

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

(наименование факультета)

Кафедра «ПОВТиАС»

(наименование кафедры)

|  |  |  |
| --- | --- | --- |
|  | | |
| Зав. кафедрой | | «ПОВТиАС» |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | В.В. Долгов |
| (подпись) | | (И.О.Ф.) |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | |

**ОТЧЕТ**

по преддипломной практике

вид практики

в научно-образовательном производственном центре «Визуализация, анализ и защита информации» (НОПЦ «ВАиЗИ»)

наименование базы практики

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Л. Речков

подпись, датаИ.О.Ф.

Обозначение отчета ПП.96.0000.000 Группа ВПР-42

Направление 09.03.04 Программная инженерия

коднаименование направления подготовки

Профиль Программное обеспечение вычислительной техники и автоматизированных систем

Руководитель практики:

от предприятия директор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долгов В.В.

должность подпись, дата имя, отчество, фамилия

Б.П.

от кафедры доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Криворучко

должность подпись, дата имя, отчество, фамилия

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата подпись преподавателя

Ростов-на-Дону

2020 г.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

(наименование факультета)

Кафедра «ПОВТиАС»

(наименование кафедры)

**ЗАДАНИЕ**

по преддипломной практике

вид практики

в научно-образовательном производственном центре «Визуализация, анализ и защита информации» (НОПЦ «ВАиЗИ»)

наименование базы практики

в период с « 20 » апреля 2020 г. по « 16 » мая 2020 г.

Обучающийся Речков Андрей Леонидович

Обозначение отчета ПП.96.0000.000 Группа ВПР-42

Срок представления отчета на кафедру « » 2020 г.

Содержание индивидуального задания:

|  |
| --- |
| 1. Теоретическая часть. В данном разделе предоставить описание инструментов, выбранных  для решения поставленной цели – создание веб-приложения «Chat messenger».  2. Алгоритмическое конструирование. В данном разделе предоставить описание  алгоритмов работы программного средства.  3. Программное конструирование и тестирование приложения. В данном разделе  предоставить описание файлов, содержащих реализацию представленных алгоритмов на  языке Python и TypeScript. Также предоставить результаты тестирования приложения. |

|  |  |  |
| --- | --- | --- |
| Руководитель практики от  кафедры | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | А.В. Криворучко  имя, отчество, фамилия |
|  |  |  |
| Задание принял к исполнению | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | .  имя, отчество, фамилия |



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

(наименование факультета)

Кафедра «ПОВТиАС»

(наименование кафедры)

|  |  |  |
| --- | --- | --- |
|  | | |
| Зав. кафедрой | | «ПОВТиАС» |
| \_\_\_\_\_\_\_\_\_\_\_\_ | | В.В. Долгов |
| (подпись) | | (И.О.Ф.) |
| «\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | |

**Рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| **№** | **Мероприятие** | **Срок выполнения** |
| 1 | Прохождение вводного и первичного инструктажа по охране труда на рабочем месте, и инструктажа по пожарной безопасности на объекте | 20.04.2020 |
| 2 | Аналитический обзор поставленной задачи | 21.04.2020 – 22.04.2020 |
| 3 | Алгоритмическое конструирование | 22.04.2020 – 01.05.2020 |
| 4 | Программное конструирование | 02.05.2020 – 13.05.2020 |
| 5 | Тестирование приложения и анализ расчетов | 13.05.2020 – 14.05.2020 |
| 6 | Подготовка итогового отчета | 14.05.2020 – 15.05.2020 |
| 7 | Защита итогового отчета | 16.05.2020 |

Руководитель практики:

от предприятия директор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долгов В.В.

должность подпись, дата имя, отчество, фамилия

Б.П.

от кафедры \_\_\_ доцент\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Криворучко  должность подпись, дата имя, отчество, фамилия

Ростов-на-Дону

2020 г.

ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИКИ

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Место работы | Выполняемые работы | Оценка руководителя |
| 20.04.2020 | НОПЦ «ВАиЗИ» | Знакомство с предприятием, прохождение вводного инструктажа. |  |
| 20.04.2020 | НОПЦ «ВАиЗИ» | Ознакомление с территорией предприятия, прохождение первичного инструктажа по ТБ, ПБ |  |
| 21.04.2020 – 22.04.2020 | НОПЦ «ВАиЗИ» | Аналитический обзор поставленной задачи |  |
| 22.04.2020 – 01.05.2020 | НОПЦ «ВАиЗИ» | Алгоритмическое конструирование |  |
| 02.05.2020 – 13.05.2020 | НОПЦ «ВАиЗИ» | Программное конструирование |  |
| 13.05.2020 – 14.05.2020 | НОПЦ «ВАиЗИ» | Тестирование приложения и анализ расчетов |  |
| 14.05.2020 – 15.05.2020 | НОПЦ «ВАиЗИ» | Подготовка итогового отчета |  |
| 16.05.2020 | НОПЦ «ВАиЗИ» | Защита итогового отчета |  |

Руководитель практики:

от предприятия директор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долгов В.В.

должность подпись, дата имя, отчество, фамилия

Б.П.

