

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ

на клиент-серверное приложение для решения СЛАУ первого порядка

методом Гаусса

по дисциплине

«Технологии и методы программирования.»

Выполнила:

ст. гр. 241-3211

Минеева Анастасия

Москва – 2025

Оглавление

1. Общие сведения.....	3
1.1 Клиент.	3
1.2 Сервер.	5
2. Назначение	6
3. Функциональные требования.....	6
4. Нефункциональные требования.....	7
5. Требования к интерфейсу пользователя.....	7
6. Валидация и сообщения об ошибках	7
7. Технические требования.....	9
8. Безопасность и защита данных	9
9. Возможности расширения	9
10. Описание предметной области.....	10
11. Архитектура системы	10
11.1 Клиентская часть (Qt GUI + C++ на Qt).....	10
11.2 Серверная часть (C++ на Qt).....	11
12. Диаграммы.....	13
12.1. Диаграмма классов.	13
12.2 Диаграмма Use Case	14
13. Тестирование	15
14. Скриншоты интерфейса.....	15
15. Заключение	21
16. Список использованных источников	22

1. Общие сведения

Данный документ описывает функциональные и нефункциональные требования к информационной системе — клиент-серверному приложению, предназначенному для решения систем линейных алгебраических уравнений первого порядка (СЛАУ) методом Гаусса.

Приложение реализовано на языке C++ с использованием фреймворка Qt. Оно построено по архитектуре клиент-сервер, где клиентская часть предоставляет удобный графический интерфейс для взаимодействия пользователя с системой, а серверная часть отвечает за обработку данных, парсинг уравнений и вычисление решения методом Гаусса.

1.1 Клиент.

Клиент реализует следующие функции:

- Взаимодействие с сервером построено с использованием паттерна Singleton, что позволяет централизованно управлять сетевыми запросами и гарантирует единый экземпляр сетевого клиента на всё приложение.
- Проверка корректности заполнения полей ввода осуществляется до отправки запроса на сервер. Пользователь получает понятные уведомления на русском языке о некорректно введенных данных.
- Вся информация, введенная пользователем (система линейных уравнений, логин, пароль, email и прочее), собирается, формируется в структуру запроса и передается на сервер через JSON-объекты.

- В случае успешного ответа от сервера, клиент отображает результат решения в специально отведённой области.
- При возникновении ошибок — как на стороне клиента, так и на стороне сервера — пользователь немедленно получает наглядное уведомление с объяснением проблемы (например, "Неверный логин или пароль", "Ошибка в уравнении", "Не удалось соединиться с сервером").
- Создание окна регистрации и авторизации. Функция отвечает за инициализацию и отображение интерфейса, включающего поля ввода логина, пароля, кнопки для входа, регистрации и восстановления пароля. В этом окне реализована логика переключения между режимами авторизации и регистрации, а также базовая валидация вводимых данных.
- Создание окна восстановления пароля. В этом окне пользователь вводит адрес электронной почты для получения кода подтверждения и восстановления доступа к аккаунту. Функция инициализирует поля ввода, кнопки и сообщения об ошибках, а также обрабатывает взаимодействие с сервером для отправки запроса на сброс пароля.
- Создание главного окна приложения. После успешной авторизации создаётся главное рабочее окно, содержащее приветственное сообщение, инструкции и элементы навигации по приложению, включая переход к решению системы уравнений.
- Создание окна решения системы уравнений. Окно предоставляет интерфейс для ввода коэффициентов системы, запуска решения и отображения результата. Функция отвечает за сбор данных, валидацию, отправку запроса на сервер и вывод решения в текстовом поле.
- Создание окна подтверждения почты при регистрации. В данном окне пользователь вводит код, отправленный на электронную почту, для подтверждения регистрации. Функция обеспечивает ввод кода, проверку его корректности и отправку подтверждающего запроса на сервер.

1.2 Сервер.

Серверная часть реализована на языке C++ и выполняет функции обработки, валидации и решения систем линейных уравнений. Сервер принимает запросы от клиента в формате JSON, обрабатывает их и возвращает результат или сообщение об ошибке в структурированном виде. В архитектуре сервера чётко разделены зоны ответственности: парсинг уравнений, вычисления и формирование ответов. Также сервер взаимодействует с базой данных, которая хранит в себе данные о всех пользователях.

