



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

КУРСОВАЯ РАБОТА

по дисциплине: Разработка клиент-серверных приложений
по профилю: Разработка программных продуктов и проектирование информационных систем
направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Разработка клиент-серверного фуллстек-приложения для чата сотрудников организации

Студент: Эртек Хусейн Ибрахимович

Группа: ИКБО-20-19

Работа представлена к защите _____ (дата) _____ /Эртек Х.И. /
(подпись и ф.и.о. студента)

Руководитель: доцент, Куликов Александр Анатольевич

Работа допущена к защите _____ (дата) _____ /Куликов А.А. /
(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: _____

_____ / 21.05.2022, доцент, Куликов Александр Анатольевич /

_____ / 21.05.2022, _____ /

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиПО)

ЗАДАНИЕ

на выполнение курсовой работы

по дисциплине: Разработка клиент-серверных приложений

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Эрте́к Хусейн Ибрахимович

Группа: ИКБО-20-19

Срок представления к защите: 20.05.2022

Руководитель: к.т.н., доцент Куликов Александр Анатольевич

Тема: Разработка клиент-серверного фуллстек-приложения для чата сотрудников организации


Исходные данные: MongoDB, Express.js, React, Node.js, Bootstrap, Angular, Git, GitHub, Heroku

Перечень вопросов, подлежащих разработке, и обязательного графического материала: 1. Провести анализ предметной области для выбранной темы с обоснованием выбора клиент-серверной архитектуры для разрабатываемого приложения; 2. Выбрать, ориентируясь на мировой опыт и стандарты в данной области, а также на выбранную тему, программный стек для реализации фуллстек разработки; 3. Дать описание архитектуры разрабатываемого клиент-серверного приложения, с помощью UML нотаций и с обоснованием выбора вида клиента(браузер, мобильное приложение, кроссплатформенный клиент и т.д.) ; 4.Провести реализацию фронтенд и бекенд части клиент-серверного приложения, обеспечив версиюный контроль процесса разработки с помощью Git. В разработанном приложении должна обеспечиваться авторизация и аутентификация пользователя; 5. Разместить проект клиент-серверного приложения в репозитории GitHub с приложением в тексте отчёта ссылки на данный репозиторий; 6. Интегрировать проект на GitHub с Heroku, с целью развёртывания разработанного клиент-серверного приложения в облаке. Ссылку на приложение в облаке Heroku привести в тексте пояснительной записки. 7. Провести проверку функционирования минимально жизнеспособного продукта с использованием сгенерированных тестовых данных. 8. Разработать презентацию с графическими материалами.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО:  / Болбаков Р. Г. /, « 08 » 02 2022 г.

Задание на КР выдал:  / Куликов А. А. /, « 28 » 02 2022 г.

Задание на КР получил:  / Эрте́к Х.И. /, « 28 » февраля 2022 г.

Оглавление

Введение.....	6
1 Анализ предметной области разрабатываемого веб-приложения	7
1.1 Систематика выбора предметной направленности веб-приложения.....	7
1.2 Выводы к первой главе	10
2 Выбор технологий разработки веб-приложения.....	11
2.1 TypeScript – язык программирования	11
2.2 Microsoft VS Code – среда разработки	12
2.3 СУБД PostgreSQL	12
2.4 AngularJS – JavaScript фреймворк.....	13
2.5 NestJS – фреймворк веб-приложений	13
2.6 Node.js – среда выполнения JavaScript.....	14
2.7 Вывод ко второй главе.....	14
3 Описание архитектуры веб-приложения	15
3.1 Концепция MVC	15
3.2 Обоснование выбора клиента	16
3.3 Выводы к третьей главе	18
4 Разработка клиентской-серверной части приложения.....	19
4.1 Структура пользовательского интерфейса.....	19
4.2 Структура базы данных.....	28
4.3 Выводы к четвертой главе.....	29
5 Интеграция с GitHub и облачной платформой	30
5.1 Репозиторий GitHub.....	30
5.2 Выводы к пятой главе	30
6 Проверка минимально жизнеспособного продукта.....	31
6.1 MVP.....	31
6.2 Выводы к шестой главе.....	31
Заключение.....	32
Список использованных источников.....	33

Введение

Актуальность и цель исследования

Чат сотрудников организации – это программа или платформа для компьютера или мобильного телефона, которая даёт возможность группам людей или отдельно взятому предприятию коммуницировать между собственными рабочими как в отдельности, в частных сообщениях, так и в общей беседе (чате) своего отдела.

В сегодняшних реалиях трудно представить какую-либо крупную компанию без корпоративного мессенджера, который помогает бизнесу, систематизируя корпоративную коммуникацию, а также обеспечивая максимально защищённое взаимодействие между сотрудниками и партнёрами [1].