ОТЗЫВ - ХАРАКТЕРИСТИКА

Обучающийся Речков Андрей Леонидович

фамилия, имя, отчество

4 курса группы ВПР-42 кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Вид практики \_\_преддипломная

Наименование места практики научно-образовательный производственный центр

«Визуализация, анализ и защита информации» (НОПЦ «ВАиЗИ»)

наименование предприятия, структурного подразделения

Обучающийся выполнил задания программы практики

Предоставил теоретическую часть, описывающую инструменты, выбранные инструменты, выбранные для решения поставленной цели – создания приложения «Chat messenger»

Создал алгоритмическое конструирование, в котором описал работу алгоритмов программного средства

Разработал программное средство в соответствии с требованиями, в полном

объеме по всем разделам

Дополнительно ознакомился/изучил

Алгоритм авторизации, используя фреймворк Django

Работу с библиотекой «RxJS», используя фреймворк Angular

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

Заслуживает оценки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | Руководитель практики  от предприятия  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_ г.  Б.П. |

**СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ |  | 8 |
| 1 Теоретический обзор |  | 9 |
| 1.1 Общие сведения о фреймворках Django и Angular |  | 9 |
| 1.2 Язык программирования Python и Typecript |  | 10 |
| 1.3 Среда разработки PyCharm и WebStorm |  | 10 |
| 1.4 Выводы по главе |  | 11 |
| 2 Алгоритмическое конструирование |  | 13 |
| 2.1 Общая схема работы веб-приложения «Chat messenger» |  | 13 |
| 2.2 Разработка алгоритмов для веб-приложения  «Chat messenger» |  | 14 |
| 2.2.1 Создание части backend на Django |  | 14 |
| 2.2.2 Создание части frontend на Angular |  | 15 |
| 2.3 Выводы по главе |  | 15 |
| 3 Программное конструирование |  | 17 |
| 3.1 Описание работы части backend на Django |  | 17 |
| 3.2 Описание работы части frontend на Angular |  | 18 |
| 3.3 Составление UML-диаграммы приложения |  | 19 |
| 3.4 Выводы по главе |  | 19 |
| 4 Тестирование программного средства |  | 21 |
| 4.1 Тестирование веб-приложения «Chat messenger» |  | 21 |

|  |  |
| --- | --- |
| 4.2 Выводы по главе | 23 |
| ЗАКЛЮЧЕНИЕ | 24 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 25 |
| ПРИЛОЖЕНИЕ А: Техническое задание на программное средство | 26 |
| ПРИЛОЖЕНИЕ Б: Листинг программы | 31 |

**ВВЕДЕНИЕ**

Веб-разработка – это процесс создания веб-сайта или веб-приложения. Основными этапами процесса являются веб-дизайн, верстка страниц, программирование на стороне клиента и сервера, а также конфигурация веб-сервиса. Разработка на стороне клиента и сервера в наше время зачастую осуществляется путём использования различных фреймворков (программных обеспечений, облегчающих разработку и объединение разных компонентов большого программного проекта), которые обеспечивают большую помочь разработчику в написании кода.

Сегодня наиболее распространенным фреймворками для веб-приложения является Django на стороне сервера и Angular на стороне клиента. Используя данные программные обеспечения, мы руководствуемся в ходе выполнения веб-приложения «Chat messenger».

Таким образом можно сформулировать цель данной преддипломной практики: разработка веб-приложения для общения между пользователями, хранения пользовательских данных. Допуск в веб-приложение будет осуществляться через авторизацию. Более подробно функциональные требования описаны в Приложении А.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Разработать дизайн веб-приложения.

2. Научиться проектировать и создавать программы на языке Python и TypeScript.

3. Реализовать пользовательский интерфейс.

4. Создать соответствующие программные компоненты для работы веб-приложения «Chat messanger».

**1 Теоретический раздел**

В данном разделе приведены сведения о решаемой задаче и инструментах, с помощью которых будет разработано веб-приложение.

**1.1 Общие сведения о фреймворках Django и Angular**

Django – свободный фреймворк для веб-приложений на языке Python,

использующий шаблон проектирования MVC (схема разделения данных приложения, пользовательского интерфейса и управляющей логики на при отдельных компонента: модель, представление, контроллер). Проект поддерживает организацией Django Software Foundation [1].

Для работы с базой данных Django использует собственный ORM (технология программирования, которая связывает базы данных с концепцией объектно-ориентированных языков программирования), в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

Angular – это открытая и свободная платформа для разработки веб-приложений, написанных на языке TypeScript, разрабатываемая командой из компании Google, а также сообществом разработчиков из различных компаний. Angular – это полностью переписанный фреймворк от той же команды, которая написала AngularJS.

Одной из ключевых особенностей Angular является то, что он использует в качестве языка программирования TypeScript, который требует строгой типизации [2].

**1.2 Язык программирования Python и TypeScript**

Python – это интерпретируемый объектно-ориентированный язык программирования высокого уровня с динамической семантикой. Его встроенные структуры данных высокого уровня в сочетании с динамической типизацией и динамическим связыванием делают его очень привлекательным для быстрой разработки приложения, а также для использования в качестве скриптового или связующего языка соединения существующих компонентов.

Интерпретатор Python может быть легко дополнен функционалом за счёт новых функций и типов данных, реализованных в C и C ++ (или любом другом языке, вызываемом из C). Существует достаточно много разновидностей версий Python, но наиболее востребованные из них это версия 2.7 и 3.6.10 (последняя вышедшая версия). Код написанный на разных версиях может быть очень похож, но интерпретатор будет корректно работать лишь с кодом, одной версии [3].

TypeScript – это язык программирования с открытым исходным кодом, разработанный и поддерживаемый Microsoft. Он является обратно совместимый с JavaScript (язык программирования, поддерживаемый объектно-ориентированное программирования) и компилируется в последний. Фактически, после компиляции программу на TypeScript можно выполнять в любом современном браузере или использовать совместно с серверной платформой Node.js (программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего названия) [4].

Код экспериментального компилятора, транслирующего TypeScript в JavaScript. TypeScript отличается от JavaScript возможностью явного статического назначения типов, поддержкой использования полноценных классов, а также поддержкой подключения модулей, что призвано повысить скорость разработки, облегчить читаемость, рефакторинг.

**1.3 Среда разработки PyCharm и WebStorm**

PyCharm – интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA. Также это кроссплатформенная среда разработки, которая совместима с Windows, MacOS, Linux [5].

Возможности:

1. Статический анализ кода, подсветка синтаксиса и ошибок.
2. Навигация по проекту и исходному коду.
3. Встроенный отладчик для Python.
4. Инструменты для веб-разработки с Django и т.д.

