**Спецификация требований к клиент-серверному приложению**

Приложение представляет собой клиент-серверную систему для решения задач по математической логике и теории алгоритмов (МЛТА). Система обеспечивает автоматизированную генерацию задач, их решение и проверку с разделением ролей (студент/преподаватель).

* Графический интерфейс (GUI) на Qt для клиентской части.
* Авторизация и регистрация пользователей с ролями: гость, пользователь, администратор.
* Шифрование данных (алгоритмы md5 и tripple des).
* проверка того, что ответ пользователя есть цикл заданного графа
* Управление базой данных SQLite и статистикой для администратора.

**Серверная часть:**- Обработка одновременных подключений от нескольких клиентов- Работа с SQLite базой данных (хранение пользователей, задач, статистики)- Реализация синглтона для доступа к БД

(`Database`)- Генерация 4 типов задач:

1. Система представителей двудольного графа

2. Проверка двудольности графа

3. Поиск гамильтонова цикла

4. Алгоритм Дейкстры (кратчайшие пути)- Docker-контейнеризация

**Клиентская часть:**- Оконный интерфейс (Qt) с экранами:

- Авторизация (`AuthWindow`)

- Главное окно студента (`MainWindow`)

- Окно решения задачи (`TaskForm`)

- Окно преподавателя (`TeacherWindow`)- Синглтон для управления соединением с сервером (`Network`)

**Защита передаваемых данных**

Все данные, передаваемые между клиентом и сервером, шифруются с использованием шифра tripple des.

**Защита хранимых паролей**

Система использует алгоритм md5

Тестирование

Должен быть реализован UnitTest для каждой из следующих функций: md5, шифр tripple des, метод секущих, проверка того, что ответ пользователя есть цикл заданного графа. Необходимо убедиться в успешном тестировании. При обнаружении сбоев/ошибок, провести редактирование функционала. А также интеграционные тесты, включающие взаимодействие клиент-сервер и обработку ошибок (неверный логин, разрыв соединения).

Функциональные требования

1. Общие требования

* Сервер:
  + Обработка одновременных подключений от нескольких клиентов.
  + Работа с БД (хранение логинов, паролей, email, даты регистрации).
  + Реализация синглтона для доступа к БД.
  + Docker-контейнеризация.
* Клиент:
  + Оконный интерфейс (Qt) с экранами:
    - Авторизация (LoginWindow).
    - Регистрация (RegisterWindow).
    - Главное меню (MainWindow).
    - Таблицы данных (TaskWindow).
  + Синглтон для управления соединением с сервером.

1. Безопасность

* Регистрация/авторизация:
  + Валидация логина/пароля.
  + Хранение паролей в БД в виде хешей (md5).
* Шифрование:
  + Реализация шифра tripple des для передачи данных.

1. Математические функции

* Метод секущих

1. Администрирование

* Роли:
  + Администратор: доступ к таблице пользователей, блокировка аккаунтов.
  + Пользователь: ограниченный функционал.
* Функции администратора:
  + Просмотр всех пользователей в таблице (id, login, password, email, created\_at).
  + Сброс паролей.

1. Отображение данных

Таблицы:

* Автоматическое обновление при изменении данных.
* Колонки: ID, Логин, Пароль, Email, Дата регистрации.
* Метки: Вывод результатов вычислений (корни уравнений, декодированные сообщения).

2. Безопасность- Регистрация/авторизация:

- Валидация логина/пароля

- Хранение токенов сессии вместо паролей

- Использование MD5-хеширования при создании токенов- Защита передаваемых данных:

- JSON-формат сообщений

- Аутентификация по токену

3.Документация- Doxygen: описание классов и методов- Тест-план: Excel-файл с тест-кейсами- Диаграммы:

- UseCase

- Классов (сервер/клиент)

- Последовательности

- Базы данных

Требования к базе данных**Структура БД:**

В базе данных должны хранится:

* Логины и пароли пользователей (в зашифрованном виде).
* Email и дата регистрации.
* Роли (администратор, обычный пользователь).

Сценарии использования

Регистрация

1. Пользователь вводит логин, пароль, email.
2. Сервер проверяет уникальность логина.
3. Пароль хешируется (md5) и сохраняется в БД.