

# **Отчет №1**

## **1. Анализ ТЗ и спецификация**

На основании представленного технического задания было проведено изучение функциональных и нефункциональных требований к системе учета заявок на ремонт климатического оборудования. В результате анализа выделены основные сущности системы: заявки, пользователи, комментарии, а также определены входные и выходные данные для каждой роли. Входные данные включают информацию о заявке, данные пользователя при регистрации и аутентификации. Выходные данные формируются в виде списков заявок, статистических отчетов и системных уведомлений. Для реализации бэкенда выбран фреймворк FastAPI, который обеспечивает высокую производительность, автоматическую генерацию интерактивной документации Swagger и удобную работу с асинхронными запросами. Для разработки пользовательского интерфейса выбран фреймворк Streamlit, позволяющий быстро создавать интерактивные веб-приложения на Python с минимальными усилиями по верстке и настройке.

## **2. Алгоритм работы системы**

Разработан основной алгоритм обработки заявки, включающий этапы создания, регистрации, обработки, исполнения и закрытия. Каждый этап сопровождается изменением статуса и возможностью добавления комментариев. Для статистических расчетов реализованы алгоритмы подсчета общего количества заявок, количества выполненных заявок и среднего времени ремонта. В качестве базы данных использована SQLite, которая была представлена в готовом виде, что соответствует требованию использования существующей структуры данных. Алгоритмы доступа к данным учитывают ролевую модель, обеспечивая разграничение прав для менеджеров, операторов, специалистов и заказчиков.

## **3. Реализация интерфейса**

Интерфейс программного модуля реализован на Streamlit, который позволяет быстро создавать адаптивные веб-интерфейсы с использованием встроенных

компонентов. Интерфейс включает главную страницу с QR-кодом для оценки качества сервиса, раздел работы с заявками, страницу управления пользователями, блок комментариев и модуль статистики. Внешний вид всех компонентов системы выполнен в едином стиле с навигационной панелью, адаптированной под роль пользователя. Каждое окно имеет соответствующий заголовок, а навигация между разделами осуществляется интуитивно понятно, в том числе с возможностью возврата на предыдущую страницу.

#### **4. Структура проекта**

Проект организован в виде двух основных модулей: бэкенд на FastAPI и фронтенд на Streamlit. Бэкенд включает модули для работы с данными, схемы валидации и маршрутизацию API. Фронтенд реализован в виде единого приложения с четким разделением страниц по функциональности. Такое разделение обеспечивает поддержку принципа единственной ответственности и упрощает дальнейшую модификацию системы.