Основные функции сервера включают:

- Приём и десериализация входящих JSON-запросов от клиента, содержащих данные, введённые пользователем (уравнения, логин, пароль и др.).
- Анализ и парсинг математических выражений с использованием регулярных выражений. Сервер извлекает коэффициенты переменных и свободные члены, формируя матрицу коэффициентов и вектор правой части.
- При обнаружении синтаксических ошибок в уравнениях (например, отсутствие знака «=», неизвестная переменная, некорректный числовой коэффициент) сервер формирует детализированное сообщение об ошибке, которое отправляется клиенту.
- Решение системы уравнений методом Гаусса реализовано на сервере. Алгоритм учитывает случаи несовместности, бесконечного числа решений и выдает точный результат или соответствующее уведомление.

- Сервер обрабатывает также запросы на регистрацию, авторизацию и восстановление пароля, работая с локальной базой данных или хранилищем, проверяя корректность логина, пароля и email.
- Все сообщения, отправляемые клиенту, оформлены в виде информативных JSON-ответов, что позволяет клиенту адекватно интерпретировать результат.

2. Назначение

Система предназначена для студентов и преподавателей технических вузов, а также других пользователей, нуждающихся в автоматизированном решении СЛАУ. Программа упрощает процесс ввода, проверки и обработки уравнений, сокращая время и снижая вероятность ошибок.

3. Функциональные требования

- Регистрация и авторизация пользователей.
- Восстановление пароля с отправкой кода подтверждения на e-mail.
- Основное окно с инструкцией по вводу системы уравнений.
- Ввод от 2 до 5 уравнений в формате: $2x+3y=5$
- Автоматическая отправка системы уравнений на сервер.
- Проверка корректности данных на сервере и клиенте.
- Решение системы уравнений методом Гаусса.
- Вывод пошагового решения или результата.
- Сообщения об ошибках с понятными пояснениями на русском языке.

4. Нефункциональные требования

- Приложение работает на Windows и macOS.
- Язык интерфейса — русский.
- Время отклика сервера — не более 2 секунд.
- Уровень ошибок — не более 1% некорректных обработок корректного ввода.
- Обработка всех ошибок с подробным объяснением.

5. Требования к интерфейсу пользователя

- Простота и понятность.
- Подсказки по заполнению полей и функционалу.
- Инструкция в основном окне.
- Сообщения об ошибках в виде диалоговых окон.
- Окно с результатом форматировано как текстовая область (TextBrouser), доступная только для чтения.

6. Валидация и сообщения об ошибках

При нажатии на кнопку "Решить уравнение" выполняются следующие проверки:

- **Пустое поле уравнения**

Сообщение: «Ошибка: все строки должны быть заполнены.»

Проверка в «EquationParser: parseSystem ()».

- **Отсутствует знак "="**

Сообщение: «Ошибка: уравнение должно содержать ровно один знак равенства.»

Проверка в «parseEquation ()».

- **Недопустимые символы (русские буквы, спецсимволы)**

Сообщение: «Ошибка: используйте только латинские буквы (a-z, A-Z), цифры и символы =, +, -, .»

Проверка в «containsInvalidCharacters ()».

- **Несовпадение переменных в уравнениях**

Сообщение: «Ошибка: во всех уравнениях должны использоваться одинаковые переменные (например, x, y, z).»

Проверка в «parseSystem ()».

- **Некорректный коэффициент (не число)**

Сообщение: «Ошибка: неверный формат коэффициента.

Пример: $2x + 3y = 5$ »

Проверка в «parseEquation ()».

- **Пустая правая часть (=)**

Сообщение: «Ошибка: правая часть уравнения не может быть пустой.»

Проверка в «parseEquation ()».

- **Минимальное количество уравнений (2)**

Сообщение: «Нельзя удалить поле. Минимум должно быть 2 уравнения.»

Проверка в «MainWindow: on_pushButtonDelUrav_2_clicked ()».

- **«Ненадёжный пароль (менее 5 символов)»**

Сообщение: «Пароль должен содержать минимум 5 символов.»

Проверка в «AuthRegForm: on_pushButtonReg_clicked ()».

Все ошибки отображаются пользователю в модальных окнах «QMessageBox: critical».

7. Технические требования

- Qt 6.8.9 (MSVC 2022, x86_64) или выше.
- Компилятор C++17.
- Серверная часть — консольное приложение Qt или C++ с JSON API.
- Протокол взаимодействия: JSON через TCP.

