и настройки. Для этого система должна сопровождаться подробной инструкцией по развертыванию. Помимо этого, желательно реализовать возможность автоматизированной установки в различных окружениях. Это обеспечит гибкость в применении и ускорит процессы внедрения.

**7. Приложения**

**Глоссарий**

**БД** - база данных.

**ПО** - программное обеспечение.

**СУБД** - система управления базами данных.

**GUI** (Graphical User Interface) - графический пользовательский интерфейс.

**TCP/IP** - набор протоколов передачи данных, используемых в сетях, включая Интернет.

**SQL** - язык структурированных запросов, используемый для работы с реляционными базами данных.

**Метод хорд** - численный метод решения нелинейных уравнений, основанный на аппроксимации функции хордой.

**SHA-384** (Secure Hash Algorithm 384) - криптографическая хеш-функция, которая генерирует 384-битное (48-байтное) значение хеша.