WebStorm – интегрированная среда разработки на JavaScript, CSS, и HTML. Он обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями (программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости переключаться между ними) [5].

Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами. Зачастую данную среду разработки используют для написания кода не только JavaScript, CSS и HTML, но и для TypeScript, SASS (метоязык на основе CSS, предназначенный для увеличения уровня абстракций CSS кода и упрощения файлов каскадных таблиц). Существует два синтаксиса SASS и SCSS. SASS – не использует фигурных скобок, SCSS – использует, как и CSS. В данной работе мы будем использовать SCSS.

**1.4 Выводы по главе**

В данном разделе приведены теоретические сведения, на которые мы можем опираться при написании кода.

При разработке веб-приложения были использованы такие фреймворки, как Django и Angular. Для работы с базой данных Django использует собственную технологию программирования, которая связывает базы данных с концепцией объектно-ориентированных языков программирования. Для разбиения одного компонента на несколько отдельных используется шаблон проектирования MVC, в качестве языка программирования используется Python.

Python – это высокоуровневый язык программирования с динамической семантикой. Существует множество разновидностей версий Python, самыми востребованными из которых являются версии 2.7 и 3.6.10.

Для разработки клиентского интерфейса используется фреймворк Angular, который использует в качестве языка программирования TypeScript, разрабатываемый командой из компании Google.

TypeScript – это язык программирования, разработанный и поддерживаемый Microsoft. Он является обратно совместимым с JavaScript (языком программирования, поддерживаемым с ООП).

Для работы с данными фреймворками чаще всего используются такие среды разработки, как Charm IDEA и WebStorm IDEA.

После изучения данной информации можно приступать к алгоритмическому конструированию.

**2 Алгоритмическое конструирование веб-приложения «Сhat messenger»**

В этой главе описывается алгоритмическое решение поставленной задачи: общий алгоритм работы программного средства и отдельных его частей и описание функций, реализующих данный алгоритм.

**2.1 Общая схема работы веб-приложения «Chat messenger»**

Под определением «Chat messenger» мы будем понимать самостоятельную компьютерную программу или компонент программного комплекса, предназначенную для представления возможности общения между пользователями.

Веб-приложение предоставляет пользователю возможность общения с другим пользователем, путём регистрирования в веб-приложении и последующим входом.

Пользователю будет предоставлен удобный интерфейс для начала или продолжения общения с зарегистрированными пользователями, путём перехода между канала протокола WebSocket, который предоставляет двунаправленный канал, в котором и будет происходить общение. Данный протокол будет активироваться с помощью Django.

Пользователь может свободно переключаться между созданными каналами, путём клика по интересующему каналу.

Канал будет представлять, как список чатов, данный удобный интерфейс будет реализован путём использования Angular.

**2.2 Разработка алгоритмов для приложения «Chat messenger»**

В работе была поставлена задача реализации веб-приложения «Chat messenger». Тем самым, на основании поставленной задачи была создана программа, состоящая из двух основным частей: серверная часть представлена Django и клиентской части, реализованной на Angular.

**2.2.1 Создание части backend на Django**

Для алгоритмического описания создания части backend на Django веб-приложения на рисунке 2.1 представлена соответствующая схема.

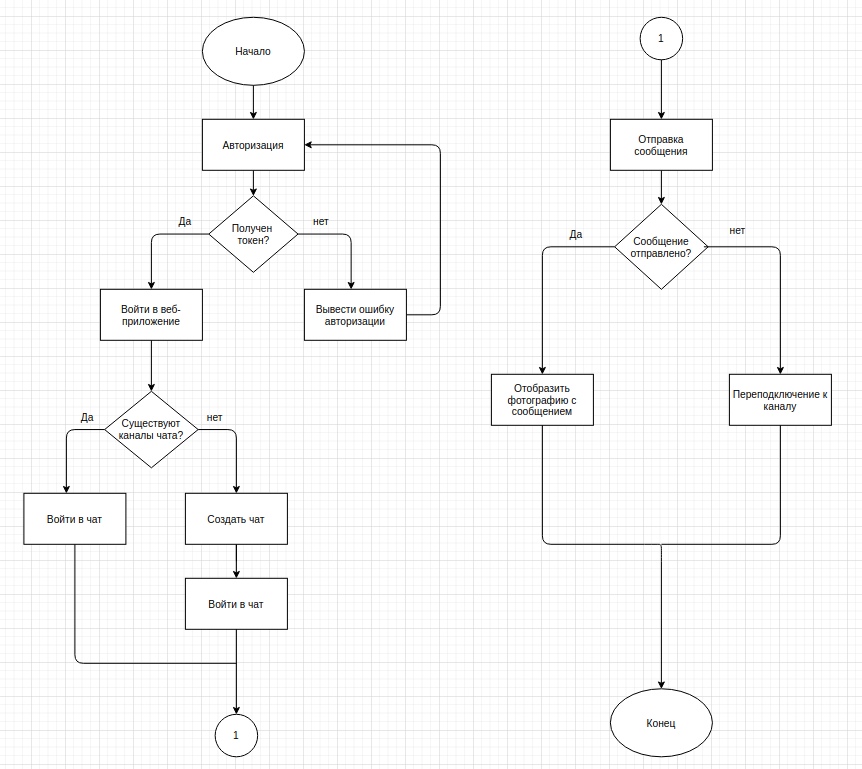


Рисунок 2.1 – схема создания backend на Django

**2.2.2 Создание части frontend на Angular**

Для алгоритмического описания создания части frontend на Angular веб-приложения на рисунке 2.2 представлена соответствующая схема.

