

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа «Программная инженерия»

**СОГЛАСОВАНО**  
Доцент департамента образовательной  
программы «Программная инженерия»,  
кандидат педагогических наук

\_\_\_\_\_ С. А. Виденин  
«\_\_» \_\_\_\_\_ 2024 г.

**УТВЕРЖДАЮ**  
Академический руководитель  
образовательной программы  
«Программная инженерия»,  
старший преподаватель департамента  
программной инженерии

\_\_\_\_\_ Н. А. Павлочев  
«\_\_» \_\_\_\_\_ 2024 г.

**ЧАТ-РУЛЕТКА ДЛЯ РАЗВИТИЯ КОММУНИКАТИВНЫХ НАВЫКОВ НА  
ИНОСТРАННОМ ЯЗЫКЕ**

**Текст программы**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.03-01 12 01-1-ЛУ**

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Исполнители  
студент группы БПИ225  
\_\_\_\_\_ / Д. А. Мирошниченко /  
«\_\_» \_\_\_\_\_ 2024 г.

студент группы БПИ225  
\_\_\_\_\_ / Л. А. Светличный /  
«\_\_» \_\_\_\_\_ 2024 г.

**Москва 2024**

УТВЕРЖДЕН  
RU.17701729.05.03-01 12 01-1-ЛУ

ЧАТ-РУЛЕТКА ДЛЯ РАЗВИТИЯ КОММУНИКАТИВНЫХ НАВЫКОВ НА  
ИНОСТРАННОМ ЯЗЫКЕ

Текст программы

RU.17701729.05.03-01 12 01-1

Листов 12

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

Москва 2024

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**СОДЕРЖАНИЕ**

<b>СОДЕРЖАНИЕ .....</b>	<b>2</b>
<b>1. ТЕКСТ ПРОГРАММЫ .....</b>	<b>3</b>
1.1. Бэкенд .....	3
1.2. Фронтенд .....	3
<b>2. ОПИСАНИЕ ПРОГРАММЫ .....</b>	<b>4</b>
2.1. Описание бэкенда .....	4
2.2. Описание фронтенда .....	9
<b>СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....</b>	<b>11</b>
<b>ПРИЛОЖЕНИЕЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ .....</b>	<b>12</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 1. ТЕКСТ ПРОГРАММЫ

### 1.1. Бэкенд

Исходный код бэкенд-части программы на языке C# представлен на удалённом репозитории Github по ссылке <https://github.com/Platinaa777/ChatRoulette.Backend>.

### 1.2. Фронтенд

Исходный код фронтенд-части программы на языке JavaScript (с использованием HTML и CSS) представлен на удалённом репозитории Github по ссылке <https://github.com/Platinaa777/ChatRoulette.Frontend>.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 2. ОПИСАНИЕ ПРОГРАММЫ

### 2.1. Описание бэкенда

Серверная часть выполнена на языке C# в рамках фреймворка ASP.NET Core.

В репозитории с исходным кодом программы содержатся следующие директории:

- Директория ApiGateways содержит логику по маршрутизации трафика:
  - В директории Ocelot.Gateway содержится реализация .NET API-шлюза. Для данного проекта было принято решение использовать библиотеку Ocelot, поскольку весь проект состоит из микросервисов;
  - В директории endpoints содержатся JSON-файлы с описанием определённых точек доступа, которые предназначены для простоты ориентации взаимодействия серверной и клиентских частей приложения;
  - В директории Common содержатся скрипты с развёртыванием приложения в режиме production, а именно содержатся docker-compose файлы с ElasticSearch, RabbitMQ;
- Директория Infrastructure представляет собой проекты-библиотеки, которые используются в микросервисах:
  - SwaggerConfigurations содержит зависимости для подключения инструментария Swagger;
  - MessageBroker содержит общие классы-контракты, используемые при передаче сообщений через брокер RabbitMQ, а также класс-клиент для отправки информации в брокер;
- Директория Services содержит программные микросервисы, составляющие архитектуру приложения:
  - AuthService, отвечающий за авторизацию в приложении:
    - В директории Api содержатся:
      - Controllers (папка с контроллером для доступа к функциональности сервиса аутентификации/авторизации);
      - Consumers – слушатели определённых событий, которые происходят в системе, для дальнейшего применения этих изменений в данном сервисе;
      - Services – регистрация зависимостей для запуска проекта;
      - Mappers – утилиты по преобразованию запросов от клиента серверу и выдачи ответов клиенту;
    - В директории Application содержатся:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- Assembly – класс, являющийся сборкой проекта;
- Behaviours – папка с классами, обрабатывающими определённые действия для дальнейшей передачи их в бизнес-логику (например, валидация, логгирование, и т.д.);
- Cache – интерфейс для взаимодействия с хранилищем Redis;
- Commands – команды, изменяющие состояние системы;
- JwtConfigs – класс с полями для генерации jwt-токенов; Models – модели, передаваемые на уровень Api;
- Queries – запросы, не способные изменять состояние системы и предназначенные только для получения информации;
- Security – интерфейсы для взаимодействия с хешированием, jwt-менеджером и паролями.
- В директории DataContext содержатся:
  - Configuration – структура ORM (EntityFrameworkCore), которая перерабатывает объекты в сущности базы данных и в обратную сторону;
  - Database – поля, отображающие таблицы в базе данных;
- В директории Domain содержатся:
  - Errors – ошибки, которые могут возникать в агрегатах приложения;
  - Models – агрегаты с поведением валидации, взаимодействия с другими агрегатами, определением данных контракта для общения с ними;
- В директории HttpModels содержатся:
  - Requests – контракты запросов от пользователя;
  - Responses – контракты ответов для пользователя;
- В директории Infrastructure содержатся:
  - Cache – реализация взаимодействия с кэшем приложения;
  - JwtCreator – логика для генерации токенов доступа;
  - Repos – реализация репозитория из уровня Domain;
- Директория Migrations содержит миграции для базы данных;
- Chat, отвечающий за видеосвязь и передачу текстовых сообщений между пользователями;
- В директории Api содержатся:
  - Controllers – папка с контроллером для доступа к

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

функциональности чата;

- Infrastructure – регистрация зависимостей для запуска проекта;
- Mappers – преобразования запросов от клиента серверу и выдачи ответов клиенту;
- В WebSockets содержится хаб для принятия WebSocket-запросов;
- В директории Application содержатся:
  - Commands – команды системе, способные изменять её состояние;
  - Queries – запросы системе, не способные изменять её состояние и предназначенные исключительно для получения информации;
- В директории Domain содержатся:
  - Entities – сущности пользователей, находящихся в чат-рулетке и комнатах;
  - Repositories – интерфейсы для внешнего взаимодействия с сущностями из Entities;
  - Shared – вспомогательные классы для реализации подхода DDD;
- В директории HttpModels содержатся:
  - Requests – контракты запросов от пользователя;
  - Responses – контракты ответов для пользователя;
- В директории Infrastructure содержится Repositories – реализация репозитория из Domain;
- EmailingService отвечает за отправку e-mail сообщений, в частности – за отправку пользователям писем подтверждения при прохождении регистрации;
- В директории Api содержатся:
  - Controllers – контроллер для доступа к функциональности почтового сервиса;
  - Consumers – слушатели определённых событий, происходящих в системе, для применения этих изменений в данном сервисе;
  - Infrastructure – регистрация зависимостей для запуска проекта;
  - Configurations – конфигурационный файл для SMTP-клиента;
  - EmailTemplates – шаблон для отправки почтового сообщения;
- В директории HttpModels содержатся:
  - Requests – контракты запросов от пользователя;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- Responses – контракты ответов для пользователя;
- ProfileService отвечает за сбор аналитики по аккаунтам пользователей, хранению и работе данных о пользовательских аккаунтах, и содержит систему «друзей»:
- В директории Api содержатся:
  - Controllers – контроллер для доступа к функциональности почтового сервиса;
  - Consumers – слушатели определённых событий, происходящих в системе, для применения этих изменений в данном сервисе;
  - Extensions – регистрация зависимостей для запуска проекта;
- В директории Application содержатся:
  - Assembly – класс, являющийся сборкой проекта;
  - Behaviours – папка с классами, обрабатывающими определённые действия для дальнейшей передачи их в бизнес-логику (например, валидация, логгирование, и т.д.);
  - Commands – команды системе, способные изменять её состояние;
  - Queries – запросы, не способные изменять состояние системы и предназначенные только для получения информации;
  - Models – модели, передаваемые на уровень api;
- В директории Domain содержатся:
  - Models – агрегаты с поведением валидации, взаимодействия с другими агрегатами, определением данных контракта для общения с ними;
  - Shared – вспомогательные классы для реализации подхода DDD;
- В директории HttpModels содержатся:
  - Requests – контракты запросов от пользователя;
  - Responses – контракты ответов для пользователя;
- В директории Infrastructure содержатся:
  - Configuration – класс, представляющий строку подключения к базе данных;
  - OutboxPattern – сущность для реализации паттерна Outbox;
  - Repos – классы для взаимодействия с внешними системами;
- В директории Migrations содержатся кастомные миграции для базы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