8. Безопасность и защита данных

- Пароли хранятся в виде хешей.
- Проверка кода подтверждения при восстановлении пароля.
- Подтверждение почты при регистрации (аналогично с отправкой кода подтверждения)
- Защита от SQL-инъекций (при использовании базы данных).

9. Возможности расширения

- Поддержка других методов решения СЛАУ.
- История решений пользователя.
- Поддержка систем более 5 уравнений.
- Интеграция с образовательными платформами.
- Решение системы линейных уравнений второго порядка

10. Описание предметной области

Решение систем линейных уравнений первого порядка — базовая задача курса линейной алгебры и аналитической геометрии. Проект автоматизирует решение, обучая пользователей правильному вводу и логике метода Гаусса.

11. Архитектура системы

11.1 Клиентская часть (Qt GUI + C++ на Qt))

Назначение: взаимодействие с пользователем (ввод уравнений, регистрация, подтверждение, восстановление пароля и т.д.)

Основные компоненты:

- MainWindow, AuthRegForm, EmailConfirmationForm, PasswordRecoveryForm и др. — формы интерфейса.
- NetworkClient — отвечает за отправку/получение данных от сервера по TCP.
- **Функции:**
 - Отображение GUI.
 - Сбор данных (уравнений, регистрационных форм).
 - Формирование JSON-запросов и отправка на сервер.
 - Приём ответа и его отображение пользователю (например, результат решения СЛАУ методом Гаусса).
 - Проверка на корректное заполнение полей.
 - Уведомление пользователя об ошибках.
 - Создание удобного функционала и реализация функционала кнопок.

11.2 Серверная часть (C++ на Qt)

Назначение: приём JSON-запросов от клиента, парсинг уравнений, решение СЛАУ, регистрация пользователей, отправка писем и возвращение результатов.

1. Сетевой уровень:

- **Класс:** MyTcpServer
- **Функции:**
 - Инициализация TCP-сервера.
 - Принятие входящих соединений.
 - Получение JSON-запросов от клиента.
 - Передача данных в RequestHandler для обработки.

2. Обработка запросов:

- **Класс:** RequestHandler
- **Функции:**
 - Распознавание типа запроса (например, "solve_equation", "register_user" и т.п.).
 - Десериализация JSON.
 - Вызов соответствующих модулей: парсер, решатель, БД и др.
 - Формирование ответа и отправка его клиенту.

3. Парсинг уравнений:

- **Класс:** EquationParser
- **Функции:**
 - Преобразование текстовых уравнений в матричное представление ($Ax = b$).
 - Работа с JSON-входом.

- Подготовка данных для решателя (матрицы коэффициентов и свободных членов).

4. Решение СЛАУ:

- **Класс:** GaussianElimination
- **Функции:**
 - Реализация метода Гаусса для решения СЛАУ.
 - Обработка некорректных данных (нулевые строки, несовместные системы).
 - Возврат решения в виде массива чисел (или JSON).

5. Вспомогательные модули:

- DatabaseManager — управление пользователями, сохранение данных в SQLite/MySQL.
- SMTPClient — отправка email (например, при подтверждении регистрации или восстановлении пароля).
- ConfigLoader — загрузка параметров сервера (порты, IP, конфигурации)

12. Диаграммы

12.1. Диаграмма классов.