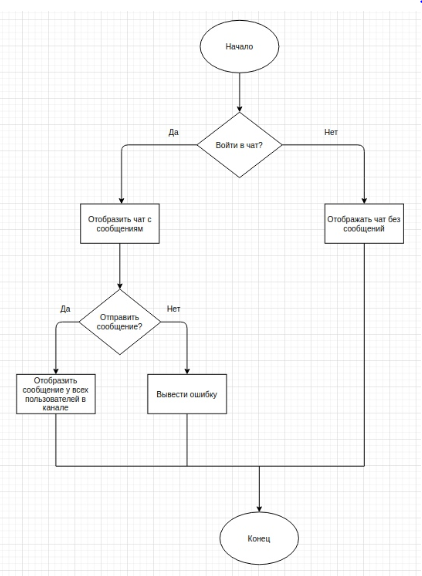


Рисунок 2.2 – схема создания части frontend на Angular

**2.3 Выводы по главе**

В данном разделе были обозначены основные вопросы, касающиеся алгоритмического конструирования программного средства.

Программное средство «Chat messenger» предоставляет возможность общения между пользователями посредством отправки и получения сообщений. Для входа в систему пользователю необходимо авторизоваться либо пройти регистрацию, далее пользователь осуществляет выбор между каналами протокола WebSocket, после чего в выбранном канале может отправлять и получать сообщения.

Если каналов не существует, пользователь может выбрать другого пользователя из всех зарегистровавшихся ранее и перейти в только что созданный канал, в котором будут находиться данные только о них двоих. В случае повторного подключения к каналу (открытия чата), будут загружены их ранее отправленные сообщения.

Разработанные алгоритмы позволят корректно реализовать веб-приложение «Chat messenger», которое будет удовлетворять всем необходимым требованиям, таким как обязательная авторизация для доступа к каналам, засекреченность данных и сохранение ранее отправленных сообщений. Реализация данных алгоритмов рассматривается в следующем разделе.

**3 Программное конструирование приложения «Chat messenger»**

В данной главе представлено описание структуры веб-приложения «Chat messenger»

**3.1 Описание работы части backend на Django**

Основная часть программного кода в приложении относится к классу ChatConsumer. Именно в нём реализуется функционал, который создаёт канал подключения, подключает только тех пользователей, которые находят в канале, загружает сообщения канала.

Точка входа в приложении – функция connect(self). В результате её работы создаётся двусторонний канал протокола WebSocket, методы которого описаны в сводной таблице 3.1.

Таблица 3.1 – Основные методы класса ChatConsumer

|  |  |  |  |
| --- | --- | --- | --- |
| Название метода | Входные параметры | Возвращаемое значение | Краткое описание метода |
| 1 | 2 | 3 | 4 |
| Класс ChatConsumer | | | |
| fetch\_messages | self, data | void | Загружает сообщения находящиеся в канале |
| new\_message | self, data | void | Создаёт новое сообщение |
| connect | self | void | Выполняет подключение к каналу WebSocket |
| disconnect | self, close\_code | void | Отключение от канала  WebSocket |

Окончание таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| receive | self, text\_data | Void | Определяет команду создание или загрузки сообщений |
| send\_message | self, message | Void | Отправка сообщения |
| message\_to\_json | self, message | array | Изменение формата сообщения |
| chat\_message | self, evemt | Void | Отправка на клиент сообщения |

**3.2 Описание работы части frontend на Angular**

Вход в веб-приложение «Chat messenger» осуществляется через клиентское приложение на Angular - Sing класс, отвечающий за проверку введённых данных для пользователей, считывающий данные и отправляющий на сервер захэшированный пароль и логин для получения токена. Если токен получен, происходит переход к классу Chat. Для выбора канала чата, пользователю предлагается список каналов (чатов). После выбора чата, пользователь переход в класс ChatContent, который является дочерним классом класса Chat.

При отсутствии данных для входа в веб-приложения, пользователю предоставляется возможность зарегистрироваться и ввести личные данные, которые будут использоваться в отображении у других пользователей (Имя, Фамилия).

При отображении данных либо отсутствии их, пользователь может отправить сообщение, которое занесётся в redis (временное хранилище данных) и отобразится на стороне клиентов, параллельно будет отправлено сообщение на сервер с помощью Rest Api. Таким образом происходит снижение нагрузки сервера путём загрузки сообщений из хранилища redis.

В случае ошибок отправления данных пользователь будет уведомлён о них и будет сделано переподключение к каналу WebSocket.

**3.3 Составление UML-диаграммы приложения**

В ходе разработке программного средства было создано несколько пользовательских классов. Основные из них изображены в виде UML-диаграммы на рисунке 3.1.

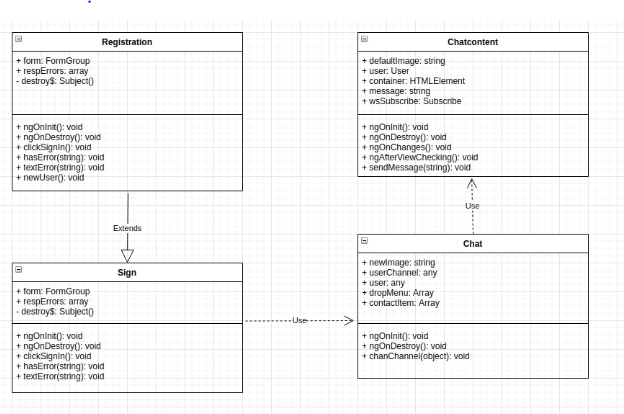


Рисунок 3.1 – Диаграмма классов программного средства

**3.4 Выводы по главе**

В данном разделе было выполнено программное конструирование разрабатываемого веб-приложения. Рассмотрены основные классы, методы, скрипты.

Основным классном части backend является ChatConsumer, который создает канал для подключения только тех пользователей, которые были добавлены при его создании. В данном классе используется протокол WebSocket – коммуникационный протокол, обеспечивающий двухканальную связь по одному TCP – соединению. Основными методами данного класса являются:

1. connect(), выполняющий подключение к каналу;
2. disconnect(), выполняющий отключение от канала;
3. send\_message(), выполняющий отправку сообщения;
4. receive(), определющий команду создания или загрузки сообщений.