Однако в нынешних реалиях лишь некоторые крупные компании прибегают к созданию собственной корпоративного мессенджера, в связи большими затратами на разработку и поддержку мессенджера. Чаще всего наиболее востребованными приложениями для коммуникации предприятий становятся: WhatsApp, Telegram, так как они являются бесплатными (WhatsApp имеет функцию «Бизнес-аккаунт» для работы малого бизнеса, но эти функции можно не использовать) [2, 3].

В данной курсовой работе была выбрана тема “Чат сотрудников организации” для объяснения и визуализации принципа работы корпоративных мессенджеров, широко используемых в нынешних реалиях.

Целью курсовой работы является изучение и понимание принципа работы клиентской и серверной частей веб-приложений, а также создание приложения в полном цикле разработки, от анализа решения до его реализации и внедрения.

1 Анализ предметной области разрабатываемого веб-приложения

Тема веб-приложения – «Чат сотрудников организации».

1.1 Систематика выбора предметной направленности веб-приложения

Чтобы наиболее полно понимать, что из себя представляет чат сотрудников организации (корпоративный мессенджер), стоит рассмотреть несколько успешных и популярных примеров, присутствующих на рынке на данный момент:

1) WhatsApp (Рисунок 1)

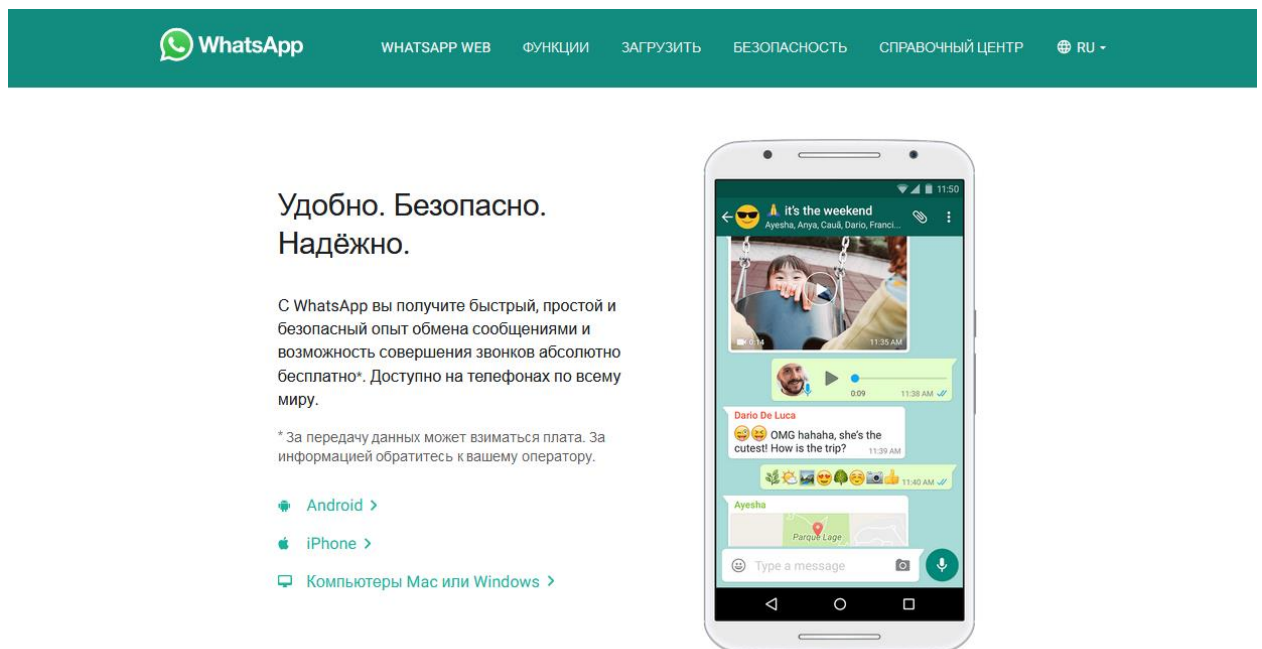


Рисунок 1 – Главная страница сайта whatsapp.com

Данный сервис позволяет любому пользователю, а также предприятиям коммуницировать друг с другом в частных сообщениях или чат-беседах разных групп или отделов. Пользователи могут обмениваться информацией, отправлять документы, таблицы, презентации без использования сторонних ресурсов (Рисунок 2, 3).

