|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_РАДИОТЕХНИЧЕСКИЙ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***Распределённая информационная система обмена\_\_***

***сообщениями в реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_РТ5-61Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_** **Н.Д. Голумин \_**\_\_\_\_\_

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_А.И. Канев\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_ИУ5\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_В.И. Терехов\_\_

(И.О.Фамилия)

«\_09\_» \_\_\_\_февраля\_\_\_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине \_\_\_\_Сетевые технологии в АСОИУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_РТ5-61Б\_\_

\_\_\_\_\_\_\_\_\_\_ Голумин Н.Д\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсовой работы Распределенная информационная система обмена сообщениями\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_в реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность КР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_УЧЕБНАЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_КАФЕДРА\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения работы: 25% к \_3\_ нед., 50% к \_8\_ нед., 75% к 12 нед., 100% к 15 нед.

***Задание*** Разработать автоматизированную распределенную систему для обмена сообщениями в\_\_\_\_ реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление курсовой работы:***

Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_09\_» \_\_\_\_февраля\_\_\_\_\_\_\_ 2024 г.

**Руководитель курсовой работы**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_А.И. Канев\_\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**\_\_\_ Н.Д. Голумин \_\_\_\_

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_РАДИОТЕХНИЧЕСКИЙ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***Распределённая информационная система обмена\_\_***

***сообщениями в реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_РТ5-61Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **А.А. Надыршина \_**\_\_\_\_

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_А.И. Канев\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_ИУ5\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_В.И. Терехов\_\_

(И.О.Фамилия)

«\_09\_» \_\_\_\_февраля\_\_\_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине \_\_\_\_Сетевые технологии в АСОИУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_РТ5-61Б\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Надыршина А.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсовой работы Распределенная информационная система обмена сообщениями\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_в реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность КР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_УЧЕБНАЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_КАФЕДРА\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения работы: 25% к \_3\_ нед., 50% к \_8\_ нед., 75% к 12 нед., 100% к 15 нед.

***Задание*** Разработать автоматизированную распределенную систему для обмена сообщениями в\_\_\_\_ реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление курсовой работы:***

Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_09\_» \_\_\_\_февраля\_\_\_\_\_\_\_ 2024 г.

**Руководитель курсовой работы**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_А.И. Канев\_\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**\_\_\_ А.А. Надыршина\_

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_РАДИОТЕХНИЧЕСКИЙ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***Распределённая информационная система обмена\_\_***

***сообщениями в реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_РТ5-61Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **И.А. Крайников \_**\_\_\_\_

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_А.И. Канев\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2024 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_ИУ5\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_В.И. Терехов\_\_

(И.О.Фамилия)

«\_09\_» \_\_\_\_февраля\_\_\_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине \_\_\_\_Сетевые технологии в АСОИУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы \_РТ5-61Б\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Крайников И.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Тема курсовой работы Распределенная информационная система обмена сообщениями\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_в реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность КР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_УЧЕБНАЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_КАФЕДРА\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения работы: 25% к \_3\_ нед., 50% к \_8\_ нед., 75% к 12 нед., 100% к 15 нед.

***Задание*** Разработать автоматизированную распределенную систему для обмена сообщениями в\_\_\_\_ реальном времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление курсовой работы:***

Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания «\_09\_» \_\_\_\_февраля\_\_\_\_\_\_\_ 2024 г.

**Руководитель курсовой работы**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_А.И. Канев\_\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**\_\_\_ И.А. Крайников\_\_\_

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 8](#_Toc164110846)

[1. Транспортный уровень 10](#_Toc164110847)

[2. Канальный уровень 12](#_Toc164110848)

[3. Прикладной уровень 14](#_Toc164110849)

[Заключение 18](#_Toc164110850)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc164110851)

[ПРИЛОЖЕНИЕ 1 Техническое задание 20](#_Toc164110852)

[ПРИЛОЖЕНИЕ 2 Программа и методика испытаний 27](#_Toc164110853)

[ПРИЛОЖЕНИЕ 3 Руководство пользователя 38](#_Toc164110854)

[ПРИЛОЖЕНИЕ 4 Руководство системного администратора 44](#_Toc164110855)

# **ВВЕДЕНИЕ**

История индустрии обмена сообщениями началась в 1961 году в Массачусетском технологическом институте, когда была создана первая система обмена данными, с помощью которой могли одновременно общаться до 30 пользователей в режиме, приближенном к реальному времени. В 1971 был создан первый в мире мессенджер, который получил название EMISARI. EMISARI предназначался для обмена короткими текстовыми сообщениями между госслужащими, в случае если было необходимо координировать решение различных ситуаций в стране, требующих оперативного обмена информацией. Информационные системы обмена сообщениями в реальном времени остаются актуальными и важными в современном мире. Сегодня более 2,2 млрд человек по всему миру зарегистрированы хотя бы в одном из популярных приложений для общения.

Целью работы является реализация системы для обмена сообщениями в реальном времени, состоящей из веб-сервиса и веб-приложения.

Система предназначена для обмена текстовыми сообщениями между пользователями в режиме реального времени. Чтобы отправить или прочитать сообщение пользователь должен войти в систему, введя свое имя. При отправке сообщения отображается имя отправителя, само сообщение и время отправки. В случае ошибки отправитель видит значок ошибки и сообщение «При отправке сообщения возникла ошибка». При обновлении страницы история чата не сохраняется.

Нефункциональные требования к разрабатываемой системе:

1. Должна поддерживаться кроссплатформенность.

2. Интерфейс системы и текст ошибок должны быть русифицируемы.

В ходе работы необходимо выполнить следующие задачи:

1. Разработать дизайн приложения.

2. Разработать сервис канального уровня на Django.

3. Разработать сервис транспортного уровня на Go.

4. Реализовать интерфейс приложения на React.

5. Подготовить набор документации, включающий РПЗ, ТЗ, ПМИ, РСА, РП.

# **Транспортный уровень**

Транспортный уровень является связующим между прикладным и канальным уровнем. На него ложиться вся основная нагрузка бэкенда приложения. Пронаблюдать логику и важность транспортного уровня можно на диаграмме развертывания (рис. 1).

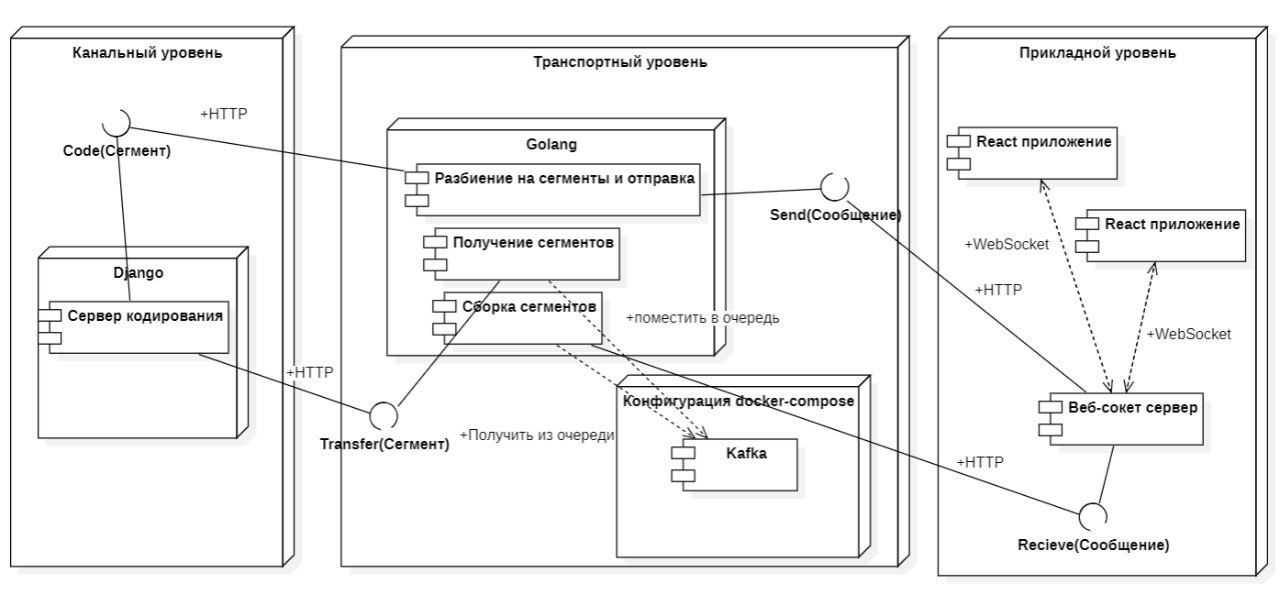
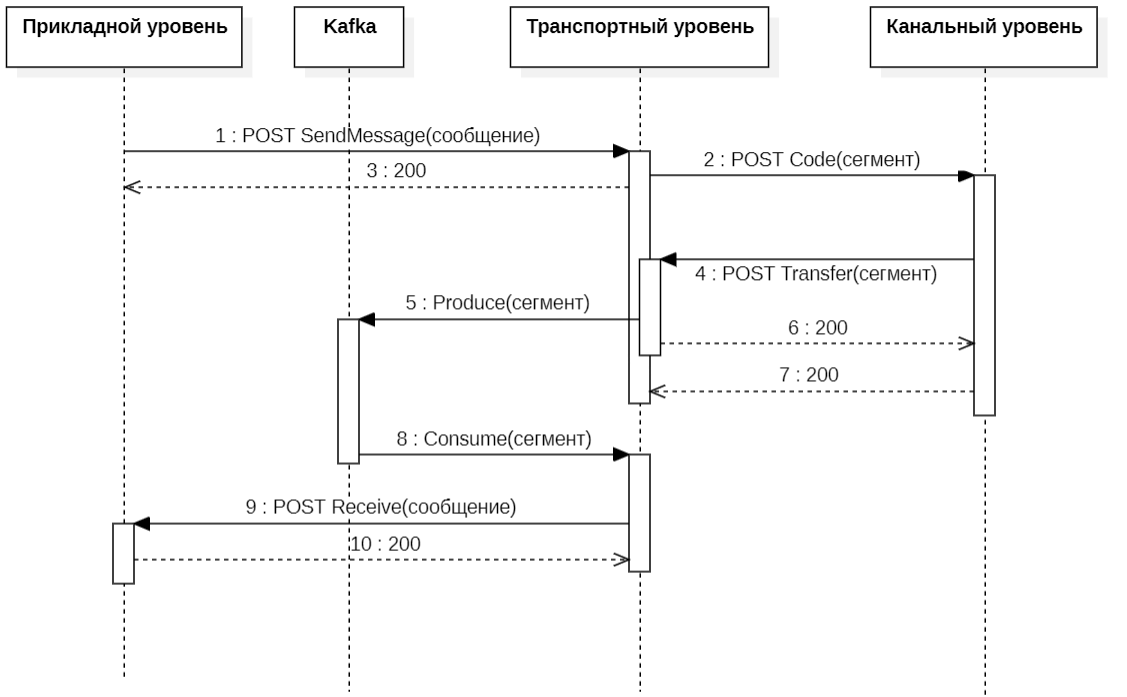


Рисунок 1 – Диаграмма развертывания приложения

Алгоритм взаимодействия транспортного уровня с прикладным и канальным можно пронаблюдать на диаграмме последовательности транспорого уровня (рис. 2).