Для входа в веб-приложение существует класс Sing, отвечающий за проверку введенных данных пользователя. При успешном входе в систему будет получен токен - последовательность символов для уникальности каждого пользователя.

Листинг файлов представлен в Приложении Б.

**4 Тестирование программного средства**

В данной главе приводятся примеры работы с разработанным программном средством.

**4.1 Тестирование регистрации**

Работа веб-приложения начинается с отображения страницы регистрации. Страница регистрации показана на рисунке 4.1.

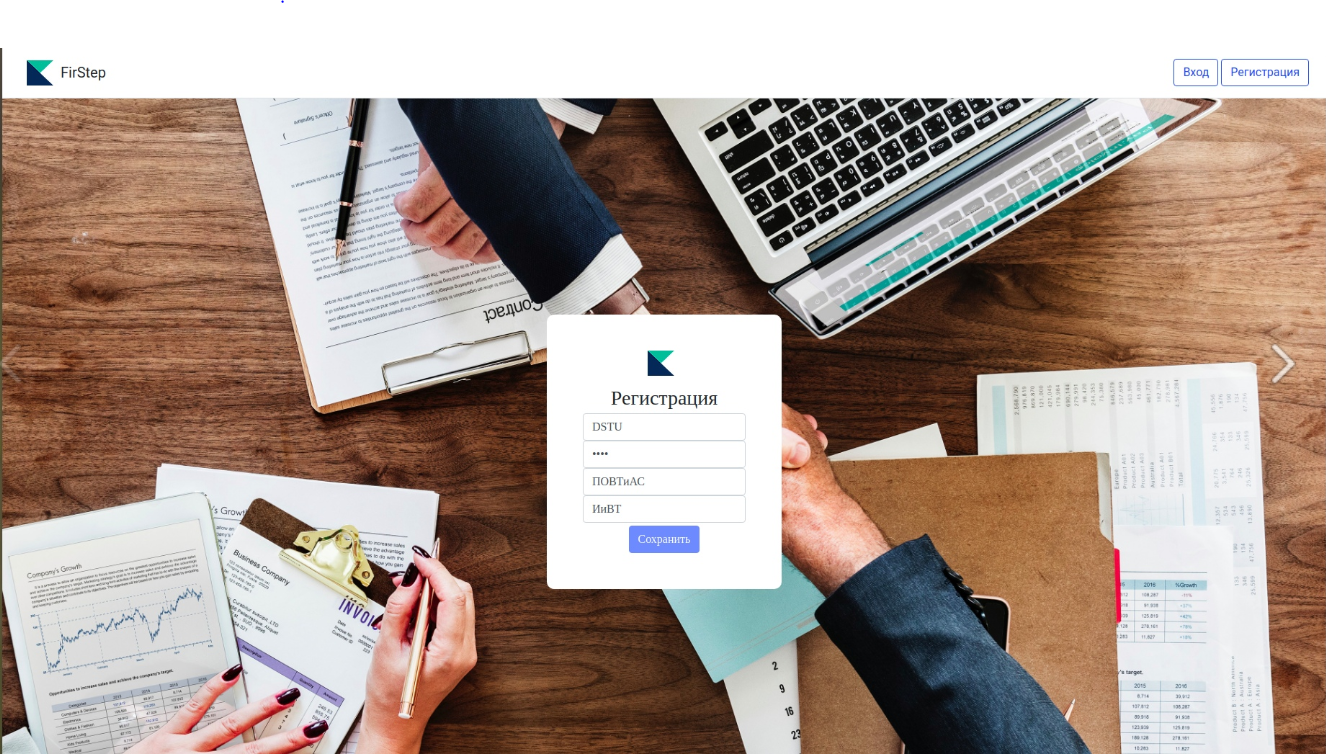


Рисунок 4.1 – Страница регистрации

Осуществим вход в веб-приложение, показанный на рисунке 4.2.



Рисунок 4.2 – Вход в веб-приложение

Теперь необходимо протестировать создание чата с зарегистрированным пользователем (рисунок 4.3).