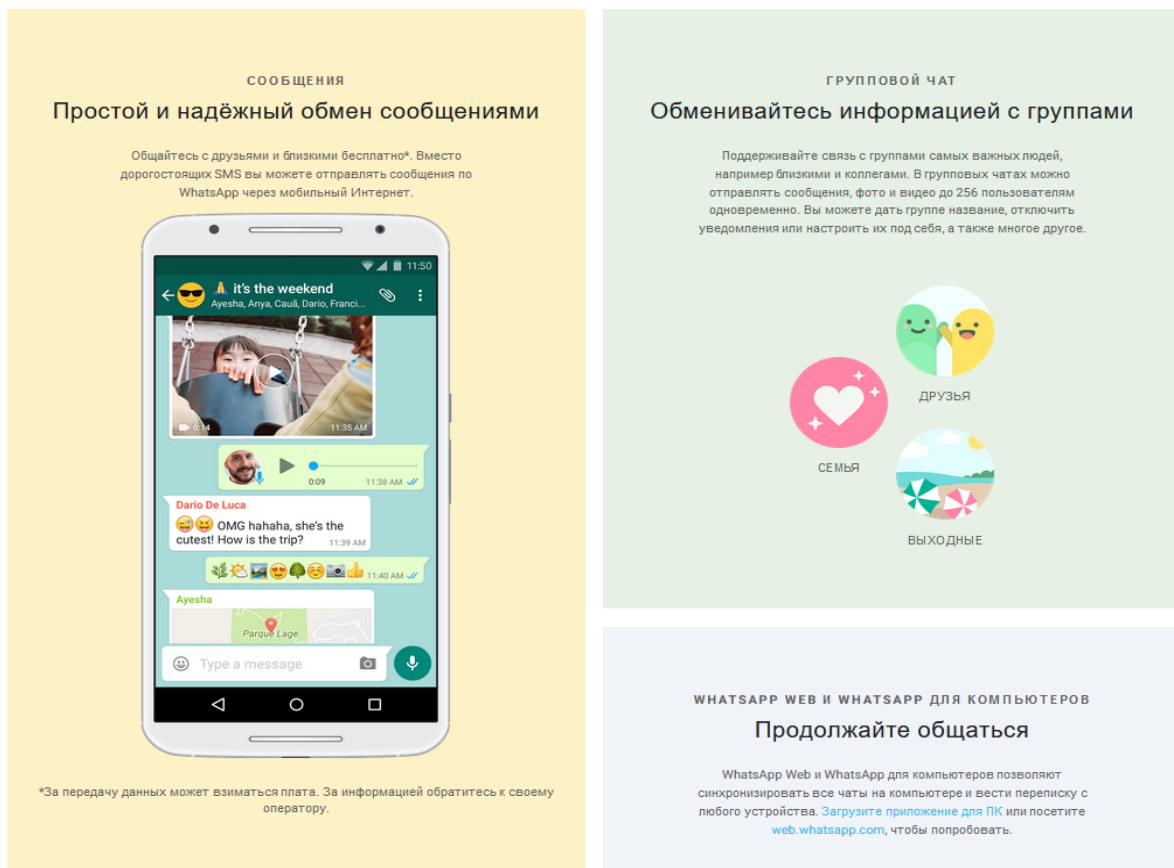


Рисунок 2 – Страница «Функции» на сайте whatsapp.com. Часть 1

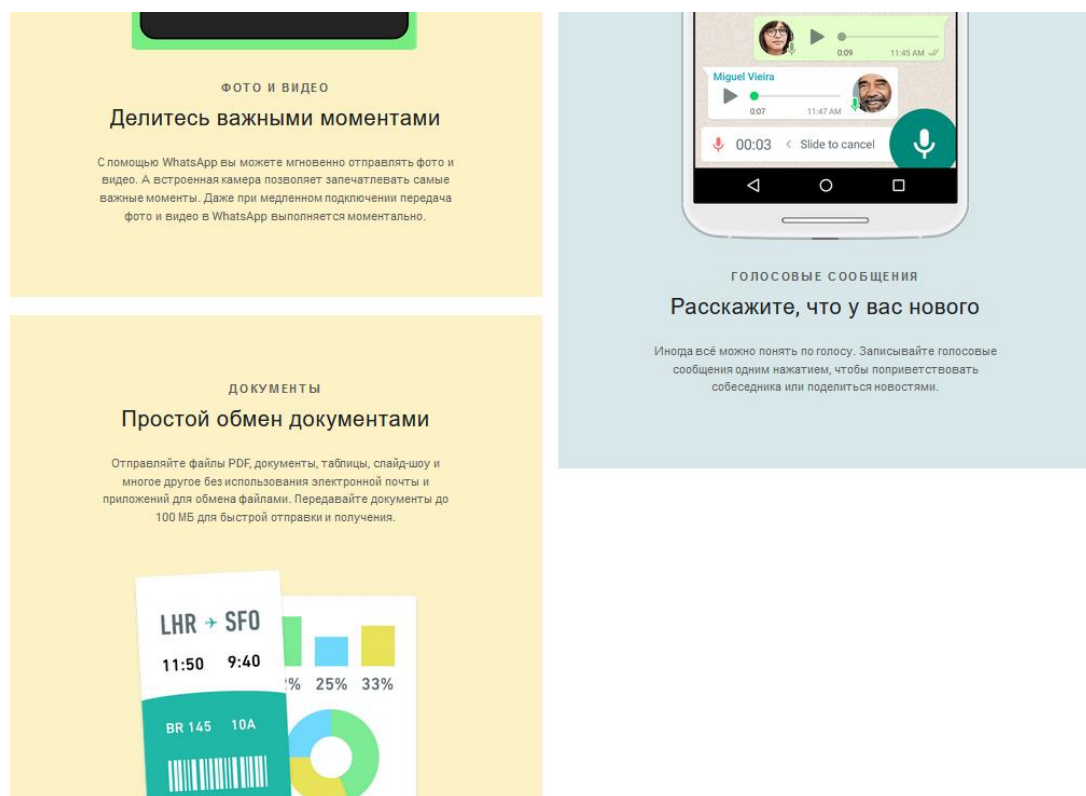


Рисунок 3 – Страница «Функции» на сайте whatsapp.com. Часть 2

2) Telegram (Рисунок 4)

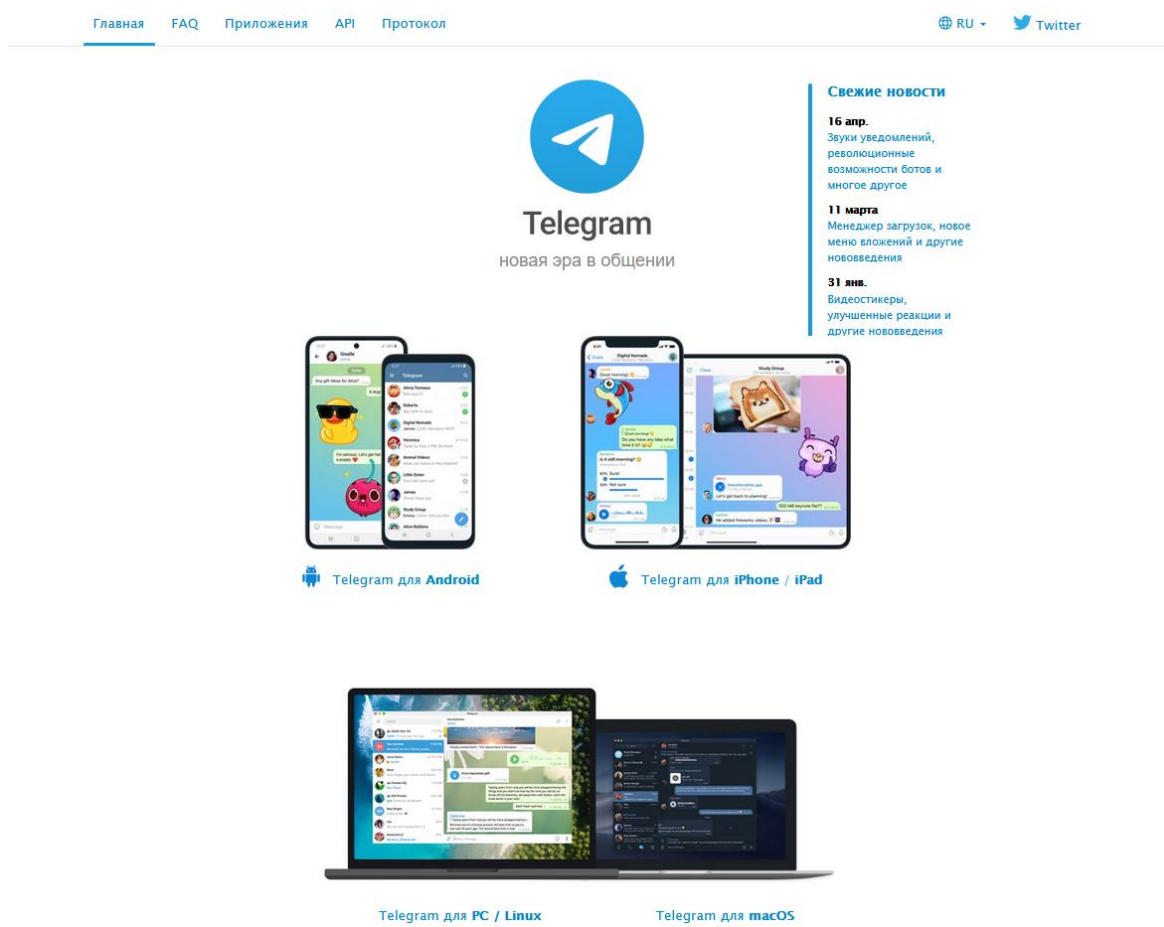


Рисунок 4 – Главная страница сайта telegram.org

Telegram является еще одним примером сервиса, который использует в качестве корпоративного мессенджера многие компании. В приложении присутствуют возможности обмена информацией, совершения звонков (приватных и групповых), создания групп или бесед для разных групп людей, а также поддержка отправки различного вида контента: видео, аудио, текста, документа и прочее. Если есть необходимость, предприятие может настроить все нужные функции мессенджера под себя, если у таковой имеется штаб программистов, способные использовать исходный код и API для разработчиков (Рисунок 5).

