**Техническое задание для проекта "Крипто кошелёк" с использованием PyQt5 и MySQL**

1. Введение

Проект "Крипто кошелёк" разрабатывается на языке Python с использованием библиотеки PyQt5 для создания графического интерфейса и MySQL в качестве базы данных. Это техническое задание описывает требования к архитектуре, функционалу приложения и структуре файлов.

2. Техническая архитектура

2.1. Язык программирования: Python 3.

2.2. Графическая библиотека: PyQt5 для создания графического интерфейса.

2.3. Система управления базами данных: MySQL.

2.4 Шифрование данных: Использование библиотеки для шифрования данных, такой как cryptography.

2.5 Контроль версий: Git

3. Функционал приложения

3.1. Регистрация и аутентификация

Регистрация новых пользователей с использованием электронной почты и пароля.

Аутентификация пользователя по логину и паролю.

3.2. Управление крипто валютными активами

Создание и хранение кошельков для различных криптовалют.

Возможность отправки и получения криптовалют между кошельками.

Отображение баланса в каждой поддерживаемой криптовалюте.

3.3. Безопасное хранение и шифрование

Шифрование приватных ключей и данных пользователя.

Защита приватных ключей и паролей.

3.4. Отслеживание портфеля

Отображение стоимости портфеля пользователя в реальном времени.

История транзакций с возможностью фильтрации и поиска.

3.5. Обмен криптовалют

Возможность обмена одной криптовалюты на другую внутри приложения.

3.7. Много валютная поддержка

Поддержка различных криптовалют и токенов.

3.8. Удобство и интуитивный интерфейс

Разработка графического интерфейса с использованием PyQt5 для обеспечения удобства использования.

5. Требования к безопасности

5.1. Шифрование данных

Использование библиотек для шифрования данных, таких как cryptography, для хранения приватных ключей и чувствительных данных пользователя.

5.2. Защита от несанкционированного доступа

Реализация мер безопасности для защиты от несанкционированного доступа к кошелькам и данным пользователя.

5.3. Защита от атак

Защита от атак перебора паролей и SQL-инъекций.

6. Тестирование

6.1. Тестирование

Разработка модульных тестов для проверки функционала.

Интеграционное тестирование для проверки взаимодействия между компонентами приложения.

7. Дополнительные требования

7.1. Документация

Написание документации, включая описание архитектуры, API и инструкции для пользователя.