



**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

_____ 2025г.

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Сетевые технологии в АСОИУ»

Техническое задание

Вариант 33

Студенты группы ИУ5-61Б:

Ларкин Б.В.

Цыпышев Т.А.

Серов С.С.

2025г.

1. Наименование:

Разработать распределенную систему обмена сообщениями в реальном времени.

2. Основание для разработки:

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

3. Исполнители:

Ларкин Б.В. (прикладной уровень) – группа ИУ5-61Б

Цыпышев Т.А. (транспортный уровень) – группа ИУ5-61Б

Серов С.С. (канальный уровень) – группа ИУ5-61Б

4. Цель разработки:

Разработать распределенную систему для обмена сообщениями в реальном времени, состоящую из трех уровней: прикладной, транспортный и канальный, каждый из которых реализуется как отдельный веб-сервис.

5. Функциональные требования

5.1. Прикладной уровень:

5.1.1. Страница приложения представляет собой чат, где пользователи могут отправлять и получать текстовые сообщения. Каждое сообщение будет отображать информацию об отправителе и времени отправки.

5.1.1.1. При подключении к чату пользователю следует ввести свое имя и выбрать сервер (Земля или Марс), к которому он хочет подключиться. Это имя будет передаваться вместе с каждым новым текстовым сообщением;

5.1.1.2. После успешной авторизации пользователи с Земли смогут отправлять текстовые сообщения на Марс через установленное WebSocket-соединение и напрямую на Землю. Для ввода текста предусмотрено поле ввода. Для отправки сообщения предусмотрена кнопка "Отправить";

5.1.1.3. У пользователей с Марса отсутствует возможность отправки сообщений. При попытке отправки пользователь увидит ошибку от сервера.

5.1.1.4. Если при получении сообщения возникнет ошибка, то это сообщение не будет отображаться, а пользователи увидят вместо него уведомление о сбое;

5.1.1.5. При нажатии кнопки "Выйти" все данные (чат и логин) очищаются, а активное WebSocket-соединение закрывается;

5.1.1.6. Дизайн приложения должен соответствовать сайту stepik.org и быть адаптирован для мобильных устройств;

5.1.2. WebSocket-сервер

5.1.2.1. WebSocket-сервер управляет списком пользователей, поддерживающих активные соединения. Он хранит информацию об имени пользователя и местонахождении для каждого активного соединения;

5.1.2.2. Сервер предоставляет возможность открывать и закрывать WebSocket-соединения. Полученные сообщения от клиентов с Земли сервер рассылает всем подключенным пользователям, за исключением тех, кто находится на Земле или чье имя совпадает с именем отправителя;

5.1.3. Реализация HTTP-метода `Receive` для получения сообщения с транспортного уровня:

5.1.3.1. Каждое сообщение, передаваемое по HTTP, содержит информацию о том, кто является отправителем,

когда было отправлено сообщение, есть ли ошибка в передаче, а также сам текст сообщения;

5.1.3.2. После получения сообщения по HTTP, сервер передает его всем подключенным WebSocket-клиентам через широковещательную рассылку, кроме того клиента, который отправил это сообщение или клиентов, которые находятся на Земле;

5.2. Транспортный уровень:

5.2.1. Реализация HTTP-метода Send, который отвечает за разделение текстового сообщения на несколько частей (сегментов):

5.2.1.1. Каждое сообщение будет разделяться на сегменты по 140 байт. Каждый сегмент будет последовательно передаваться через метод Code на канальный уровень;

5.2.1.2. Каждый сегмент будет содержать информацию о времени отправки сообщения (чтобы все части можно было собрать в правильном порядке), общую длину сообщения, порядковый номер сегмента и сам текст сообщения;

5.2.2. Реализация HTTP-метода Transfer, который собирает полученные сегменты в полное сообщение прикладного уровня:

5.2.2.1. Полученные сегменты будут храниться в очереди, и раз в 1 секунду сервер будет собирать их в одно сообщение и передавать на прикладной уровень;

5.2.2.2. Если некоторые сегменты не были получены или переданы с ошибкой, сервер будет отправлять полное сообщение с признаком ошибки;

5.3. Канальный уровень:

5.3.1. Сервис канала связи имитирует возможные потери данных. Он добавляет случайные ошибки в передаваемые сообщения:

5.3.1.1. Сервис должен вносить ошибку с вероятностью 10% в один случайный бит каждого сформированного кадра;

5.3.1.2. Сервис должен терять передаваемый кадр с вероятностью 2%;

5.3.2. Реализация HTTP-метода Code, который выполняет кодирование и декодирование полученных от транспортного уровня сегментов:

5.3.2.1. Каждый сегмент, полученный от транспортного уровня, кодируется с использованием циклического [7,4]-кода, что позволяет обнаруживать и исправлять ошибки при передаче данных;

5.3.2.2. После того как ошибка внесена в кадр, он будет декодирован, исправлен и передан дальше на транспортный уровень в виде конкретного сегмента;

6. Требования к составу технических средств:

6.1. Прикладной уровень:

6.1.1. Серверная часть

6.1.1.1. ПК с ОС Linux(5.2 и выше)

6.1.1.2. Node.js (1.20 и выше)

6.1.2. Клиентская часть

6.1.2.1. Телефон с ОС Android(10 и выше)

6.1.2.2. Веб-браузер: Chrome(100 и выше)

6.2. Транспортный уровень:

6.2.1.1. ПК с ОС Linux(5.2 и выше)

6.2.1.2. GoLang (4.2 и выше)

6.2.1.3. Kafka (2.12 и выше)

6.3. Канальный уровень:

6.3.1.1. ПК с ОС MacOS(6.4 и выше)

6.3.1.2. GoLang (4.2 и выше)

7. Этапы разработки:

- 7.1. Выбрать тему-вариант, определить команду и сформировать ТЗ – 3 неделя;
- 7.2. Разработать макет figma, три диаграммы последовательности и описать HTTP-методы в swagger – 8 неделя;
- 7.3. Разработать и отладить приложение, подготовить полный комплект документов (ТЗ, РПЗ, ПМИ, РП, РСА) – 12 неделя;
- 7.4. Исправить замечания, защитить проект – 14 неделя.

8. 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

9. Порядок приемки работы:

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

Работа защищается перед комиссией преподавателей кафедры.

10. Дополнительные условия:

Данное Техническое Задание может дополняться и изменяться в установленном порядке.