Девять причин пользоваться Telegram



Рисунок 5 – Главная страница сайта telegram.org

1.2 Выводы к первой главе

Основные причины выбора данной темы:

1. Обширное использование приложений и сервисов мессенджеров
2. Повысить интерес в создании собственных приложений для чатов организаций в сфере корпоративных мессенджеров
3. Поддержка дополнительных возможностей (видеосвязь, обмен медиафайлами и пр.), расширяющих функциональность сервиса

Все эти факторы показывают востребованность данной тематики в наше время, что позитивно сказывается на развитии технологий для поддержки подобных сервисов и приложений.

В данном веб-приложении реализован удобный и интуитивно понятный интерфейс для взаимодействия пользователя с системой, позволяющий совершать пять основных действия: аутентификация в приложении, создание, изменение и удаление постов, которые размещают пользователи корпоративного мессенджера, а также обмен сообщениями.

2 Выбор технологий разработки веб-приложения

Веб-приложение было реализовано при помощи следующего стека технологий: PostgreSQL – реляционная база данных [4], Angular.js – фреймворк для создания клиентской части веб-приложения [5], Nest.js – фреймворк для создания серверной части веб-приложения [6], Node.js – среда для исполнения JavaScript (TypeScript) кода [7]. В качестве IDE была использована программа от компании Microsoft – VS Code [8].

2.1 TypeScript – язык программирования

TypeScript – является обратно совместимым с JavaScript и компилируется в последний. Фактически, после компиляции программу на TypeScript можно выполнять в любом современном браузере или использовать совместно с серверной платформой Node.js. TypeScript отличается от JavaScript возможностью явного статического назначения типов, поддержкой использования полноценных классов (как в традиционных объектно-ориентированных языках), а также поддержкой подключения модулей, что призвано повысить скорость разработки, облегчить читаемость, рефакторинг и повторное использование кода, помочь осуществлять поиск ошибок на этапе разработки и компиляции, и, возможно, ускорить выполнение программ. Планируется, что в силу полной обратной совместимости адаптация

существующих приложений на новый язык программирования может происходить поэтапно, путём постепенного определения типов. [9, 10].

Был выбран в качестве основного языка программирования в курсовой работе, так как подходит для одновременной работы с фреймворками AngularJS и NestJS.

2.2 Microsoft VS Code – среда разработки

VS Code - редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки облачных и веб-приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией [8].

Данная среда разработки была выбрана по причине функциональности и удобной системы тестирования и отладки кода.

2.3 СУБД PostgreSQL

PostgreSQL – это объектно-реляционная система управления базами данных (ORDBMS). Базируется на языке SQL и поддерживает многочисленные возможности. Наиболее развита из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных [11].

Преимущества PostgreSQL [12]:

- Поддержка БД неограниченного размера;
- Мощные и надёжные механизмы транзакций и репликации;
- Расширяемая система встроенных языков программирования и поддержка загрузки C-совместимых модулей;
- Наследование;
- Легкая расширяемость.

MongoDB был заменён в связи в сложившейся ситуации в мире.

2.4 AngularJS – JavaScript фреймворк

AngularJS представляет собой opensource JavaScript-фреймворк, использующий шаблон MVC. Собственно использование MVC является его одной из отличительных особенностей. Для описания интерфейса используется декларативное программирование, а бизнес-логика отделена от кода интерфейса, что позволяет улучшить тестируемость и расширяемость приложений. Другой отличительной чертой фреймворка является двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом. Таким образом, AngularJS синхронизирует модель и представление. Кроме того, AngularJS поддерживает такие функциональности, как Ajax, управление структурой DOM, анимация, шаблоны, маршрутизация и так далее. Мощь фреймворка, наличие богатого функционала во многом повлияла на то, что он находит свое применение во все большем количестве веб-приложений, являясь на данный момент, наверное, одним из самых популярных javascript-фреймворков. [5].

Angular был выбран по причине более удобной работы с механизмом построения приложений, направленных на социальное взаимодействие.