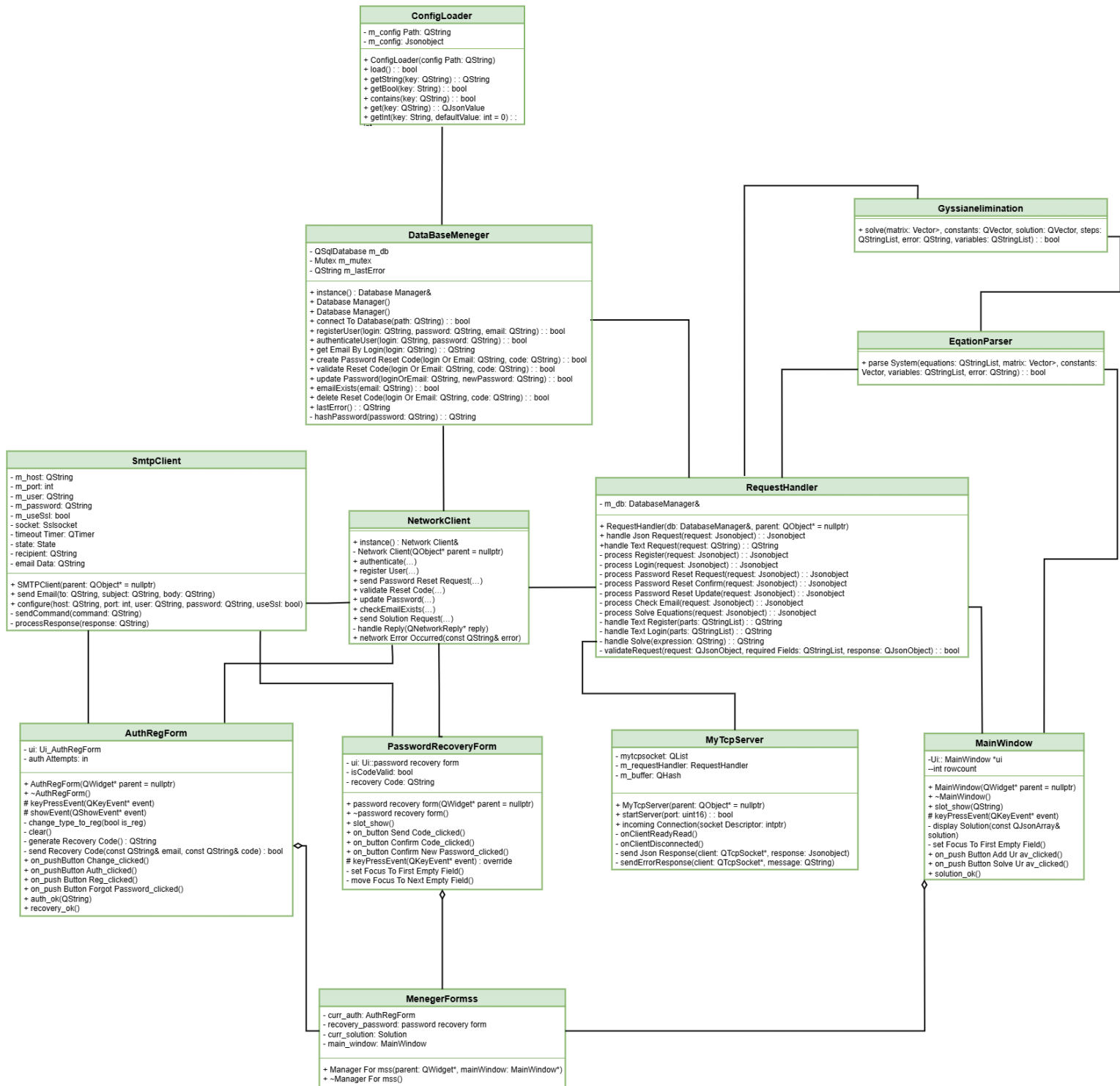


Рисунок 1 — диаграмма классов.