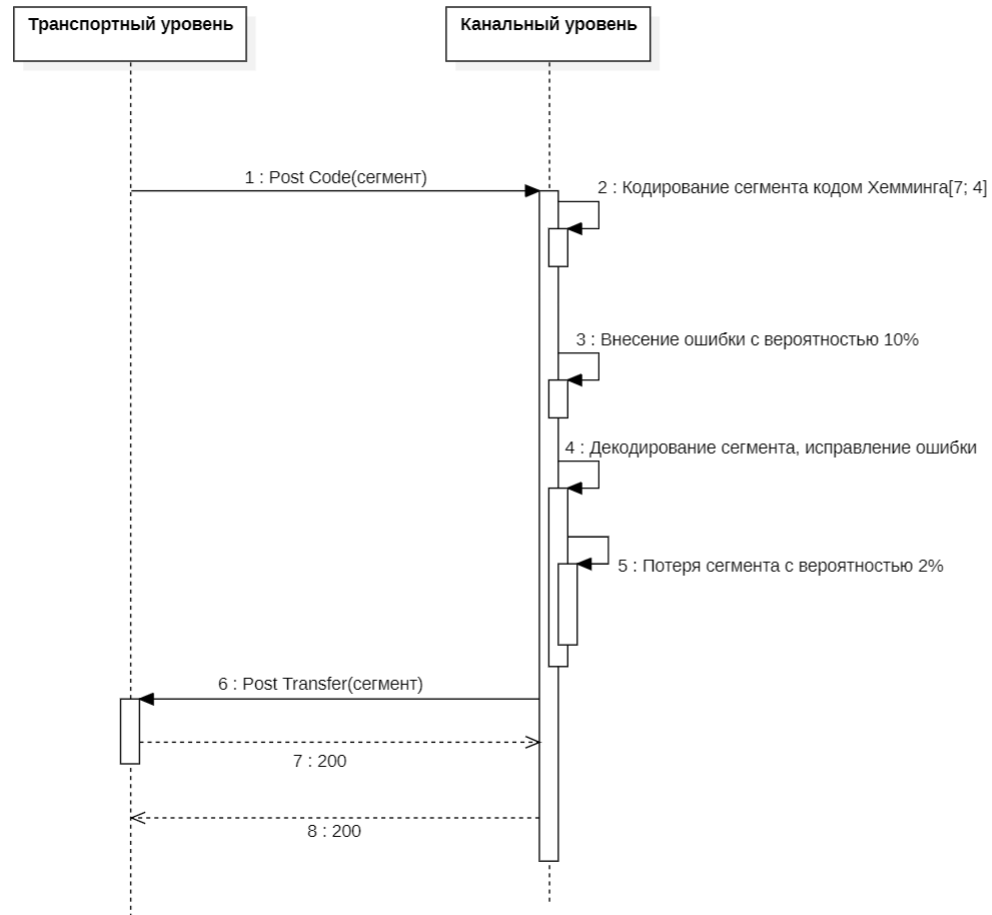
 Рисунок 2 – Диаграмма последовательности транспортого уровня

Взаимодействие начинается с вызова метода SendMessage с прикладного уровня на транспортный, в теле которого передается сообщение отправленное одним из пользователей. После чего на трансортном уровне происходит сегментация сообщения в блоки по 300байт и отправка каждого сегмента на канальный уровень посредством метода Code. После обработки сегмента на канальном уровне он возвращается на транспортный уровень с помощью метода Transfer и следом отправляется в очередь Kafka с помощью метода Produce. Раз в 1 секунду сегменты в очереди собираются в единое сообщение посредством метода Consume и отправляются, как собранное сообщение, на прикладной уровень методом Receive.

# **Канальный уровень**

Данный уровень эмитирует взаимодействие с удаленным сетевым узлом через канал с помехами. Для этого используется алгоритм Хэмминга[7;4] для кодирования передаваемых сообщений.

Алгоритм работы системы на канальном уровне отображен на диаграмме последовательностей (рис. 3)

 Рисунок 3 – Диаграмма последовательности канального уровня

Канальный уровень получает сегмент 300 байт от транспортного уровня, после чего кодирует сегмент кодом Хэмминга[7;4] и вносит ошибку с вероятностью 10% в случайный бит. Далее происходит декодирование сегмента и исправление однократной ошибки, в случае если она была допущена. После чего на канальном уровне сегмент теряется с вероятностью 2% и, если он не потерялся отправляется транспортному уровню с помощью метода Transfer.

# **Прикладной уровень**

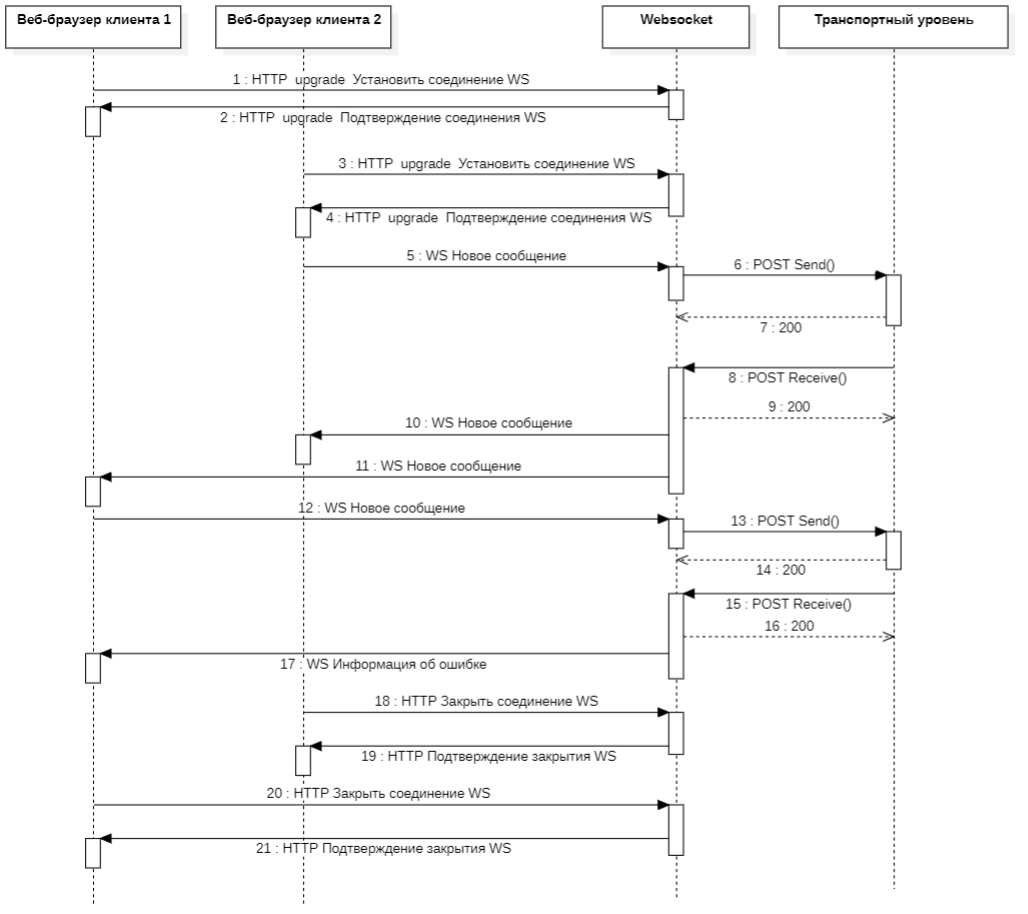
Алгоритм работы системы отображен на диаграмме последовательности (рис. 4).

Рисунок 4 – Диаграмма последовательности прикладного уровня

В начале при подключении к системе происходит установка WebSocket соединения. Для этого пользователь отправляет через графический интерфейс запрос, передавая в нем свое имя. Далее пользователь получает подтверждение соединения WebSocket и перейдет на страницу чата. Затем пользователь вводит сообщение и, нажимая на кнопку «Отправить», отправляет запрос на отправку сообщения, содержащий имя отправителя, время отправки и текст, на транспортный уровень. Далее при отсутствии ошибки участники чата видят сообщение. Если сообщение пришло с признаком ошибки, то отправитель видит значок ошибки и сообщение об ошибке, а получатель не видит ничего. При нажатии на кнопки «Выйти» закрывается WebSocket соединение, сбрасывается имя пользователя и очищается чат.

При запуске приложения пользователь видит стартовую страницу (рис. 5).

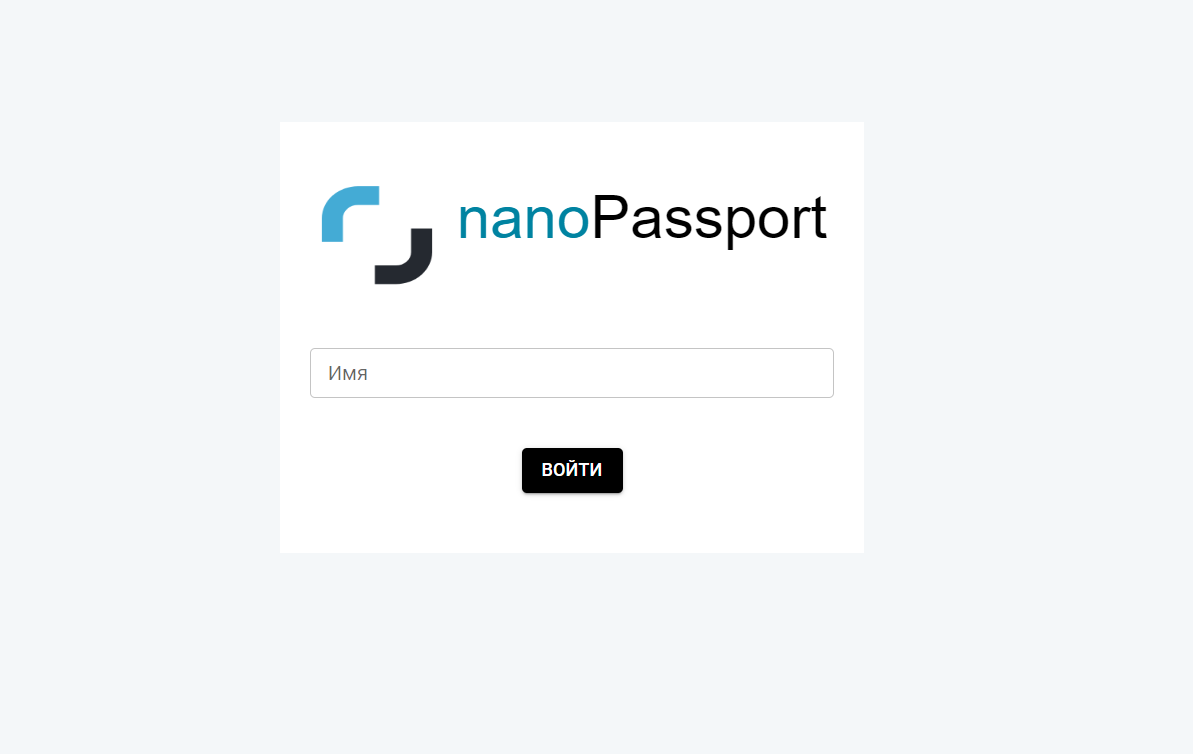


Рисунок 5 – Стартовая страница

Пользователь вводит свое имя и нажимает на кнопку «Войти». После этого открывается WebSocket соединение, пользователь переходит на страницу чата (рис. 6).