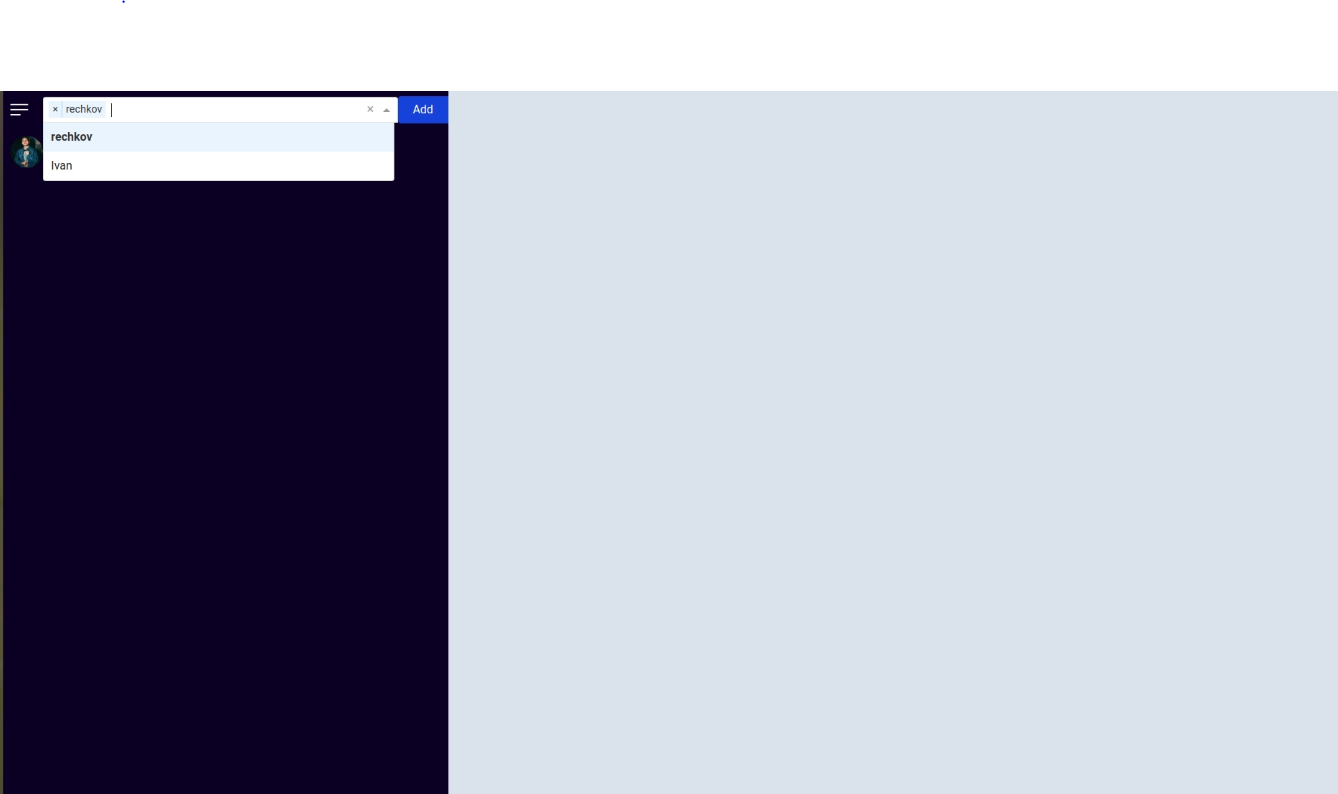


Рисунок 4.3 – Создание чата

Теперь реализуем общение в реальном времени: в левой части изображения отображено окно браузера созданного нами пользователя, в правой части - администратора, созданного ранее. Осуществим общение между ними (рисунок 4.4).

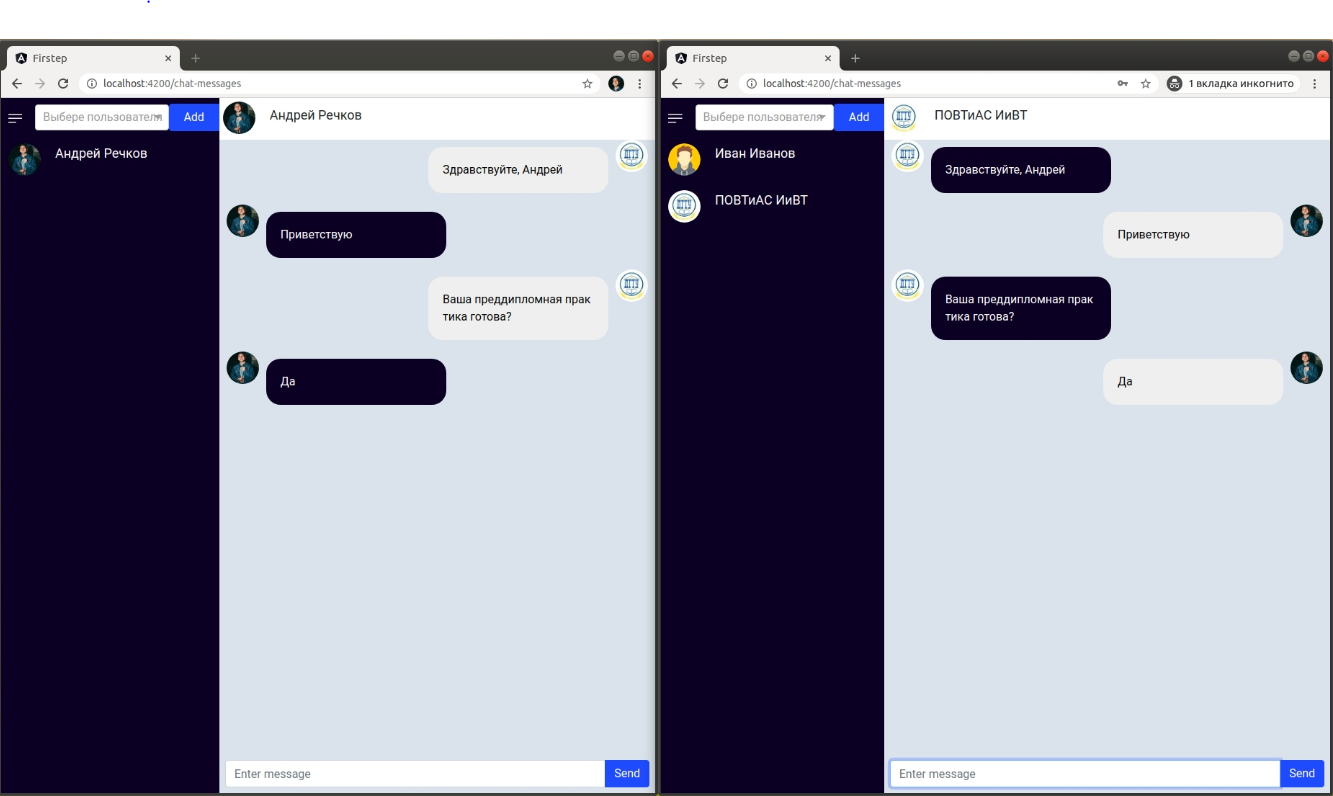


Рисунок 4.4 – Общение между пользователями

Результат работы корректный, так как ожидаемые требования выполнены.

**4.2 Выводы по главе**

В данном разделе была протестирована основная работа интерфейса программного средства, проанализированы результаты работы алгоритмов.

Веб-приложение начинает свою работу с предоставления пользователю возможности войти в систему или пройти регистрацию, в которую необходимо внести обязательные пункты, такие как:

1. имя для входа в систему (Uzername);
2. пароль (Password);
3. имя пользователя (First name);
4. фамилия пользователя (Last name).