12.2 Диаграмма Use Case

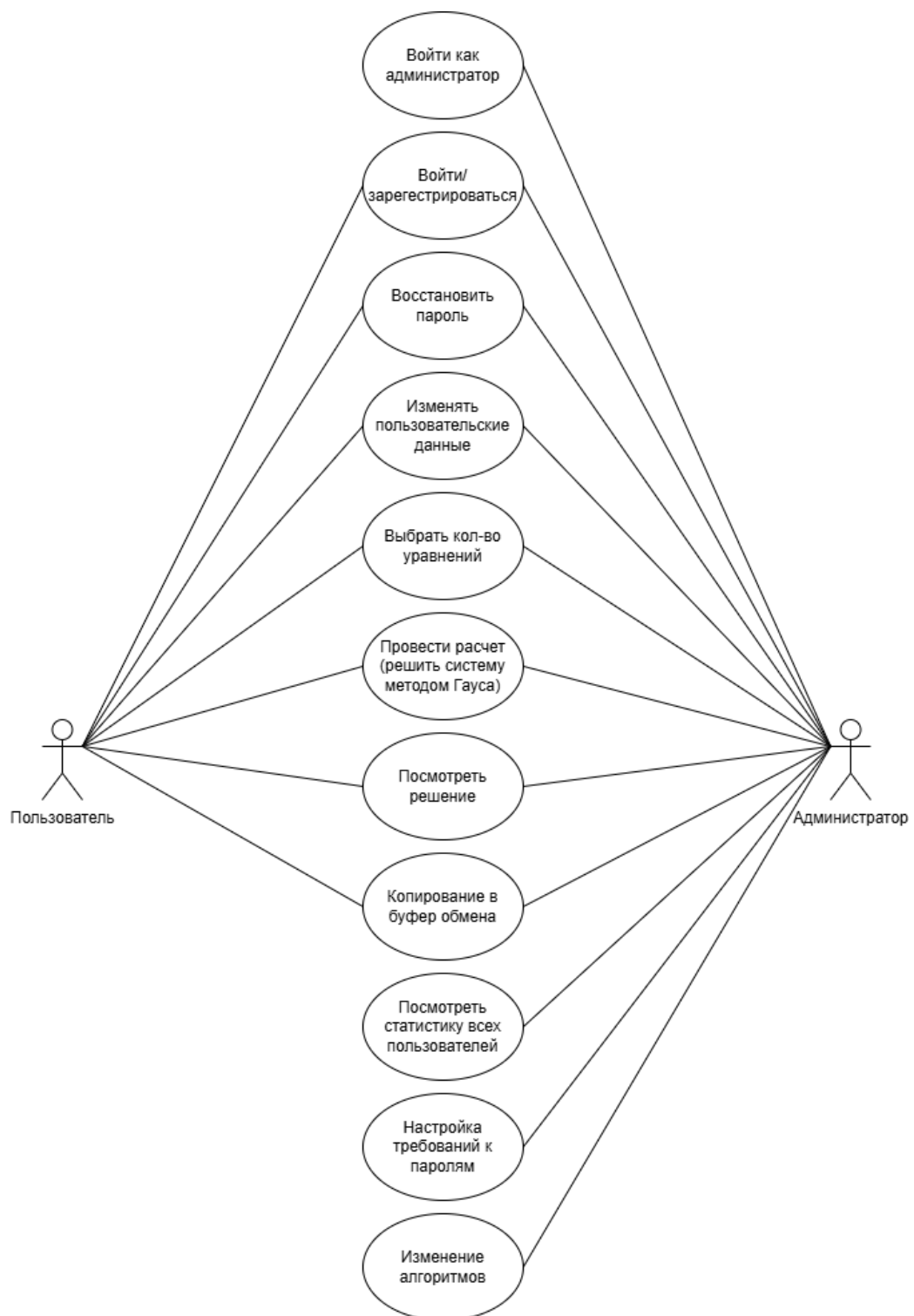


Рисунок 2 — Use Case диаграмма.

13. Тестирование

Проводилось ручное тестирование всех модулей:

- Валидация ввода.
- Обработка ошибок.
- Корректность метода Гаусса (сравнение с Wolfram Alpha).
- Отказоустойчивость при пустом вводе, неправильном формате.
- Unit test в qt

14. Скриншоты интерфейса

- **Окно авторизации**

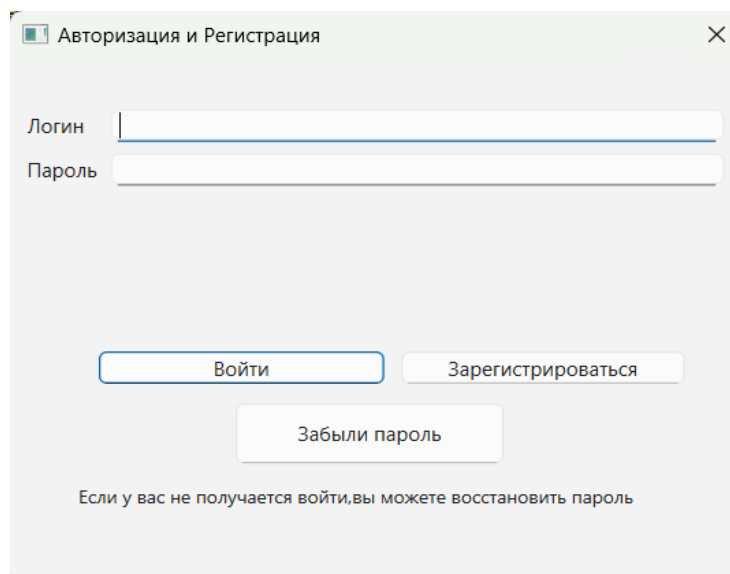


Рисунок 3 — окно авторизации.

1. Поля для ввода логина и пароля. (не допускаются пустые поля, ограничение на размер пароля – максимум 10 символов, минимум 5 символов, курсор автоматически помещается в верхнее незаполненное поле при открытии окна, при нажатии «Enter» - переход к следующему полю.)