#### **5. Обработка ошибок и валидация**

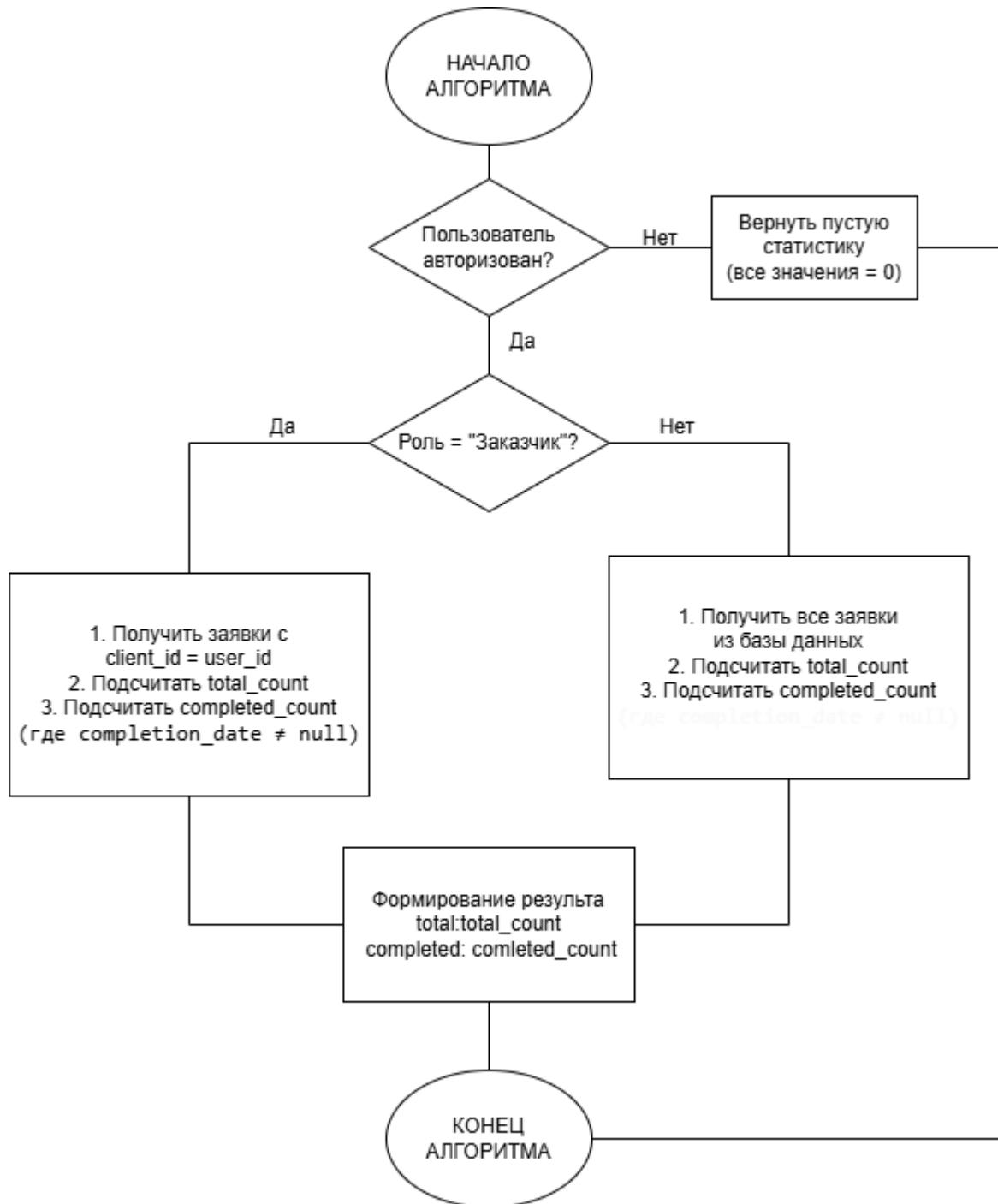
Для обеспечения корректности работы системы реализована валидация входных данных с использованием Pydantic-схем. Все передаваемые через API данные проверяются на соответствие типам и форматам. Обработка исключительных ситуаций включает уведомление пользователя о возникающих ошибках, запрос подтверждения перед удалением данных, информирование об отсутствии результатов поиска. Сообщения об ошибках содержат полное описание проблемы и рекомендации по ее устранению. Визуальные подсказки при вводе данных помогают пользователю избегать ошибок заполнения форм.

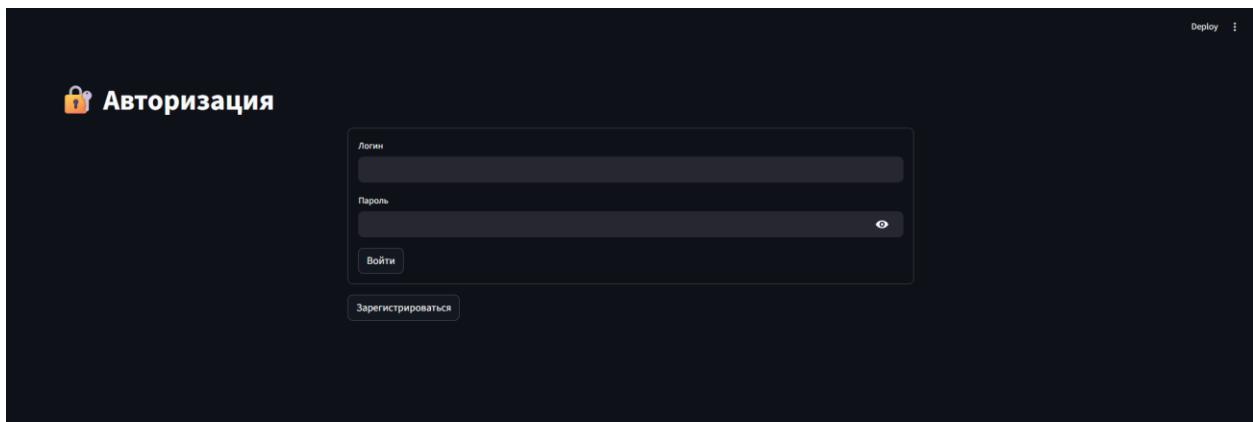
#### **6. Детализированный алгоритм расчёта статистики по заявкам**