2.5 NestJS – фреймворк веб-приложений

NestJS — это фреймворк, который ускоряет и упрощает разработку масштабируемых серверных приложений на основе программной платформы Node.js. Он использует прогрессивный JavaScript, полностью поддерживает TypeScript и совмещает в себе три парадигмы (концепции) программирования: объектно-ориентированную, функциональную, функционально-реактивную [6].

Данный фреймворк был выбран, так как NestJS позволяет подключать к Node.js любые базы данных и сервисы, а также использовать любые сторонние библиотеки, что даёт возможность ускорить разработку серверной части.

2.6 Node.js – среда выполнения JavaScript

Node.js – это кроссплатформенная среда выполнения JavaScript с открытым исходным кодом, выполняющая код JavaScript вне веб-браузера. Node.js позволяет разработчикам использовать JavaScript для написания инструментов командной строки и для серверных сценариев – запускает сценарии на стороне сервера для создания динамического содержимого веб-страницы перед отправкой страницы в веб-браузер. Следовательно, Node.js представляет собой парадигму "JavaScript everywhere", объединяющую разработку веб-приложений вокруг одного языка программирования, а не разных языков для серверных и клиентских сценариев.

Node.js был выбран так как оба фреймворка взаимодействуют с данной платформой и, для Node.js есть множество обучающих пособий, помогающих в написании программного кода.

2.7 Вывод ко второй главе

Благодаря использованию данных инструментов удалось добиться выполнения поставленных в курсовой работе задач по созданию собственного веб-приложения.

3 Описание архитектуры веб-приложения

3.1 Концепция MVC

Веб-приложение построено на архитектуре MVC [14].

Компоненты архитектуры шаблона MVC предназначены для обработки различных аспектов разрабатываемого приложения. Шаблон проектирования MVC служит для отделения уровня представления от бизнес-логики.

MVC является одним из наиболее широко используемых и популярных шаблонов проектирования программного обеспечения для разработки веб-приложений. Шаблон Model View Controller состоит из 3 частей:

- Model
- View
- Controller

Model – хранит и управляет данными. View – графический пользовательский интерфейс. View – это визуальное представление данных - например, диаграмма, диаграмма, таблица, форма. View содержит все функции, которые напрямую взаимодействуют с пользователем - например, нажатие кнопки или событие ввода.

Controller – основа приложения. Контроллер соединяет модель и представление. Контроллер преобразует входные данные из представления в требования для извлечения/обновления данных в модели.

Поток данных и общая схема работы приложения изображены на Рисунке 6.

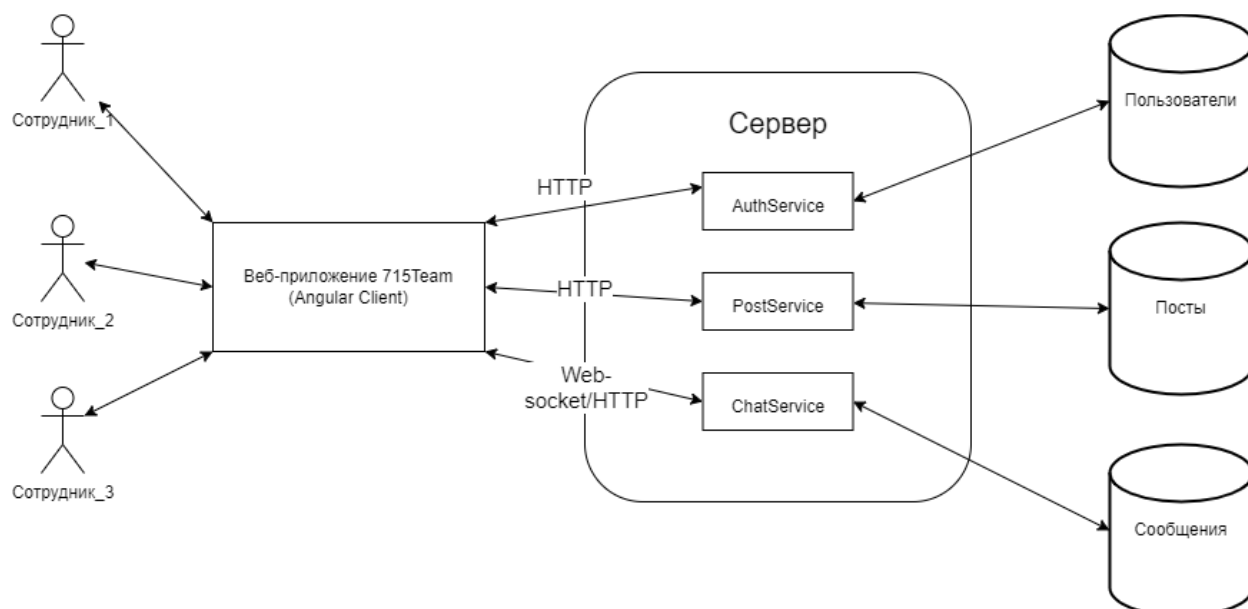


Рисунок 6 – общая архитектура приложения

Верхний уровень функциональные возможности приложения показан на диаграмме вариантов использования (Рисунок 7).