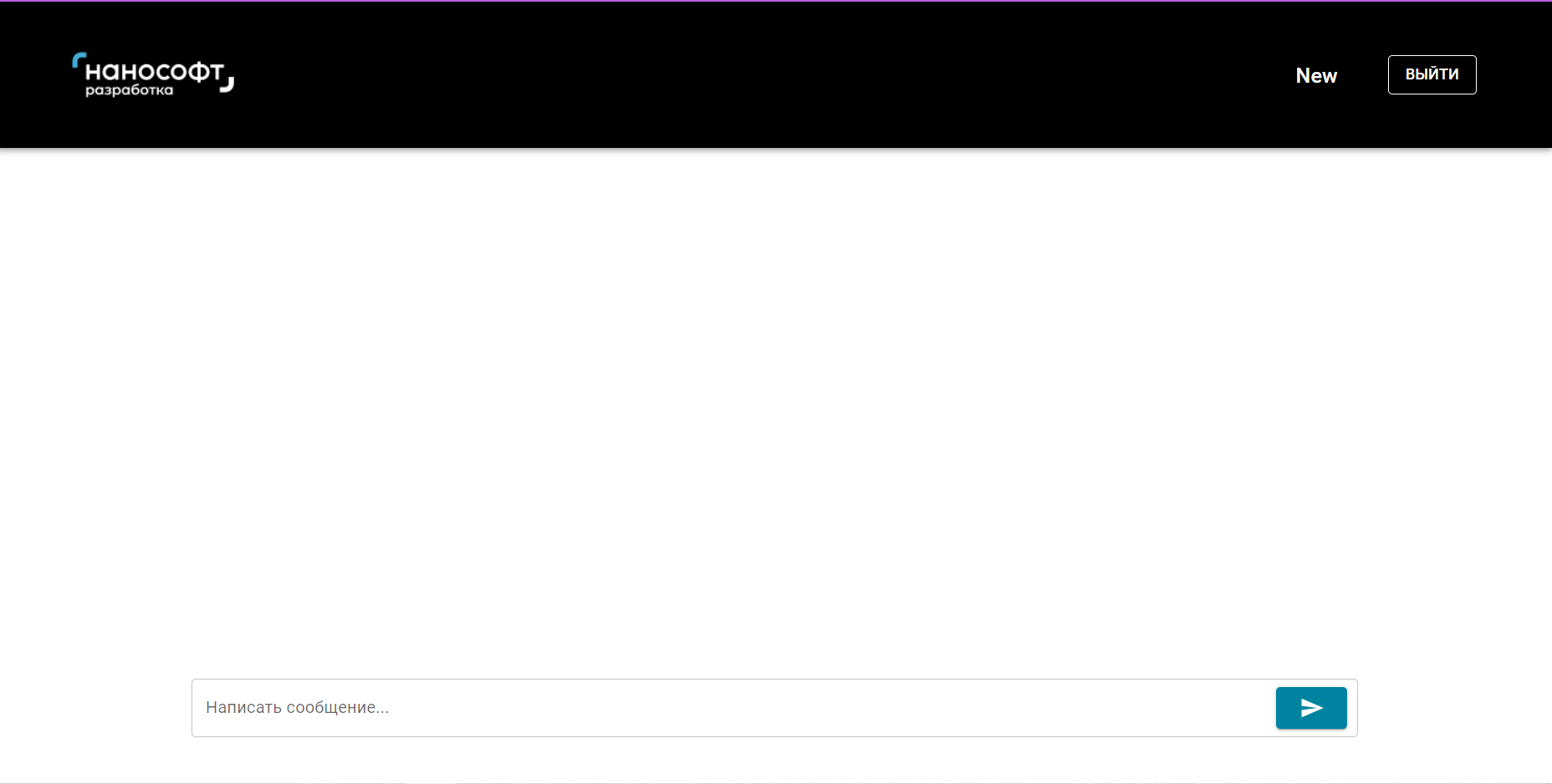


Рисунок 6 – Страница чата

Пользователь вводит в поле ввода сообщение и, нажимая на кнопку «Отправить», отправляет его. При отсутствии ошибки пользователи чата видят свое имя, текст сообщения и время отправки (рис. 7).

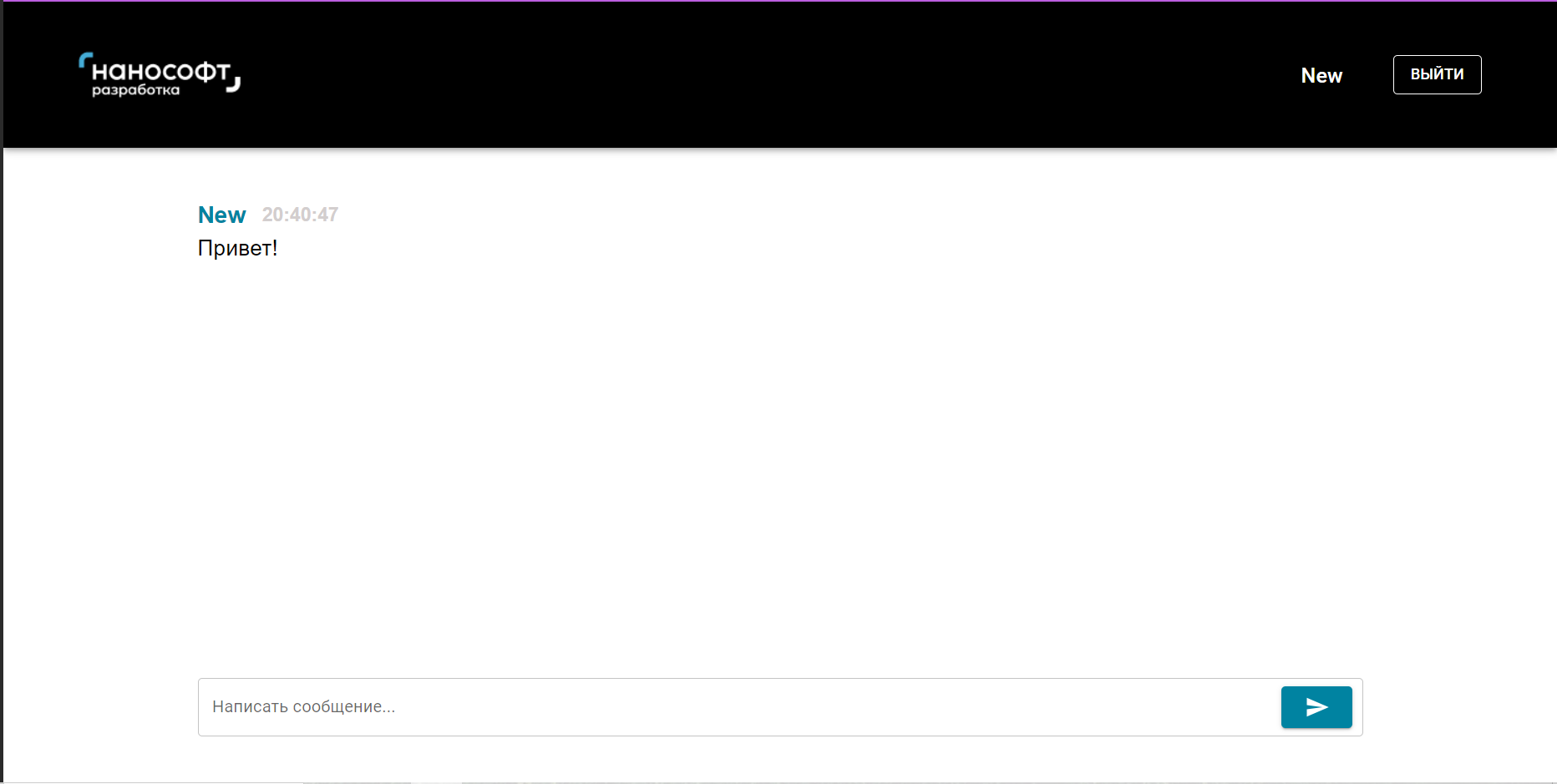


Рисунок 7 – Отправление сообщения без признака ошибки

Если сообщение пришло с признаком ошибки, отправитель видит значок ошибки и сообщение об ошибке (рис. 8).

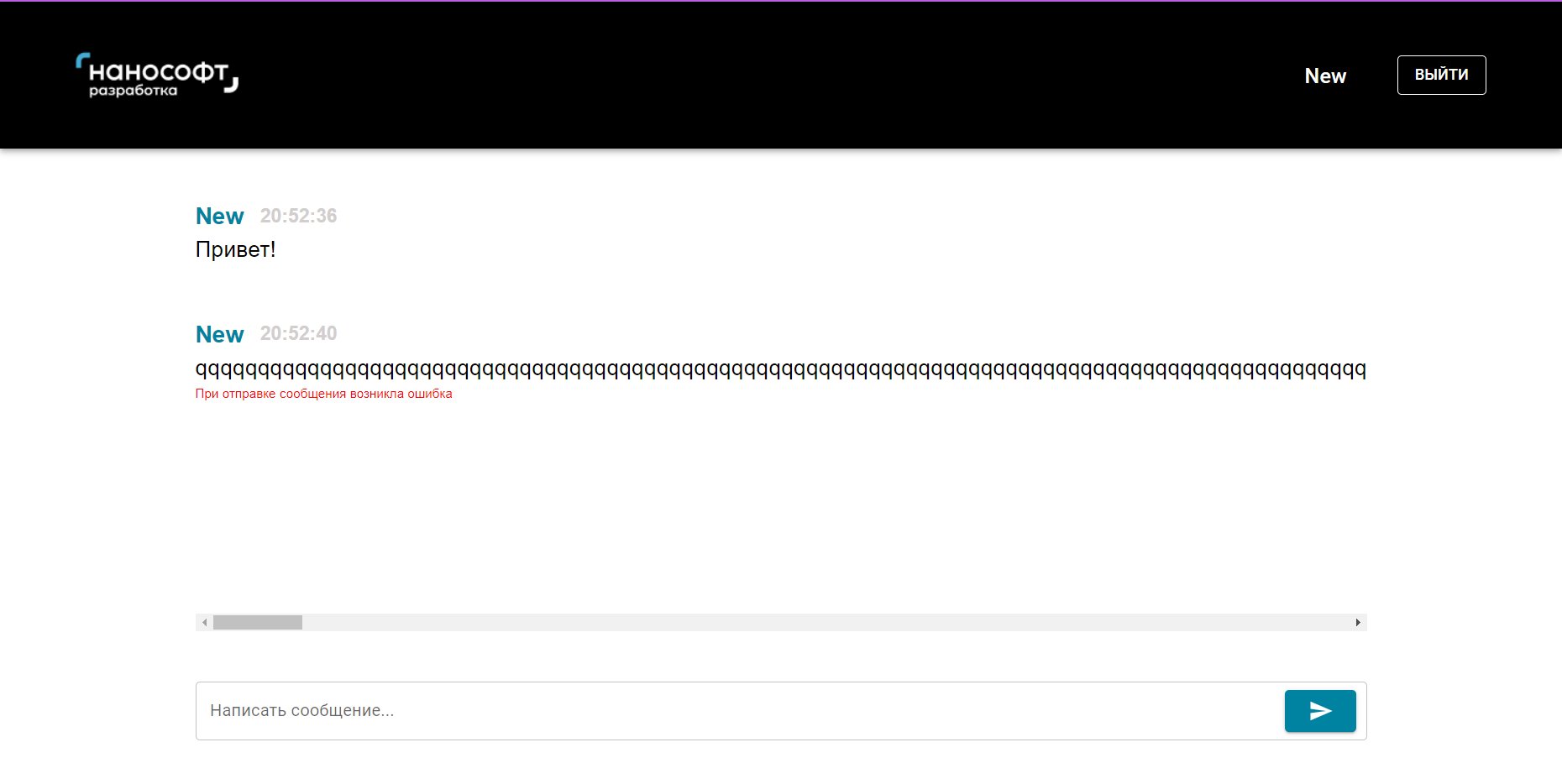


Рисунок 8 – Отправление сообщения с признаком ошибкой

При получении сообщения получатель видит имя отправителя, текст сообщения и время отправки (рис. 9).