В случае, если пользователь зарегистрировался ранее, ему предоставляется два поля ввода: Uzername и Password.

Для тестирования были созданы два пользователя, которые зашли в систему под разными хостами (путем открытия нескольких вкладок браузера) на одном ПК. При отправке сообщений в одной вкладке, сообщение отображается в реальном времени в другой вкладке, что показывает работоспособность данного веб-приложения.

На основании проведенных тестов можно сделать вывод о том, что данное программное средство корректно выполняет все заявленные функции.

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной преддипломной практики было создано веб-приложение «Chat messanger», а также детально описан весь процесс его создания. Таким образом, можно утверждать, что цель практики достигнута: разработано приложение «Chat messenger», позволяющее пользователям общаться друг с другом, а также осуществляющее хранение пользовательских данных.

В ходе работы были выполнены все задачи, поставленные в начале:

1. Разработан дизайн веб-приложения.

2. Изучено проектирование и создание программ на языке Python и TypeScript.

3. Реализован пользовательский интерфейс.

4. Созданы соответствующие программные компоненты для работы веб-приложения «Chat messenger».

В дальнейшем планируется доработка и техническая поддержка программного средства.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Лутц, М. Изучаем Python/ М. Лутц – СПб. : Символ-Плюс, 2019. – 1280 с.
2. Майрен, Х. Выразительный JavaScript/ X. Майрен – СПб. : Питер, 2019. – 480 с.
3. Крокфорд, Д. Как устроен JavaScript / Д. Крокфорд – СПб. : Питер, 2019. – 304 с.
4. Форсье, Д. Django. Разработка веб-приложения на Python / Д. Форсье, П. Биссекс – СПб. : Символ-Плюс, 2009. – 456 с.
5. Файн, Я. Angular и TypeScript. Сайтостроение для профессионалов / Я. Файн, А. Моисеев – СПб. : Питер, 2018. – 464 с.

**ПРИЛОЖЕНИЕ А**

**Техническое задание на программное средство**

СОГЛАСОВАНО

Доцент кафедры «ПОВТ и АС»

\_\_\_\_\_\_\_\_\_\_ А.В. Криворучко

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

УТВЕРЖДЕНО

Зав. кафедрой «ПОВТиАС»

\_\_\_\_\_\_\_\_\_\_ В.В. Долгов

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2020 г.

**A.1 Введение**

**А.1.1 Наименование программы**

Наименование программы – «Chat messenger».

**А.1.2 Объект внедрения**

Данное программное средство предназначено для массового использования. Не является специализированным ПО. Может быть внедрено на ПК с любой операционной системой.

**А.2 Основания для разработки**

Разработка проводится на основании документа «Учебный план для студентов ВУЗа», факультета «Информатика и вычислительная техника», обучающихся по специальности 09.03.04 «Программная инженерия»,

**A.3 Назначение разработки**

**A.3.1 Функциональное назначение**

Функциональным назначением программы является обмен сообщениями.

**A.3.2 Эксплуатационное назначение**

Конечными пользователями могут являться любые владельцы. Программа должна использоваться общения между пользователями.

**A.4 Требования к программе или программному изделию**

**А.4.1 Требования к функциональным характеристикам**

Программное средство должно осуществлять следующие функции:

1. Регистрация пользователя с помощью администратора.
2. Авторизация пользователей в системе.
3. Подключение к различным каналам связи через протокол WebSocket.
4. Отправка сообщений пользователям.

**А.4.2 Требования к надежности**

Надежное функционирование программы должно быть обеспечено выполнением совокупности нижеописанных мероприятий:

1. Выполнение требований ГОСТ 51188-98.

2. Защита информации.

3. Организацией бесперебойного питания технических средств.

4. Использованием лицензионного программного обеспечения.

**А.4.3 Условия эксплуатации**

Для функционирования программного продукта необходимо соблюдение всех требований и правил эксплуатации компьютерной и мобильной техники.

Высокая квалификация пользователя программного средства не требуется. Дополнительных требований и ограничений не вводится.

Требования к персоналу, работающему с данным программным продуктом - общие знания вычислительной техники.

**А.4.4 Требования к составу и параметрам технических средств**

Состав технических средств на стороне клиента: компьютер.

**А.4.5 Требования к информационной и программной совместимости**

Для функционирования программного средства на стороне клиента необходимо любое программное обеспечение.

**А.4.5.1 Язык программирования**

Языками программирования для разработки выбраны Python и TypeScript.

**А.4.5.2 Операционная система**

Программное средство должно работать под управлением OC Linux.

**А.4.6 Требования к упаковке и маркировке**

Требования к упаковке и маркировке программного средства не предъявляется.

**А.4.7 Требования к транспортировке и хранению**

Условия транспортирования, места хранения, условия складирования и сроки хранения в различных условиях должны соответствовать требованиям, предъявляемым к носителям информации, на которых будет содержаться данное программное изделие.

Допустимы все способы транспортирования и хранения, не нарушающие целостность используемого носителя данных. Программное средство может храниться на любом носителе информации, имеющее возможность подключения к персональному компьютеру.

**А.4.8 Специальные требования**

Для корректной работы программного средства необходимо разрешение на работу с памятью устройства, чтобы обеспечить корректное открытие и сохранение файлов.

**А.5 Требования к программной документации**

Программная документация должна состоять из следующих листов:

1. Титульный лист.

2. Пояснительная записка к производственной практике.

3. Техническое задание по ГОСТ 19.201-78 ЕСПД.

4. Исходный код программного средства по ГОСТ 19.401-79 ЕСПД.

**А.6 Технико-экономические показатели**

Ориентировочная экономическая эффективность не рассчитывается.

**А.7 Стадии и этапы разработки**

1. Постановка задачи (с 12.04.20 по 13.02.20).

2. Изучение предметной области (с 21.04.20 по 22.04.20).

3. Разработка алгоритмов решения задачи (с 22.04.20 по 01.05.20).