данных, а также консольное приложение для запуска этих миграций;

- AdminService для обработки жалоб на пользователей, выделения блокировок пользователям, и просмотра отзывов о работе приложения;

- В директории Api содержатся:

- Controllers (папка с контроллером для доступа к функциональности сервиса системы жалоб для пользователя со статусом администратор);

- Consumers – слушатели определённых событий, которые происходят в системе, для дальнейшего применения этих изменений в данном сервисе;

- Services – регистрация зависимостей для запуска проекта;

- Mappers – утилиты по преобразованию запросов от клиента серверу и выдачи ответов клиенту;

- В директории Application содержатся:

- Assembly – класс, являющийся сборкой проекта;

- Behaviours – папка с классами, обрабатывающими определённые действия для дальнейшей передачи их в бизнес-логику (например, валидация, логгирование, и т.д.);

- Commands – команды системе, способные изменять её состояние;

- Queries – запросы, не способные изменять состояние системы и предназначенные только для получения информации;

- В директории DataContext содержатся:

- Configuration – структура ORM (EntityFrameworkCore), которая перерабатывает объекты в сущности базы данных и в обратную сторону;

- Database – поля, отображающие таблицу в базе данных;

- Migrations – миграции для базы данных;

- В директории Domain содержатся:

- Errors – ошибки, которые могут возникать в агрегатах приложения;

- Models – агрегаты с поведением валидации, взаимодействия с другими агрегатами, определением данных контракта для общения с ними;

- В директории HttpModels содержатся:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- Requests – контракты запросов от пользователя;
- Responses – контракты ответов для пользователя;
- В директории Infrastructure содержатся:
  - Repos – реализация репозитория из уровня Domain;

## 2.2. Описание фронтенда

Программа выполнена на языке программирования JavaScript с использованием библиотеки React. Также используются язык разметки HTML и язык декорирования CSS.

В файле App.js представлена навигация по приложению с использованием Browser Router.

В директории static статических данных представлены пути к страницам приложения (файл Paths.js) и данные о страницах приложения (файл SidebarPages.js), которые используются в боковой панели приложения.

В директории navbar представлены компоненты и логика по постраничной навигации приложения. Помимо навигационной панели Navbar.jsx присутствуют боковая панель Sidebar.jsx и меню опций авторизации для пользователя AuthMenu.jsx, откуда можно перейти на страницы регистрации и входа в аккаунт, если пользователь ещё не авторизован, и выйти из аккаунта, если пользователь авторизован.

В директории pages содержатся директории, отвечающие за различные страницы приложения. Содержащиеся директории:

- Директория auth: в ней представлена логика по авторизации пользователя. Здесь содержатся компоненты страниц SignIn.jsx для входа и SignUp.jsx для создания аккаунта, api.js для получения токенов доступа, и оболочка UserSession.js для хранения и работы с данными о пользователе внутри приложения;
- Директория chat: здесь представлены компоненты для отображения видеочата и текстового чата, а также утилиты для работы с технологией WebRTC;
- Директория friends: в ней представлены компоненты для отображения списка друзей и списка недавних пользователей, а также утилиты для получения информации о них от сервера и отправки запросов в друзья и жалоб на пользователей;
- Директория menu: в ней представлен компонент Main.jsx для отображения главной страницы приложения;
- Директория profile: здесь присутствуют компоненты для отображения профильной страницы приложения и логика по изменению данных о пользователе и обновлении соответствующих данных на сервере;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- Директория report: компонент ReportProblem.jsx страницы для отправки отзыва о работе приложения;

- Директория admin: компонент Review.jsx для отображения страницы просмотра отзывов по работе приложения и жалоб пользователей, которая доступна только для пользователя со статусом admin. Внутри также представлена логика по выдаче пользователям блокировок.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

- 1) ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]