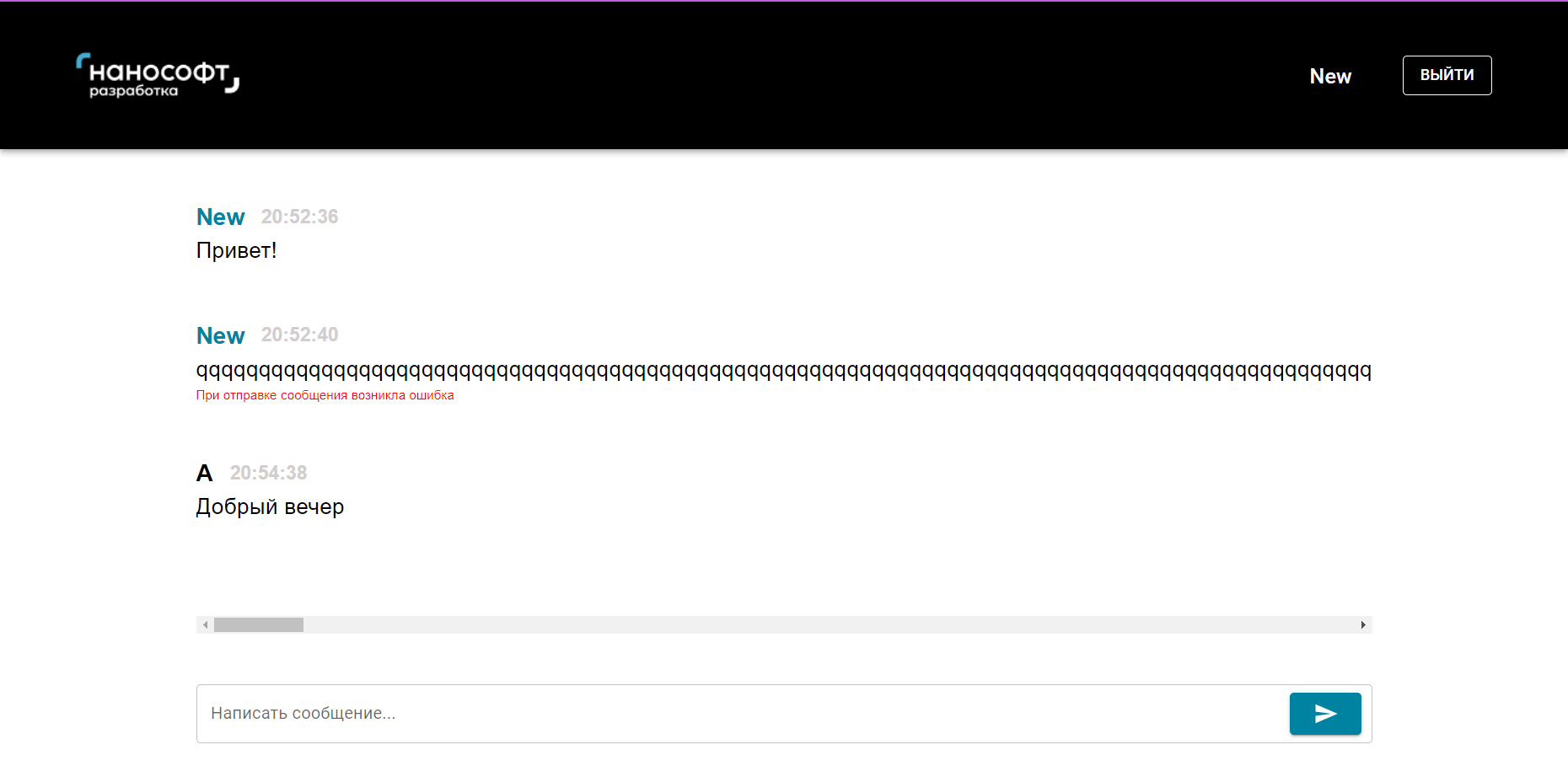


Рисунок 9 – Получение сообщения

При нажатии на кнопку «Выйти» происходит закрытие WebSocket соединения, сброс имени и очистка чата. Пользователь возвращается на стартовую страницу.

# **Заключение**

В ходе работы были достигнуты следующие результаты:

1. Разработан дизайн приложения в Figma. Ссылка: https://www.figma.com/file/1Yzy7oDf0wCm7D7hxwQfWv/Untitled?type=design&node-id=72%3A45&mode=design&t=yLzAOCo9OotszMxm-1
2. Был разработан сервис канального уровня на Django.
3. Был разработан сервис транспортного уровня на Go.
4. Разработан интерфейс приложения с использованием технологии React.
5. Подготовлен набор документации, включающий РПЗ, ТЗ, ПМИ, РСА, РП.
6. Исходный код проекта доступен в GitHub:

Прикладной уровень: https://github.com/alyona-nad/application\_layer

Канальный уровень: https://github.com/Dark1ord15/channel\_layer

Транспортный уровень: https://github.com/cry1s/transport\_layer

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Официальная документация Python: [Электронный ресурс]. // URL: https://www.python.org/doc/ (Дата обращения: 7.04.2024)
2. Официальная документация Django: [Электронный ресурс]. // URL: https://docs.djangoproject.com/en/4.2/ (Дата обращения: 7.04.2024)
3. Официальная документация React: [Электронный ресурс]. // URL: https://legacy.reactjs.org/docs/getting-started.html (Дата обращения: 15.04.2024)
4. Официальная документация MUI: [Электронный ресурс]. // URL: https://mui.com/material-ui/getting-started/overview/ (Дата обращения: 16.04.2024)
5. Руководство по Go: [Электронный ресурс] // Metanit. URL: https://metanit.com/go/tutorial/ (дата обращения: 11.04.2024)
6. Официальная документация Apache Kafka: [Электронный ресурс] // URL: https://kafka.apache.org/documentation/ (дата обращения: 11.04.2024)

# **ПРИЛОЖЕНИЕ 1 Техническое задание**

|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Утверждаю

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_\_"\_\_\_\_\_\_\_\_\_2024г.

**Факультет «Радиотехнический»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Сетевые технологии в АСОИУ»

Техническое задание

Вариант 43

Студенты группы РТ5-61Б:

Голумин Н.Д.

Надыршина А.А.

Крайников И.А.

2024г.

1. **Наименование:**

Распределённая информационная система обмена сообщениями в реальном времени.

1. **Основание для разработки:**

Основанием для разработки является учебный план МГТУ им. Баумана кафедры ИУ5 на 6 семестр.

1. **Исполнители:**

Надыршина А. А. (прикладной уровень) – группа РТ5-61Б

Крайников И. А. (транспортный уровень) – группа РТ5-61Б

Голумин Н. Д. (канальный уровень) – группа РТ5-61Б

1. **Цель разработки:**

Разработать распределённую систему для обмена сообщениями в реальном времени, состоящую из трех уровней: прикладной, транспортный и канальный, каждый из которых реализуется отдельным веб-сервисом.

1. **Функциональные требования**
   1. Прикладной уровень:
      1. Стартовая страница:
         1. При запуске приложения для пользователя открывается стартовая страница с названием компании, полем для ввода имени, которое будет передаваться с каждым новым сообщением, и кнопкой «Войти».
         2. При нажатии на кнопку «Войти» открывается WebSocket соединение и осуществляется переход на страницу 5.1.2.
      2. Страница приложения с окном чата для отправки и просмотра полученных сообщений с указанием отправителя и времени отправки:
         1. Получение сообщений. При отправке пользователю сообщения другим человеком, пользователь видит имя отправителя, сообщение и дату отправки сообщения.
         2. Отправка сообщений. При нажатии на кнопку «Отправить» отображается имя пользователя, сообщение и дата отправки сообщения.
         3. При нажатии на кнопку «Выйти» чат и логин очищаются, ws-подключение закрывается и осуществляется переход на страницу 5.1.1.
         4. В случае, если сообщение пришло с признаком ошибки, сообщение не отображается, а вместо него у получателей появляется значок ошибки и соответствующее сообщение.
         5. Дизайн приложения соответствует сайту компании «Нанософт».
      3. WebSocket-сервер:
         1. Хранит имена пользователей для всех ws-подключений.
         2. Позволяет устанавливать, закрывать ws-соединения, получать сообщения от клиентов и широковещательно рассылать сообщения подключенным клиентам.
      4. Реализация HTTP-метода Receive для получения сообщения с транспортного уровня:
         1. В json каждого сообщения указывается отправитель, время отправки, признак ошибки и полезная нагрузка.
         2. Полученное по HTTP сообщение отправляется широковещательной WebSocket рассылкой всем подключенным ws-клиентам, кроме тех, у кого логин совпадает с именем отправителя.
   2. Транспортный уровень:
      1. Реализация HTTP-метода Send для сегментирования сообщений:
         1. Разбиение сообщения на сегменты по 300 байт и их поочередная отправка на канальный уровень через метод Code.
         2. Каждый сегмент содержит время отправки (в качестве идентификатора сообщения), общую длину сообщения, номер данного сегмента в сообщении, полезную нагрузку.
      2. Реализация HTTP-метода Transfer для передачи сообщения на прикладной уровень:
         1. Формирование очереди для полученных сегментов, которые раз в 1 секунды собираются в сообщения прикладного уровня.
         2. Если часть из сегментов сообщения не была принята, оно передается на прикладной уровень с признаком ошибки.
   3. Канальный уровень:
      1. Сервис канального уровня эмулирует канал связи с потерями:
         1. Сервис должен вносить ошибку с вероятностью 10% в один случайный бит каждого сформированного кадра.
         2. Сервис должен терять передаваемый кадр с вероятностью 2%.
      2. Реализация HTTP-метода Code для кодирования и декодирования полученного от транспортного уровня сегмента:
         1. Полученный от транспортного уровня json сегмента кодируется кодом Хэмминга [7,4] для получения кадра.
         2. После внесения ошибки в кадр он декодируется с исправлением ошибки и передается далее в виде сегмента на транспортный уровень.
2. **Требования к составу технических средств:**
   1. Прикладной уровень:
      1. Серверная часть
         1. ПК с ОС Windows(7.0 и выше) или ОС Linux(6.4 и выше)
         2. Node.js (1.20 и выше)
      2. Клиентская часть
         1. ПК с ОС Windows(7.0 и выше) или ОС Linux(6.4 и выше)
         2. Веб-браузер: Chrome(40 и выше)
   2. Транспортный уровень:
      * 1. ПК с ОС Windows(7.0 и выше) или ОС Linux(6.4 и выше)
        2. Golang (1.21.0 и выше)
        3. Kafka (2.12 и выше)
   3. Канальный уровень:
      * 1. ПК с ОС Windows(7.0 и выше) или ОС Linux(6.4 и выше)
        2. Django (4.2 и выше)
3. **Этапы разработки:**
   1. Выбрать тему-вариант, определить команду и разработать ТЗ – 3 неделя;
   2. Разработать макет figma, три диаграммы последовательности и описать HTTP-методы в swagger – 8 неделя;
   3. Разработать и отладить приложение, подготовить полный комплект документов – 12 неделя;
   4. Исправить замечания, защитить проект – 14 неделя.
4. **Техническая документация, предъявляемая по окончании работы:**

Расчётно-пояснительная записка, включающая в приложении комплект технической документации на программный продукт, содержащий:

– Приложение 1 – Техническое задание

– Приложение 2 – Программа и методика испытаний

– Приложение 3 – Руководство пользователя

– Приложение 4 – Руководство системного администратора

1. **Порядок приемки работы:**

Приемка работы осуществляется в соответствии с "Программой и методикой испытаний."

Работа защищается перед комиссией преподавателей кафедры.

1. **Дополнительные условия:**

Данное Техническое Задание может дополняться и изменяться в установленном порядке.

# **ПРИЛОЖЕНИЕ 2 Программа и методика испытаний**

|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Утверждаю

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_\_"\_\_\_\_\_\_\_\_\_2024г.

**Факультет «Радиотехнический»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Сетевые технологии в АСОИУ»

Программа и методика испытаний

Вариант 43

Студенты группы РТ5-61Б:

Голумин Н.Д.

Надыршина А.А.

Крайников И.А.

2024г.

# **Объект испытаний:**

Объектом испытания является распределенная информационная система обмена сообщениями в реальном времени.

# **Цель испытаний:**

Целью проведения испытаний является доказательство работоспособности описанного в пункте 1 объекта испытаний.

# **Требования к объекту испытаний:**

Требования к объекту испытаний представлены в документе «Техническое задание».

# **Требования к программной документации:**

Во время проведения испытания должны быть представлены следующие документы:

1. Техническое задание;
2. Программа и методика испытаний.