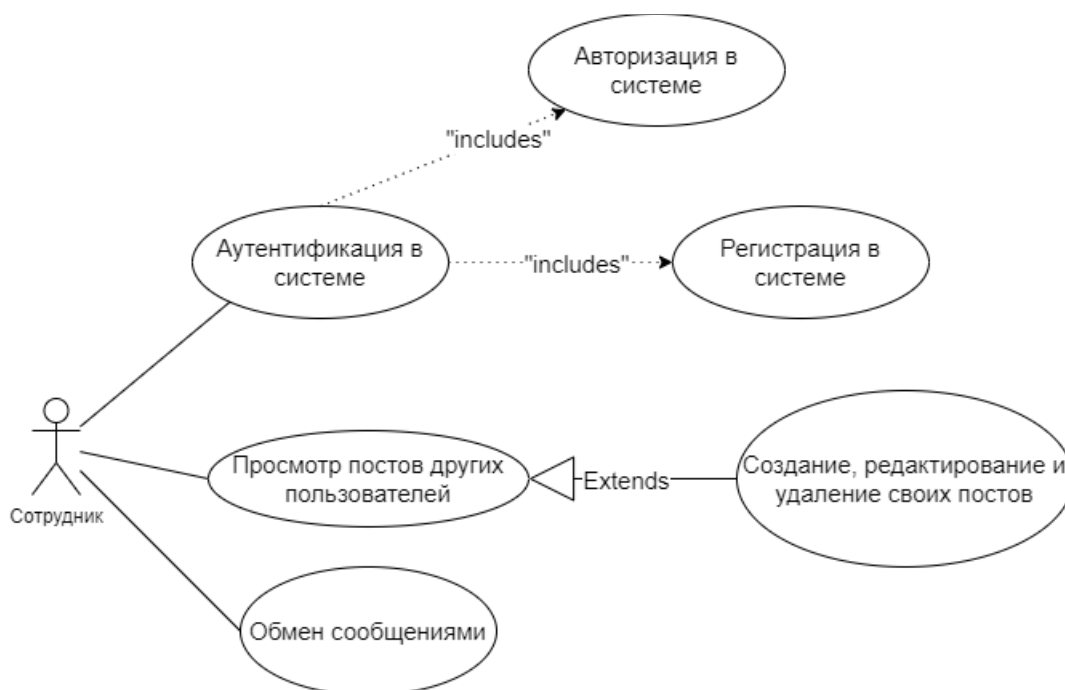


Рисунок 7 – Диаграмма вариантов использования [15]

3.2 Обоснование выбора клиента

Основной платформой (клиентом) для запуска и функционирования веб-приложения был выбран браузер, однако, само приложение адаптировано под мобильные устройства (кроссплатформенное приложение). Получение точной статистики по использованию сервиса корпоративного мессенджера не

представляется возможным, но удалось проанализировать имеющуюся статистику по мессенджеру Telegram (Рисунок 8)

Сколько людей пользуются Телеграмом?

550 млн **пользователей** были в телеграме к июлю 2021 года. Это в 10 раз больше, чем в 2014 году. 16 мар. 2022 г.

Рисунок 8 – Статистика использования сервиса Telegram [16]

Далее была проверена страница приложения Tinder в Play Market (Рисунок 9).

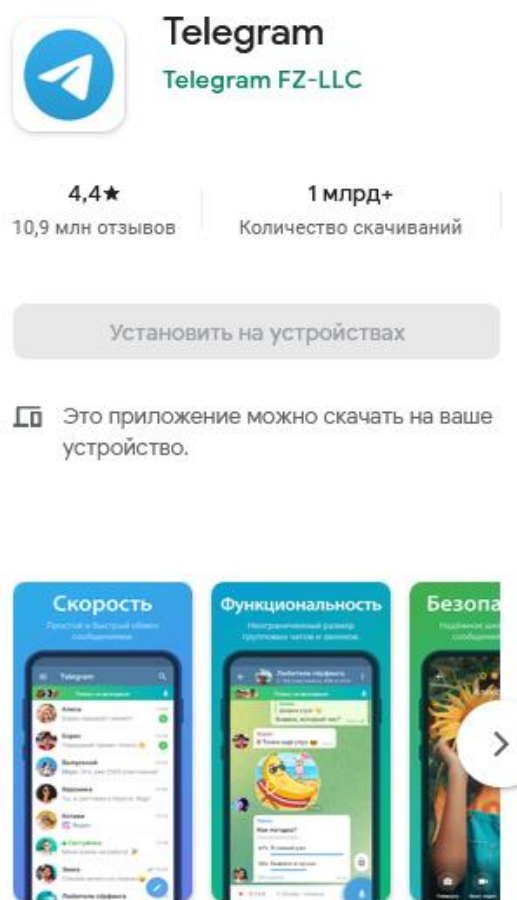


Рисунок 9 – Страница Telegram в Play Market

Как можно заметить, в случае мобильного приложения, количество пользователей демонстрирует предпочтения использования Telegram именно на мобильных устройствах. Но стоит отметить, что большая часть предприятий коммуницирует из офисов, где работа осуществляется на компьютерах.