##### **Цель алгоритма:**

Алгоритм расчета статистики предназначен для получения аналитических данных о работе системы с учетом роли текущего пользователя. Для менеджера доступна полная статистика по всем заявкам,

включая общее количество, количество выполненных заявок, среднее время выполнения и распределение по типам неисправностей. Специалист получает статистику только по своим заявкам с расчетом личной эффективности и времени выполнения работ. Заказчики и операторы имеют ограниченный доступ к статистической информации в соответствии со своими функциональными обязанностями. Алгоритм использует агрегирующие SQL-запросы к базе данных и формирует результат в удобном для визуализации формате.





## Страница авторизации

The screenshot displays the main service center dashboard. On the left, a sidebar shows the user's name (Гайдуков Вячеслав Владимирович) and role (Менеджер). Below are navigation buttons for Главная, Заявки, Пользователи, Комментарии, Статистика, and Выход. The main area features a large QR code with the text "Оцените нашу работу!" (Rate our work!) above it. A message encourages users to scan the QR code for feedback.

## Главная панель

The screenshot shows a list of repair requests. The title is "Заявки на ремонт" (Repair requests). Below it is a section titled "Текущие заявки" (Current requests) with a table. The table columns are ID, Дата (Date), Оборудование (Equipment), Модель (Model), Проблема (Problem), Статус (Status), Мастер (Master), and Клиент (Client). The table contains five entries. At the bottom, there is a "Детали заявки" (Request details) section with a dropdown menu showing ID 4, and a status summary: "Статус: Новая заявка", "Клиент ID: 8", and "Мастер ID: 2".

## Список заявок с фильтрами

The screenshot shows a dark-themed web application interface for creating a new repair request. At the top, there's a header with a document icon and the title 'Заявки на ремонт'. Below the header, a navigation bar includes links for 'Текущие заявки', 'Создать' (highlighted in red), 'Изменить', and 'Удалить'. A sub-header 'Создать новую заявку' is followed by a large form area. The form fields include:

- Дата заявки: 2025/12/19
- Статус\*: Новая заявка
- Тип оборудования\*: Например: Кондиционер, Увлажнитель
- Назначить специалиста: Не назначен
- Модель оборудования\*
- Описание проблемы\*
- ID клиента\*: 1

At the bottom right of the form is a 'Создать заявку' button, and at the bottom left is a 'На главную' button.

Форма создания с валидацией

The screenshot shows a dark-themed web application interface for viewing comments on a repair request. At the top, there's a header with a speech bubble icon and the title 'Комментарии'. Below the header, a dropdown menu lists 'Выберите заявку для просмотра комментариев' and 'ID: 1 - Кондиционер (В процессе ремонта)'. The main content area displays a comment for request ID 1:

**Комментарии к заявке ID: 1**

ID: 2  
Всё сделаем!

2025-12-19 00:37:19

Below this, there's a section titled 'Добавить комментарий' with a text input field labeled 'Текст комментария' and a 'Добавить комментарий' button. At the bottom left is a 'На главную' button.

Детали заявки с комментариями

>> Stop Deploy :

## Статистика

Общая статистика Статистика по пользователям

### Общая статистика системы

Выполнено заявок	Среднее время (дней)	Всего заявок
1	178.0	5

### Распределение по типам проблем

Данные для графика:

Тип проблемы	Количество
0 Увлажнитель воздуха продолжает работать при предельном снижении уровня воды	1
1 Пар имеет неприятный запах	1
2 Не работает	1
3 Не охлаждает воздух	1
4 Выключается сам по себе	1

Количество заявок по типам проблем

Количество	Количество
1	1
0.5	1
1	1
0.5	1
1	1

Количество

Количество

## Статистика