# **Программа испытаний:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Номер пункта ТЗ | Выполняемое действие | Результат | Иллюстрации |
| 1 | 5.1.1.1 | Запуск приложения. | Приложение запущено. Открывается стартовая страница приложения. | Рисунок 1 |
| 2 | 5.1.2.1 | Заполнение поля имени, нажатие на кнопку «Войти». | Осуществляется переход на страницу чата. В правом верхнем углу отображается имя пользователя. | Рисунок 2,  Рисунок 3 |
| 3 | 5.1.2.1 | Получение сообщения от другого пользователя. | Отображение имени отправителя черным цветом, текста сообщения и времени отправки сообщения. | Рисунок 4 |
| 4 | 5.1.2.2 | Отправка сообщений. Ввод сообщения в специальное поле, нажатие на кнопку «Отправить». | Отображение имени пользователя голубым цветом, текста сообщения и времени отправки сообщения. | Рисунок 5,  Рисунок 6 |
| 5 | 5.1.2.3 | Выход из чата. Нажатие на кнопку «Выйти». | Осуществляется переход на стартовую страницу. | Рисунок 7 |
| 6 | 5.1.2.4 | Отправка сообщения с ошибкой. Ввод сообщения в специальное поле, нажатие на кнопку «Отправить». | Отображение значка ошибки и сообщения «При отправке сообщения возникла ошибка» | Рисунок 8 |
| 7 | 5.1.3 | Запоминание имени для WebSocket подключения. Ввод имени и нажатие на кнопку «Войти». | Открытие WebSocket соединения, запоминание имени пользователя. | Рисунок 9 |
| 8 | 5.1.4 | Метод Receive. | В json сообщения указывается отправитель, текст сообщения, время отправки, признак ошибки. Сообщение отправляется всем участникам чата. | Рисунок 10 |
| 9 | 5.2.1 | Метод Send. Ввод сообщения, нажатие на кнопку «Отправить». | На транспортном уровне сообщение разбивается на сегменты по 300 байт и посегментно отправляется на канальный уровень. В сегменте содержатся время отправки, общая длина сообщения, номер данного сегмента, полезная нагрузка. | Рисунок 11 |
| 10 | 5.2.2 | Метод Transfer. | На транспортном уровне формируется очередь из полученнных сегментов. Сегменты собираются в сообщение раз в 1 секунду. Если часть сегментов не была принята, то сообщение передается на прикладной уровень с признаком ошибки. | Рисунок 12 |
| 11 | 5.3.1 | Метод Code. | Полученный от транспортного уровня json сегмента кодируется кодом Хэмминга [7, 4] в битовый формат. Вносится ошибка в 1 бит сегмента. Далее сегмент декодируется либо с потерей пакета с ошибкой, либо с исправлением ошибки. Затем сегмент передается на транспортный уровень. | Рисунок 13 |

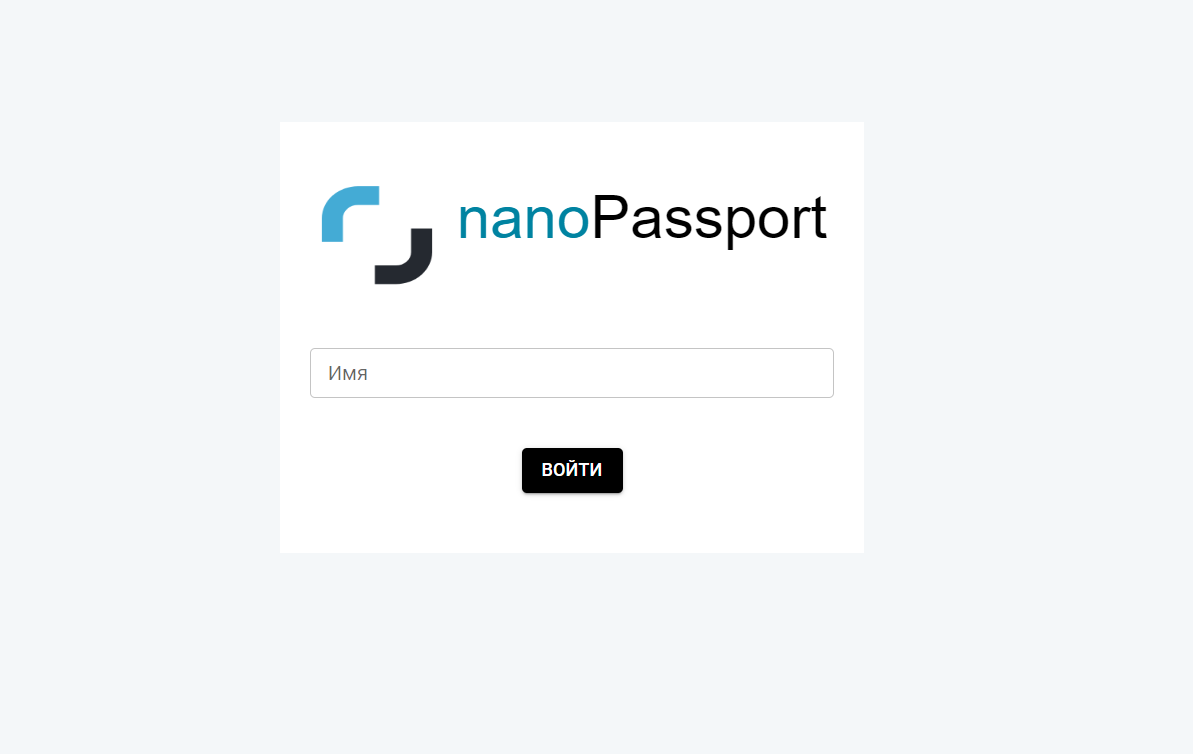
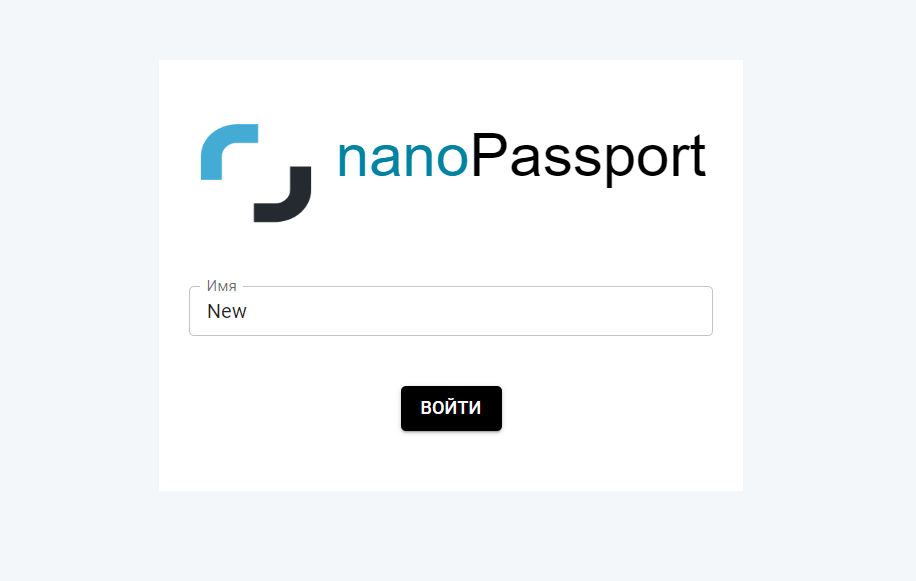