2. Кнопки «Войти», «Зарегистрироваться», «Забыли пароль».

- При нажатии кнопки «Войти» - проверяется база данных на существования такого пользователя и сравнение паролей, если все верно, то открывается главное окно программы, в обратном случае пользователя уведомляют об ошибке авторизации)
- При нажатии кнопки «Зарегистрироваться» - пользователь видит окно регистрации.
- При нажатии кнопки «Забыли пароль» - открывается окно восстановления пароля.

3. Подсказка возле кнопки восстановления пароля, для того чтобы сориентировать пользователя.

- **Окно регистрации**

Рисунок 4 — окно регистрации.

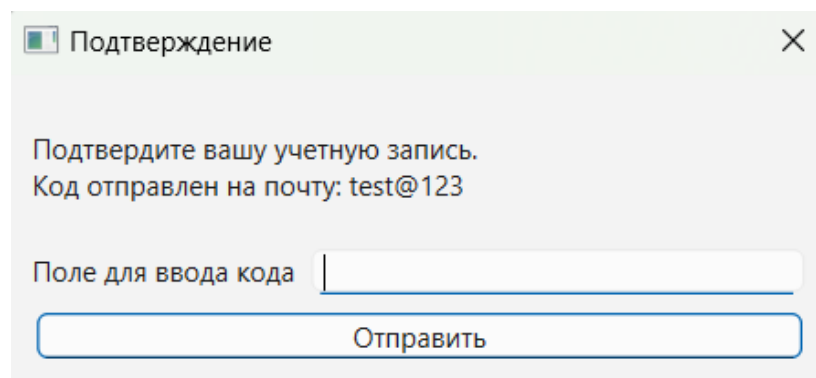
1. Поля для ввода логина, пароля, повтора пароля и почты. (не допускаются пустые поля, ограничение на размер пароля – максимум 10 символов, минимум 5 символов, курсор автоматически помещается в верхнее незаполненное поле при открытии окна, при нажатии «Enter» - переход к следующему полю.)

2. Кнопки «Регистрация», «Авторизоваться», «Забыли пароль».

- При нажатии кнопки «Регистрация» - проверяется база данных на существования такого пользователя, если пользователь с таким логином уже существует, пользователя уведомляет об этом программа, поля пароля и повтор пароля сверяются на соответствие, если все прошлые условия выполнены, то открывается окно для подтверждения почты с помощью отправленного кода.
- При нажатии кнопки «Авторизоваться» - пользователь видит окно авторизации.
- При нажатии кнопки «Забыли пароль» - открывается окно восстановления пароля.

3. Подсказка возле кнопки восстановления пароля, для того чтобы сориентировать пользователя

- **Форма подтверждения почты при регистрации (открывается при нажатии – Регистрация)**



Подтверждение

Подтвердите вашу учетную запись.
Код отправлен на почту: test@123

Поле для ввода кода

Рисунок 5 — окно подтверждения почты.

1. Поле для ввода кода (не должно быть пустым).

2. Кнопка «Отправить» - после нажатия сверяется введенный код с отправленным, при верном вводе регистрация завершается, в обратном случае сообщается пользователю, что код не верный.