4. Разработка программы (с 02.05.20 по 13.05.20).

5. Тестирование программы (с 13.05.20 по 14.05.20).

**А.8 Порядок контроля и приемки**

Порядок и контроль приёмки определяются заведующим кафедрой «ПОВТиАС» и основаны на демонстрации знаний технологии и умении создавать программные средства для различных предметных областей.

Главным требованием к приемке является наличие правильно работающего программного средства с тестовым примером и отчета, представленного в печатном виде.

Разработчик технического задания /Речков Андрей Леонидович/

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

**ПРИЛОЖЕНИЕ Б**

**Листинг программы**

Листинг Б1 - код класса ChatConsumer

import json

from django.contrib.auth import get\_user\_model

from channels.generic.websocket import WebsocketConsumer

from asgiref.sync import async\_to\_sync

from .models import Message

from .views import get\_last\_10\_messages, get\_user\_contact, get\_current\_chat

User = get\_user\_model()

class ChatConsumer(WebsocketConsumer):

def fetch\_messages(self, data):

messages = get\_last\_10\_messages(data['chatId'])

content = {

'command': 'messages',

'messages': self.messages\_to\_json(messages)

}

self.send\_message(content)

def new\_message(self, data):

user\_contact = get\_user\_contact(data['from'])

message = Message.objects.create(

contact=user\_contact,

content=data['message'])

current\_chat = get\_current\_chat(data['chatId'])

current\_chat.messages.add(message)

current\_chat.save()

content = {

'command': 'new\_message',

'message': self.message\_to\_json(message)

}

return self.send\_chat\_message(content)

def messages\_to\_json(self, messages):

result = []

for message in messages:

result.append(self.message\_to\_json(message))

return result

def message\_to\_json(self, message):

return {

'id': message.id,

'author': message.contact.user.username,

'content': message.content,

'timestamp': str(message.timestamp)

}

commands = {

'fetch\_messages': fetch\_messages,

'new\_message': new\_message

}

def connect(self):

self.room\_name =self.scope['url\_route']['kwargs']['room\_name']

self.room\_group\_name = 'chat\_%s' % self.room\_name

async\_to\_sync(self.channel\_layer.group\_add)(

self.room\_group\_name,

self.channel\_name

)

self.accept()

def disconnect(self, close\_code):

async\_to\_sync(self.channel\_layer.group\_discard)(

self.room\_group\_name,

self.channel\_name

)

def receive(self, text\_data):

data = json.loads(text\_data)

self.commands[data['command']](self, data)

def send\_chat\_message(self, message):

async\_to\_sync(self.channel\_layer.group\_send)(

self.room\_group\_name,

{

'type': 'chat\_message',

'message': message

}

)

def send\_message(self, message):

self.send(text\_data=json.dumps(message))

def chat\_message(self, event):

message = event['message']

self.send(text\_data=json.dumps(message))

Листинг Б2 - код класса ChatContent

import { Component, OnDestroy, OnInit } from '@angular/core';  
import { forkJoin, Subject } from 'rxjs';  
import { takeUntil } from 'rxjs/operators';  
import { ApiService } from '../../../shared/service/api.service';  
  
export class ChatComponent implements OnInit, OnDestroy {  
public netImage: any = "../assets/avatar-3.png";  
public userChannel: any;  
public user: any;  
public dropMenu = [];  
public contactItem = [];  
  
private destroy$ = new Subject();  
public selected: any;  
  
constructor(private api: ApiService) {}  
  
ngOnInit(): void {  
forkJoin(this.api.getChannel(this.api.getUserName()),  
this.api.getUserAll().pipe(takeUntil(this.destroy$))  
).pipe(takeUntil(this.destroy$))  
.subscribe(([participants, userAll]) => {  
participants.forEach((participant, index) => {  
const array = participant.participants.filter(player => player !== this.api.getUserName());  
this.dropMenu.push({id: index + 1, username: array[0], channelId: id, messages: participant.messages});  
})  
this.contactItem = userAll.filter(player => player.username !== this.api.getUserName());  
userAll.filter(user => this.dropMenu.some(participant => {  
if (user.username === participant.username) {  
participant['name'] = `${user.person.firstName} ${user.person.lastName}`  
if (user.person.image) {  
participant['image'] = user.person.image;  
} else {  
participant['image'] = this.netImage;  
}  
}  
}))  
});  
}  
  
ngOnDestroy(): void {  
[this.destroy$.next](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fthis.destroy%24.next)();  
this.destroy$.complete();  
}  
  
public chatChannel(user: object): void {  
this.userChannel = user;  
}  
}

<div class="container-fluid">  
<div class="sidepanel col-4">  
<div class="header">  
<svg class="start-svg" viewBox="0 0 100 80" width="40" height="40">  
<rect width="70" height="5"></rect>  
<rect y="20" width="70" height="5"></rect>  
<rect y="40" width="40" height="5"></rect>  
</svg>  
<div class="test">  
<ng-select [items]="contactItem"  
bindLabel="username"  
placeholder="Выбере пользователя"  
appendTo="body"  
multiple="true"  
[(ngModel)]="selected">  
</ng-select>  
</div>  
<div class="btn-group">  
<button [style]="{'width': '70px', 'height': '38px'}" class="btn btn-primary" type="button">Add</button>  
<div class="dropdown-menu"></div>  
</div>  
</div>  
<div class="people">  
<ul>  
<li \*ngFor="let contact of dropMenu" class="contact" (click)="chatChannel(contact)">  
<div class="people-info">  
<div class="image">  
<img [src]="contact.image" alt="" />  
</div>  
<div class="meta">  
<p class="name">{{ contact.name }}</p>  
<p class="preview">{{ contact.text }}</p>  
</div>  
</div>  
</li>  
</ul>  
</div>  
</div>  
<app-chat-content class="col-8 p-0" [userChannel]="userChannel"></app-chat-content>  
</div>