Рисунок 1



Рисунок

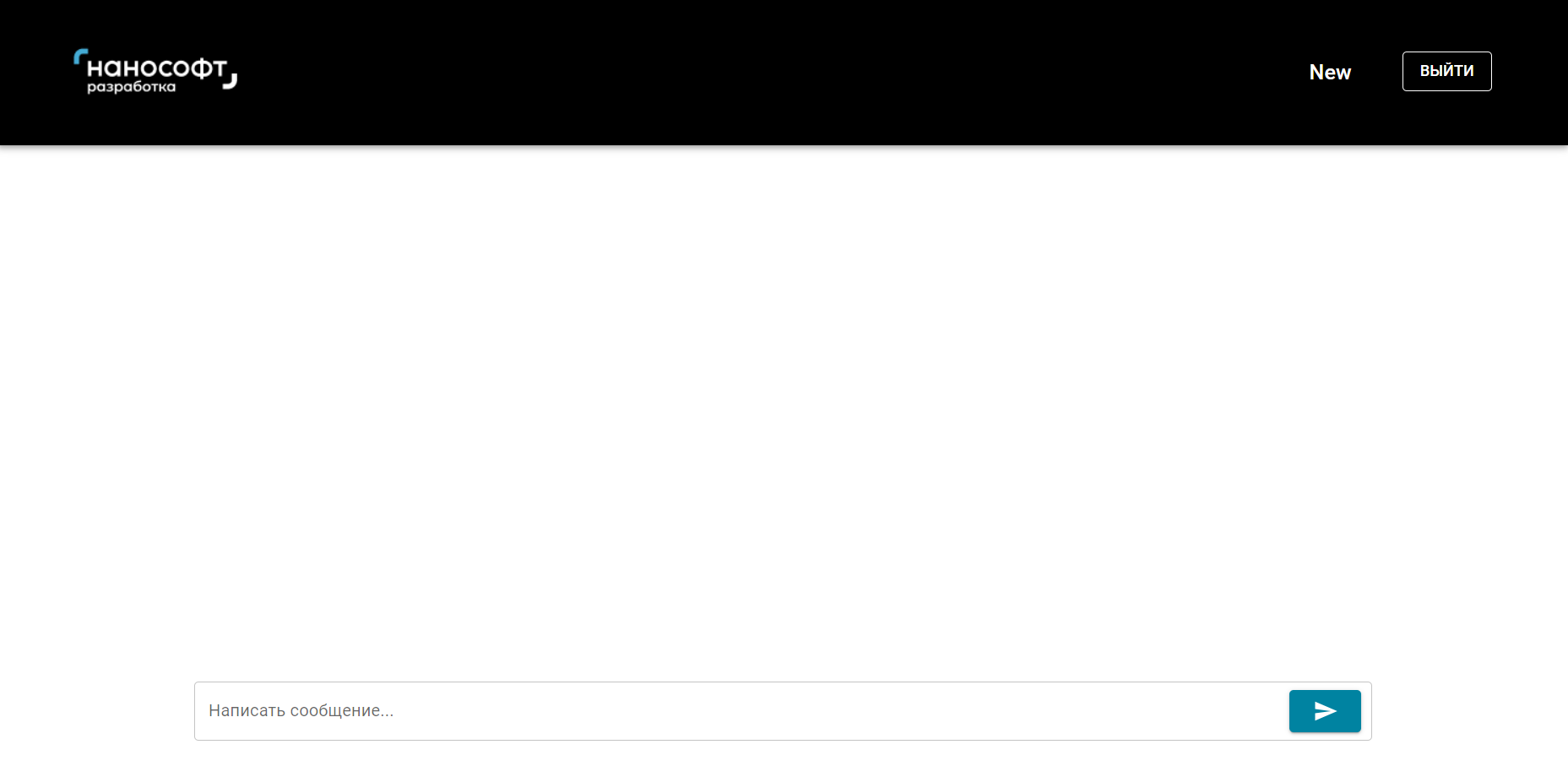


Рисунок 3

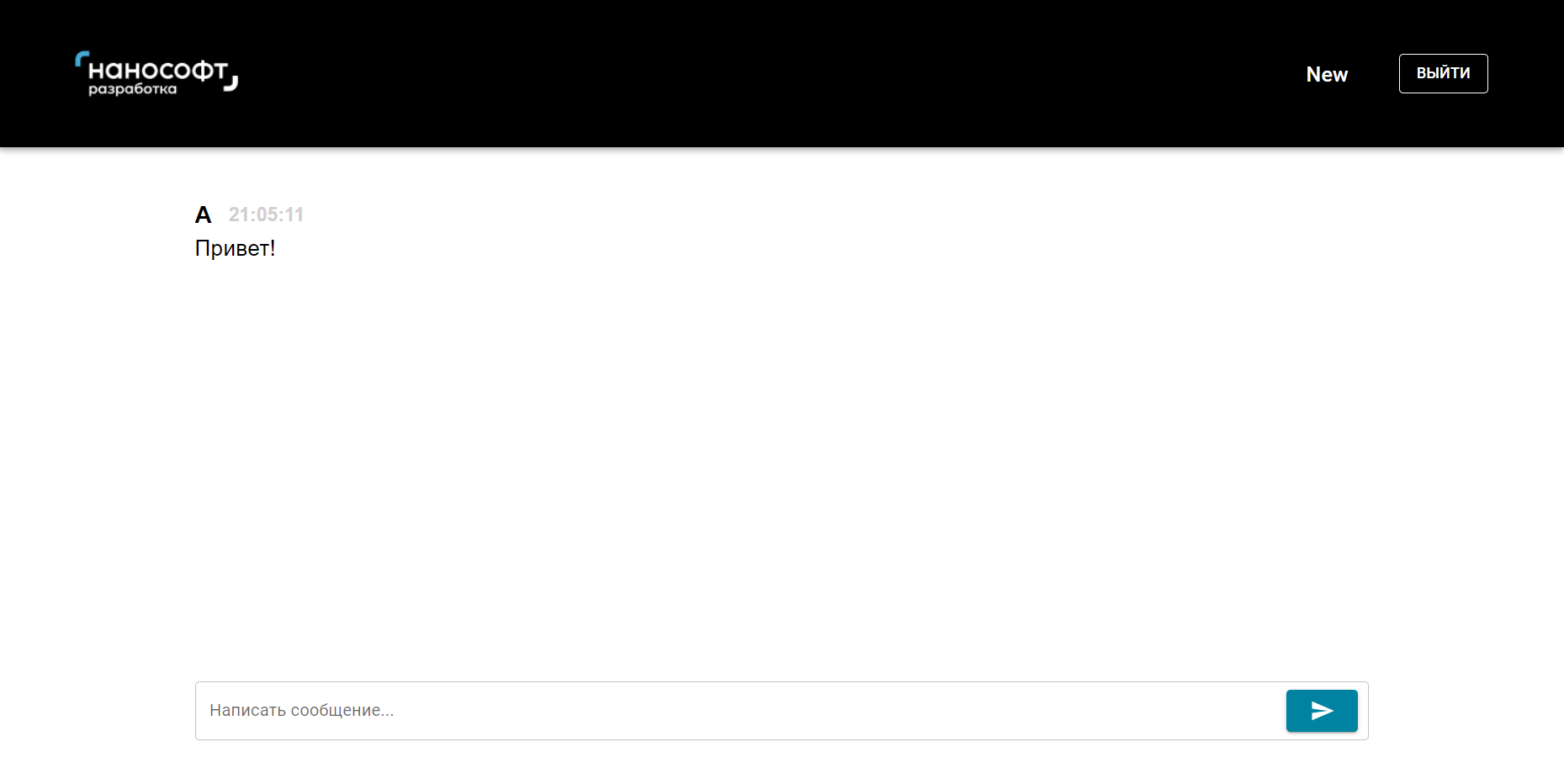
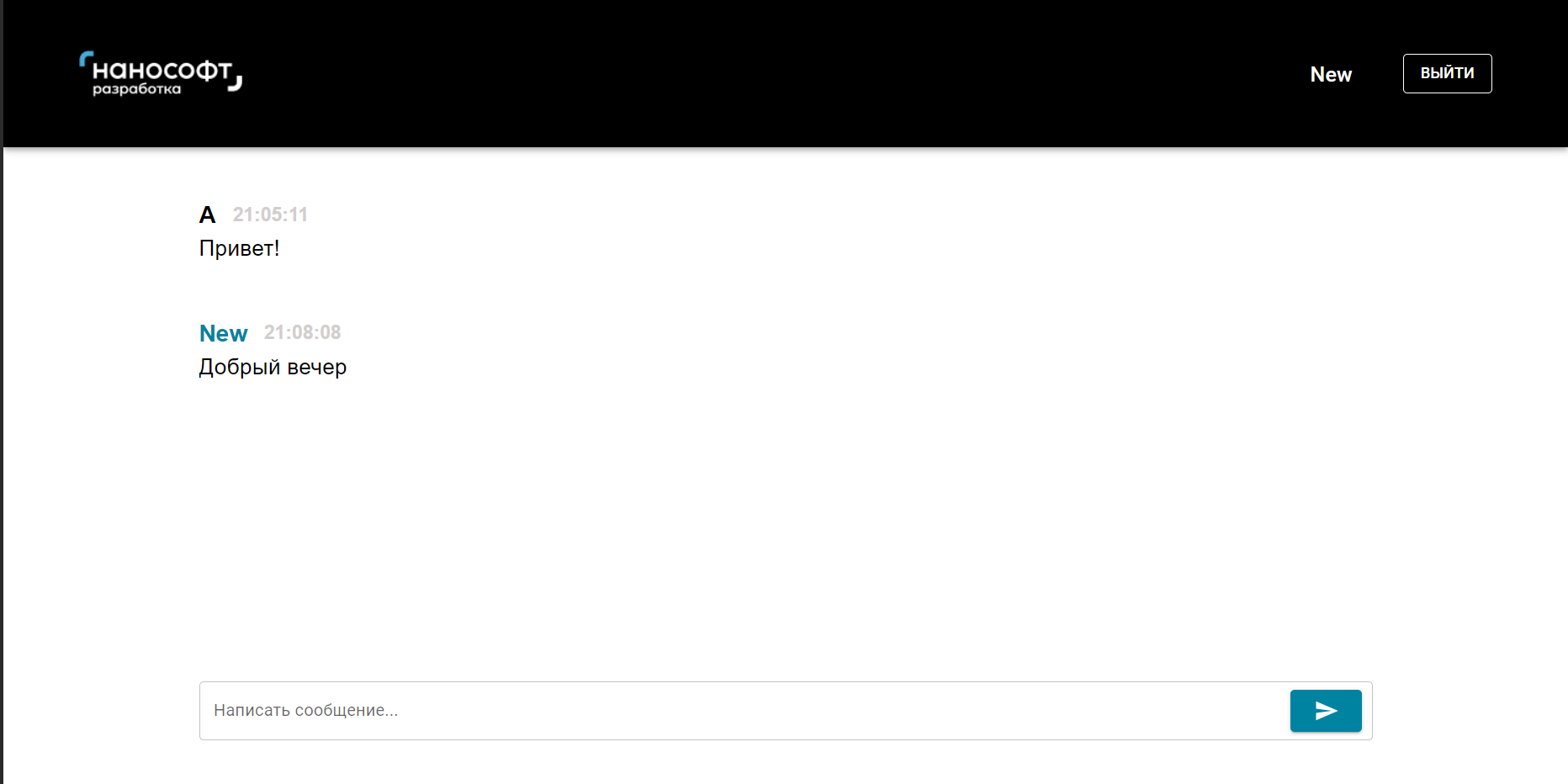