- **Форма восстановления пароля (открывается при нажатии – Забыли пароль)**

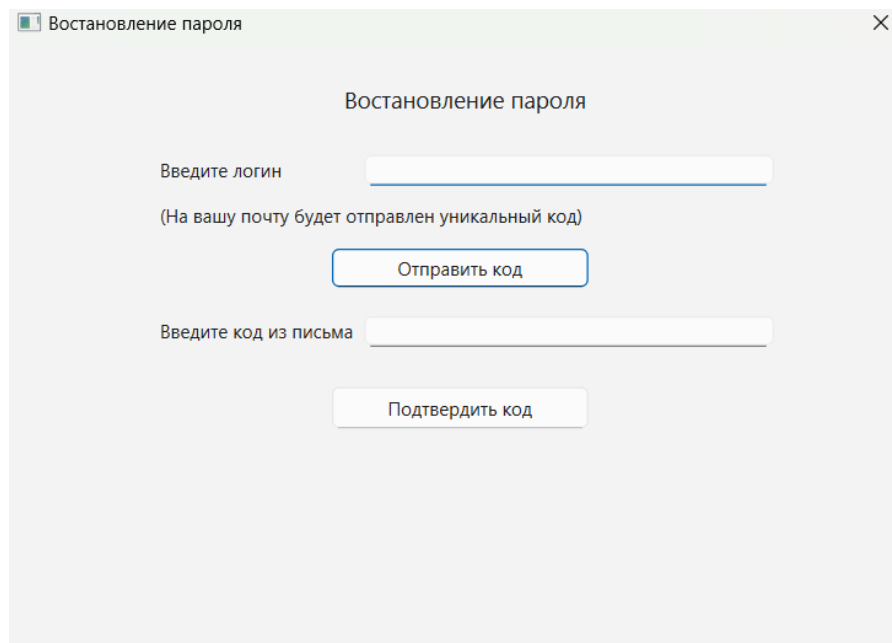


Рисунок 6 — окно восстановления пароля.

1. Поля для ввода логина, кода, пароля. (не допускаются пустые поля, ограничение на размер пароля – максимум 10 символов, минимум 5 символов, курсор автоматически помещается в верхнее незаполненное поле при открытии окна, при нажатии «Enter» - переход к следующему полю.)

2. Кнопки «Отправить код», «Подтвердить код».

- При нажатии кнопки «Отправить код» - проверяется база данных на существования такого пользователя и отправляется код подтверждения на привязанную к логину почту.

- При нажатии кнопки «Подтвердить код» - если введенный код верный, то для пользователя становятся видимыми поля для ввода нового пароля и повтора нового пароля. (а также кнопка для установления нового пароля)
- Основное окно

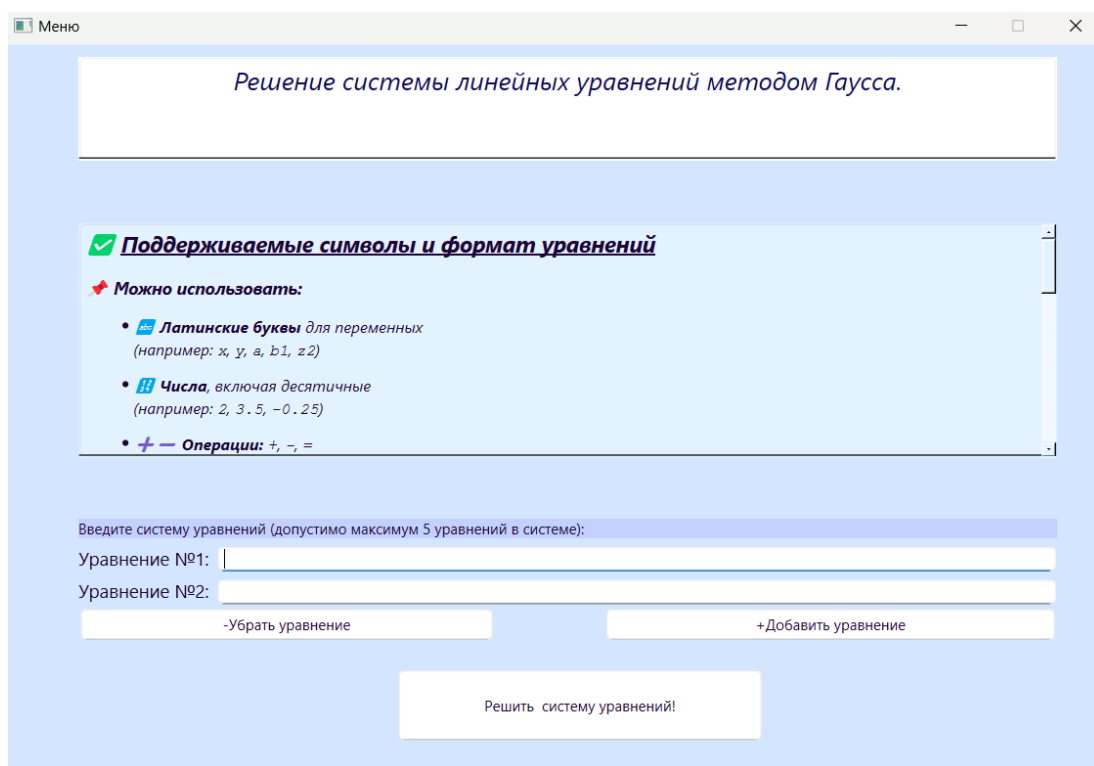


Рисунок 7— основное окно программы.