Рисунок 4



Рисунок 5



Рисунок

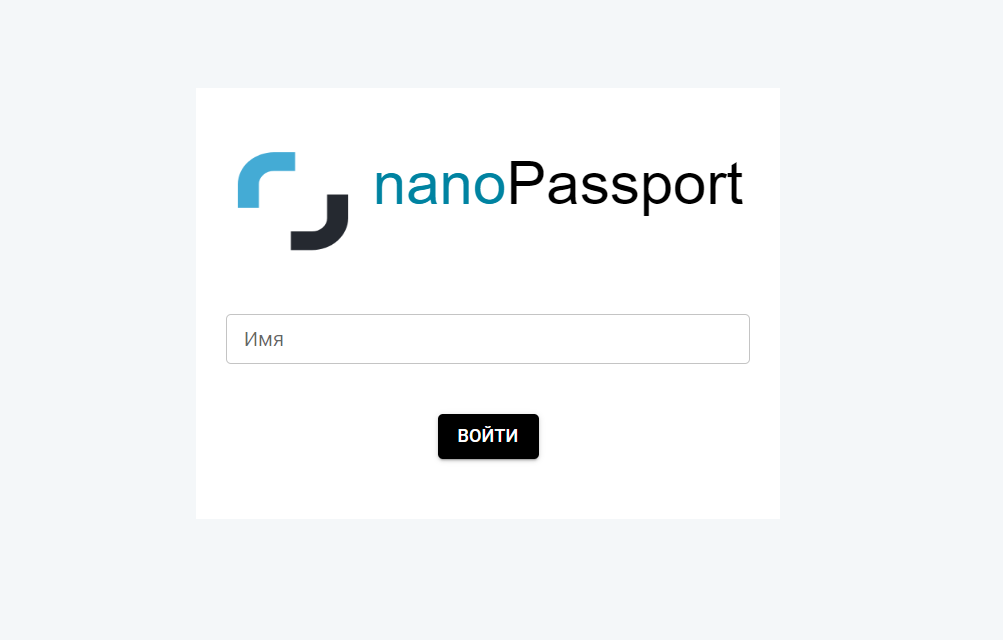


Рисунок 7

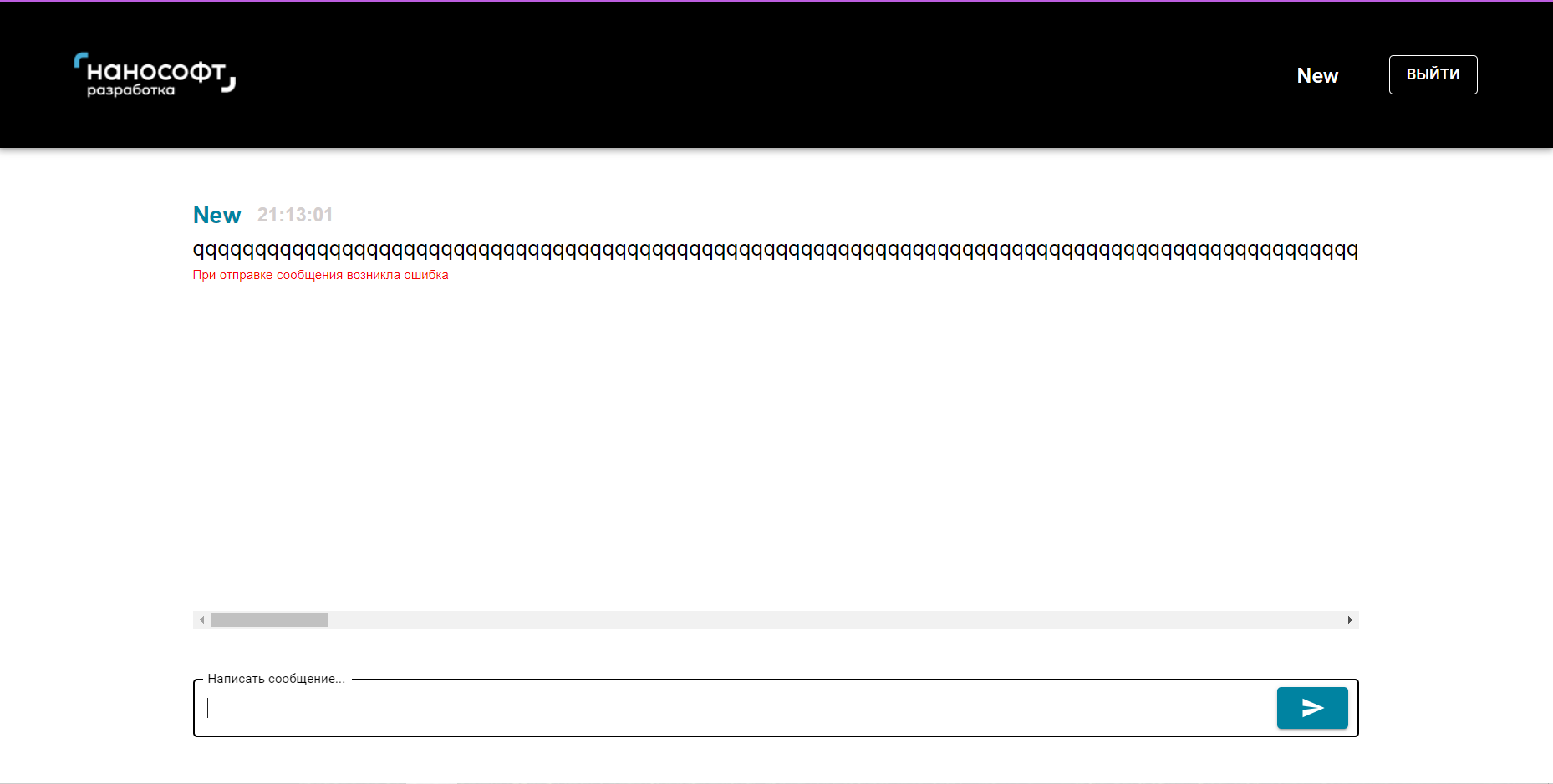


Рисунок 8

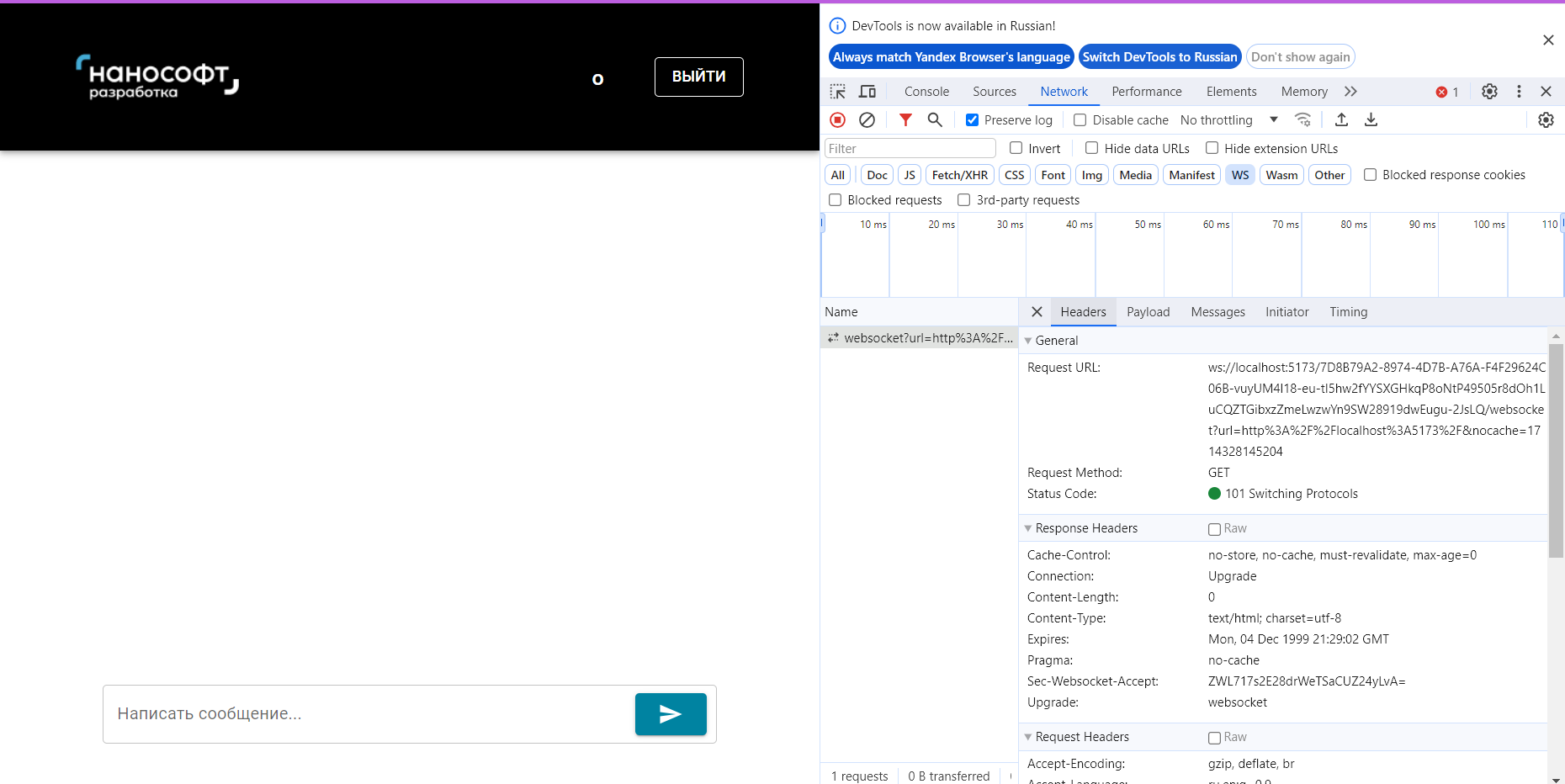


Рисунок 9

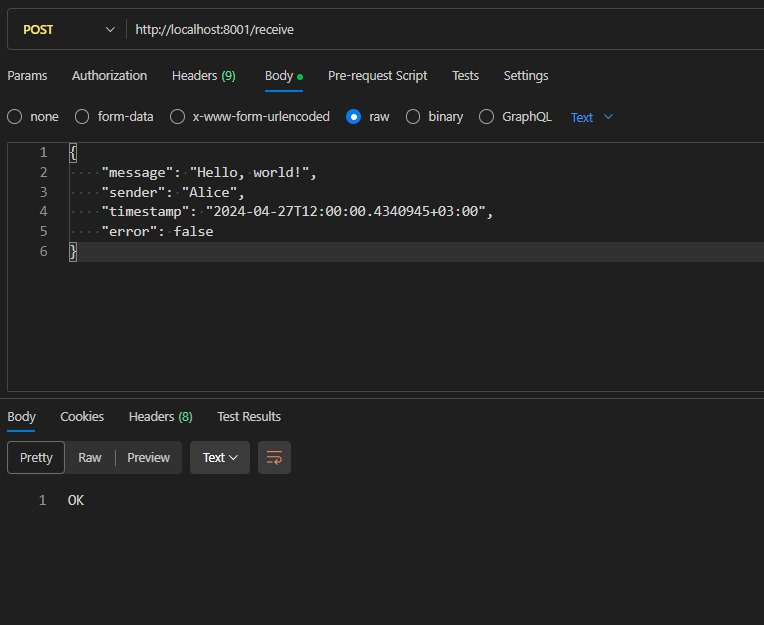


Рисунок 10

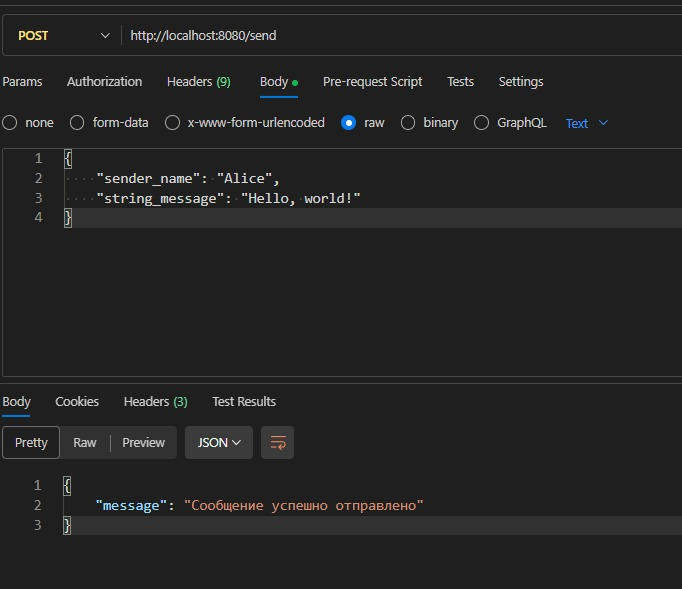


Рисунок 11

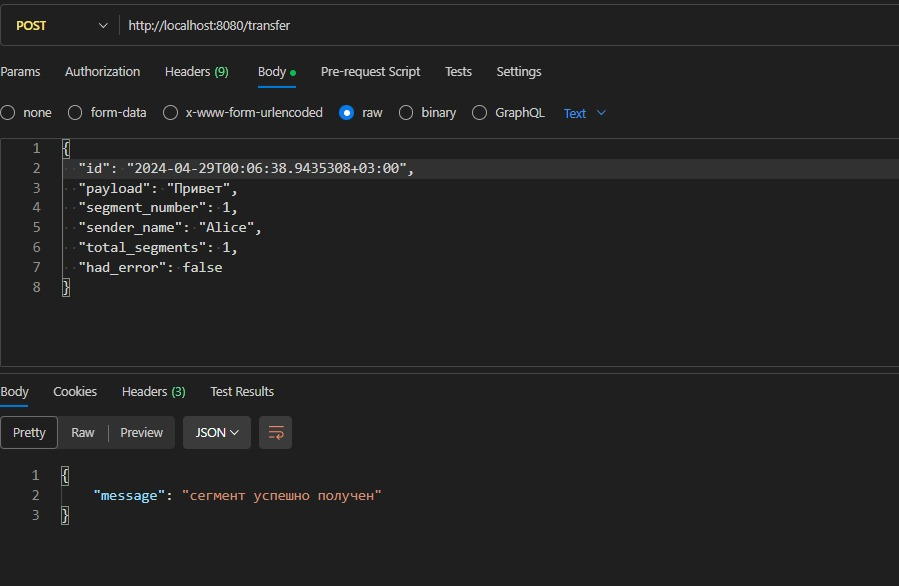


Рисунок 12

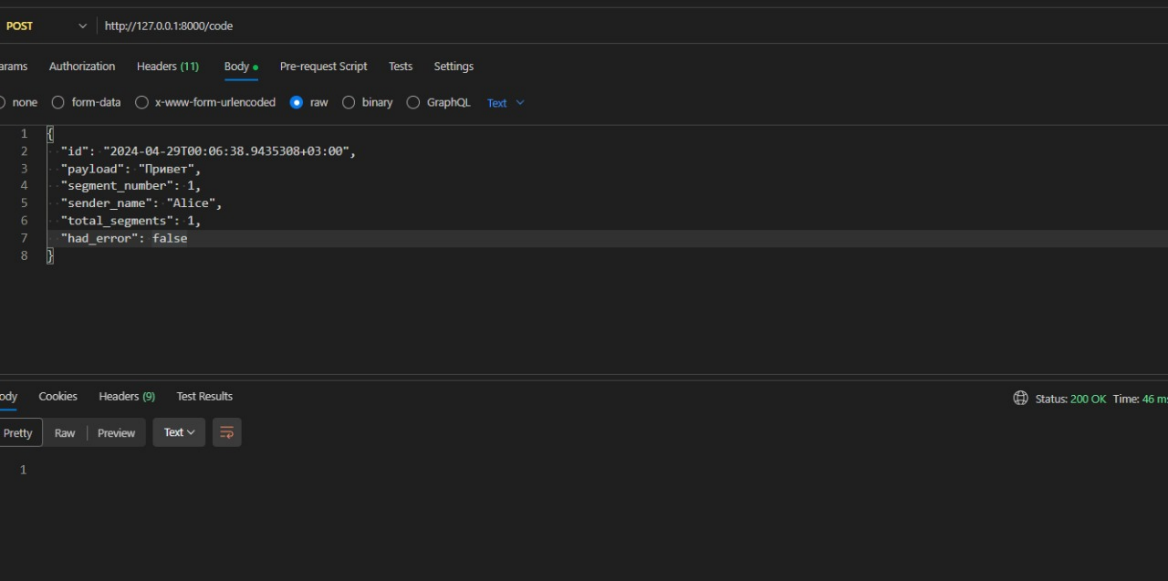


Рисунок 13

# **ПРИЛОЖЕНИЕ 3 Руководство пользователя**

|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Утверждаю

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_\_"\_\_\_\_\_\_\_\_\_2024г.

**Факультет «Радиотехнический»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Сетевые технологии в АСОИУ»

Руководство пользователя

Вариант 43

Студенты группы РТ5-61Б:

Голумин Н.Д.

Надыршина А.А.

Крайников И.А.

2024г.

**Введение**

* 1. **Область применения**

Требования настоящего документа применяются при:

* предварительных комплексных испытаниях;
* опытной эксплуатации;
* приемочных испытаниях;
* промышленной эксплуатации.
  1. **Краткое описание возможностей**

Распределённая система обмена сообщениями, представляющая собой чат для обмена сообщения между пользователями в реальном времени.

Распределенная система предоставляет возможность доступа к чату всем людям, которые перешли по ссылке и ввели свое имя в строку идентификации. При успешной идентификации пользователь получает возможность читать сообщения отправленные другими пользователями, начиная с того времени как он присоединился к чату. Также он может и сам писать и отправлять сообщения тем пользователям, которые находятся вместе с ним в чате.

Актуальность переписки поддерживается при помощи протокола WebSocket, который позволяет обновлять окно чата с сообщениями в реальном времени.

1. **Назначение и условия применения распределенной системы.**
   1. **Назначение распределенной системы.**

Распределенная система предназначена для возможности вести переписку нескольким пользователям в реальном времени.

Для использования Распределенной системы необходимо выполнение следующих условий:

* 1. **Системные требования**

Для работы клиента необходим Yandex Browser или Google Chrome.

1. **Условия выполнения программы**

Для работы программы требуется браузер Google Chrome или любой иной поддерживающий современные функции JavaScript, а также стабильное интернет-соединение.

1. **Выполнение программы**
   1. Инсталляция/деинсталляция
      1. Потребуется Web-браузер, рекомендуется, Yandex Browser или Google Chrome
   2. Запуск программы
      1. На компьютере необходимо запустить браузер и ввести в адресную строку http://localhost:5173/.
2. **Описание операций**
   1. **Идентификация в системе**

Доступно для: все пользователи.