1. Поля для ввода уравнений (изначально пользователю видны только два поля из пяти возможных, не допускаются пустые поля, курсор автоматически помещается в верхнее незаполненное поле при открытии окна, при нажатии «Enter» - переход к следующему полю, допускается ввод символов указанных в памятке над полями ввода уравнений)

2. Кнопки «+Добавить уравнение», «-Убрать уравнение», «Решить систему уравнений».

- При нажатии кнопки «+Добавить уравнение» - добавляется дополнительное поле для ввода уравнения (но при условии, что все имеющиеся на данный момент поля заполненные)
- При нажатии кнопки «-Убрать уравнение» - удаляется дополнительное поле для ввода уравнения (но при условии, что открытых полей больше, чем два и дополнительное поле - пустое.)
- При нажатии кнопки «Решить систему уравнений» - открывается окно с решением и ответом.

3. Подсказка возле полей ввода уравнений, для того чтобы сориентировать пользователя. (так же в главном окне приложения есть памятка с подробными правилами заполнения полей ввода уравнений)

• Окно результата

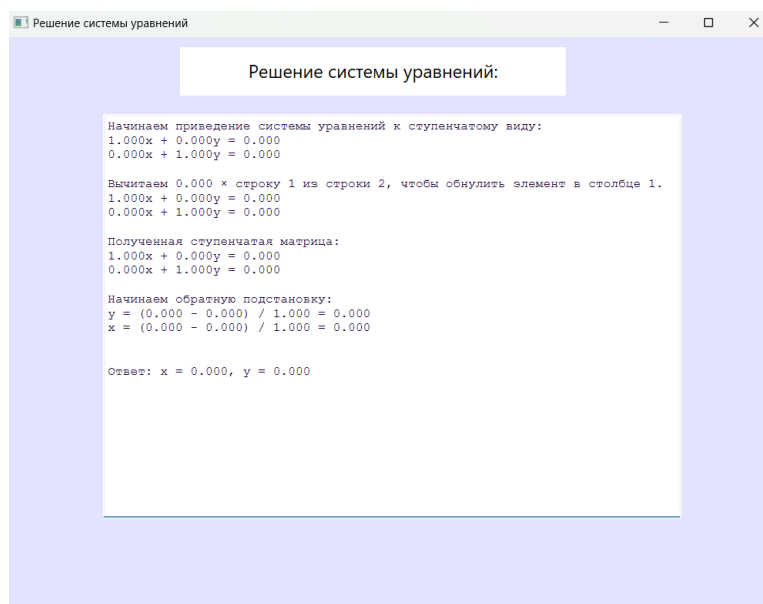


Рисунок 8 — окно результата.

1. Вывод подробного решения
2. Конечный ответ.

15. Заключение

В ходе выполнения проекта было разработано полнофункциональное клиент-серверное приложение для решения систем линейных уравнений методом Гаусса. Программа реализует следующие ключевые возможности:

1. Гибкий пользовательский интерфейс:

- Поддержка от 2 до 5 уравнений с динамическим добавлением полей
- Интуитивное управление с помощью клавиатуры (переход между полями по Enter)
- Пошаговое отображение решения с подробными пояснениями

2. Надежная система аутентификации:

- Полный цикл регистрации с подтверждением email
- Восстановление пароля через одноразовые коды
- Валидация данных на стороне клиента и сервера

3. Стабильное сетевое взаимодействие:

- Четкая JSON-спецификация запросов и ответов
- Обработка ошибок соединения и таймаутов
- Логирование сетевых операций для отладки

4. Математический модуль:

- Точный парсинг уравнений с обработкой различных форматов
- Реализация метода Гаусса с определителем матрицы
- Поддержка особых случаев (нет решений/бесконечное множество решений)

Приложение прошло комплексное тестирование, включающее:

- Проверку корректности решения систем уравнений
- Тестирование граничных случаев (вырожденные матрицы)
- Валидацию входных данных
- Проверку устойчивости к некорректному вводу

Перспективы развития:

- Добавление графического представления решений
- Поддержка нелинейных уравнений
- Экспорт результатов в LaTeX-формате
- Мобильная версия приложения
- Программа демонстрирует высокую стабильность работы и может быть использована как учебное пособие по линейной алгебре, а также как практический инструмент для инженерных расчетов.

16. Список использованных источников

1. Киселёв А.Ф. Линейная алгебра.
2. Документация Qt (<https://doc.qt.io>)
3. Википедия: Метод Гаусса
4. Лекции по архитектуре вычислительных систем
5. <https://ru.stackoverflow.com/> — примеры решений