Операция: открыть веб страницу.

Для идентификации в системе необходимо открыть страницу и ввести свое имя (рис.1).

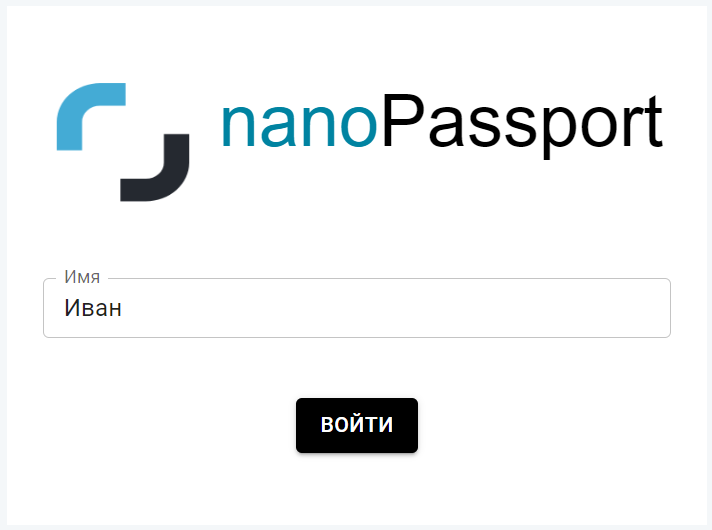


Рисунок 1 – окно идентификации

* 1. **Просмотр и отправка сообщений**

Доступно для: идентифицированным пользователям.

Операция 1: для отправки сообщения нужно вести его в строку для сообщений и нажать на значок «отправить» (рис.2).



Рисунок 2 – отправка сообщения

Операция 2: для получения сообщений от других пользователей нужно находиться на странице чата (рис.3) и ждать сообщения.

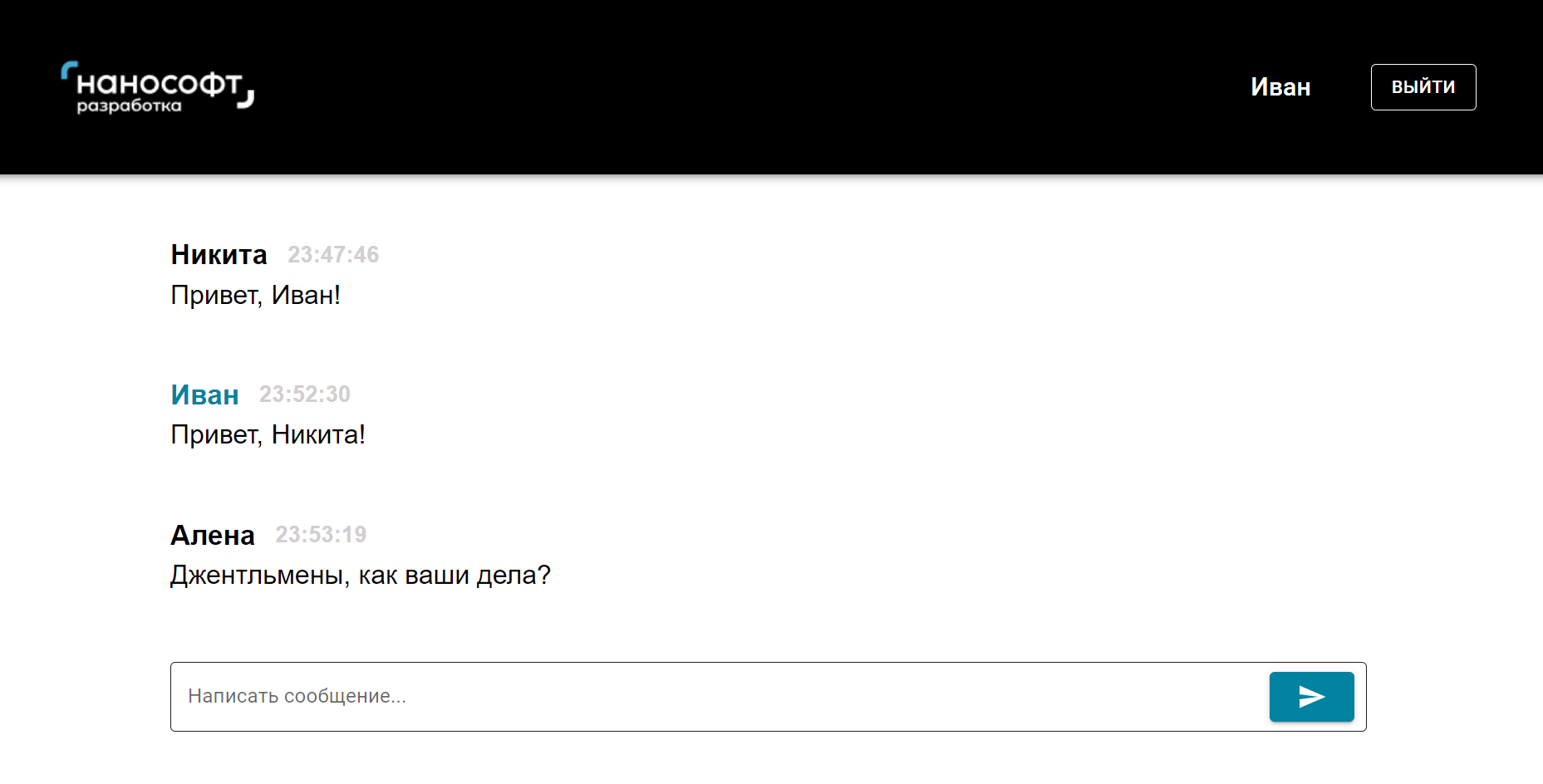


Рисунок 3 – окно чата

* 1. **Выход из чата**

Доступно для: идентифицированным пользователям.

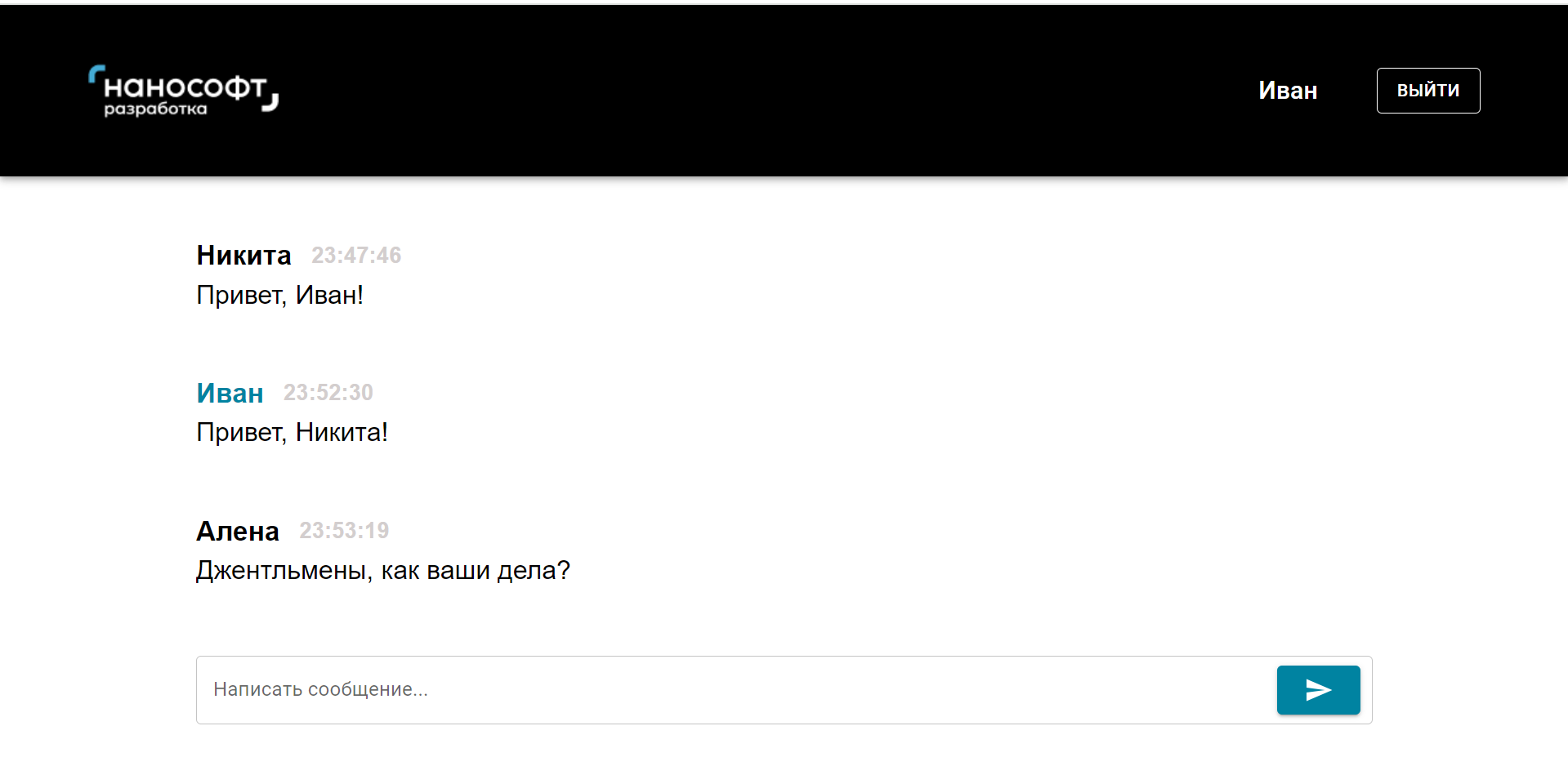
Операция: для выхода из чата необходимо нажать кнопку «выйти» (рис. 4) в правом верхнем углу страницы чата. 

Рисунок 4 – кнопка «выйти»

# **ПРИЛОЖЕНИЕ 4 Руководство системного администратора**

|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Утверждаю

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_\_"\_\_\_\_\_\_\_\_\_2024г.

**Факультет «Радиотехнический»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Сетевые технологии в АСОИУ»

Руководство системного администратора

Вариант 43

Студенты группы РТ5-61Б:

Голумин Н.Д.

Надыршина А.А.

Крайников И.А.

2024г.

1. **Системные требования**

Для работы клиента необходим Yandex Browser или Google Chrome.

* 1. **Требования к ОС**
* Windows

Windows 7, Windows 8, Windows 8.1, Windows 10 или более поздней версии.

Примечание: также поддерживаются следующие серверные ОС: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 или Windows Server 2016.

* MacOS

macOS High Sierra 10.13 или более поздней версии

* Linux

Ubuntu 18.04 (64-разрядная версия) или более поздней версии, Debian 10 или более поздней версии, openSUSE 15.2 или более поздней версии, Fedora Linux 32 или более поздней версии.

**Требования к оборудованию**

Центральный процессор: с частотой от 1 ГГц

Оперативная память: 8 Гб и выше

Пространство на жестком диске: 1 Гб

Платформа: Microsoft Windows, Mac OS, Linux x32/x64

* 1. **Требования к ПО**

1. Docker desktop версии 4.17.1;
2. Golang версии 1.21.0 и выше;
3. Django версии 4.2 и выше;
4. Node.js версии 1.20 и выше;
5. Kafka версии 2.12 и выше.
6. **Порядок развёртывания серверов:**
   1. **Сервис канального уровня**
7. Выполнить команду python -m venv venv;
8. Выполнить команду source venv/bin/activate;
9. Выполнить команду pip install -r requirements.txt;
10. Выполнить команду python manage.py runserver.

После этого на порте 8000 будет доступен сервис канального уровня.

* 1. **Сервис транспортного уровня**

1. Выполнить команду docker compose up -d –build;
2. Выполнить команду go run cmd/main.go.

После этого на порте 8080 будет доступен сервис транспортного уровня.

* 1. **Сервис фронтенда прикладного уровня**

1. Выполнить команду npm i;
2. Выполнить команду npm run dev.

После этого на порте 5173 будет доступен сервис фронтенда прикладного уровня.

* 1. **Сервис бекенда прикладного уровня**
  2. Выполнить команду npm i;
  3. Выполнить команду npm run dev.

После этого на порте 8001 будет доступен сервис фронтенда.