Полагаясь на эту информацию, было принято решение реализовать веб-приложение, доступ к которому можно получить из любого браузера, а также произведена адаптация под мобильные устройства.

3.3 Выводы к третьей главе

Все части веб-приложения были связаны, следуя архитектурному паттерну MVC, для обеспечения корректной и безотказной работы сервиса, а также проанализирована целевая платформа для запуска и функционирования приложения.

4 Разработка клиентской-серверной части приложения

4.1 Структура пользовательского интерфейса

Для реализации веб-приложения по теме “Чат сотрудников организации” было выбрано представление информации по двум основным страницам, где располагается все функциональные возможности приложения. Также присутствует начальная страница, которая также является страницей авторизации уже зарегистрированных пользователей (Рисунок 10).

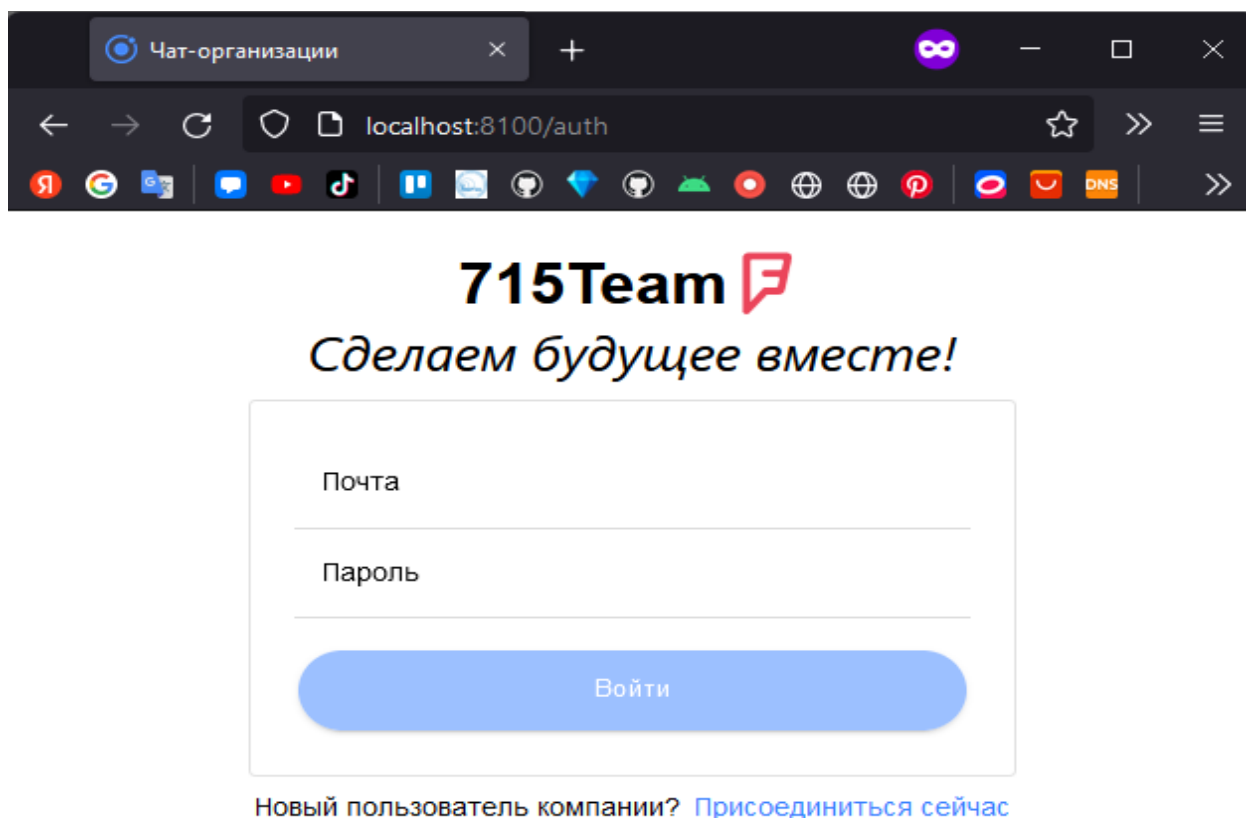
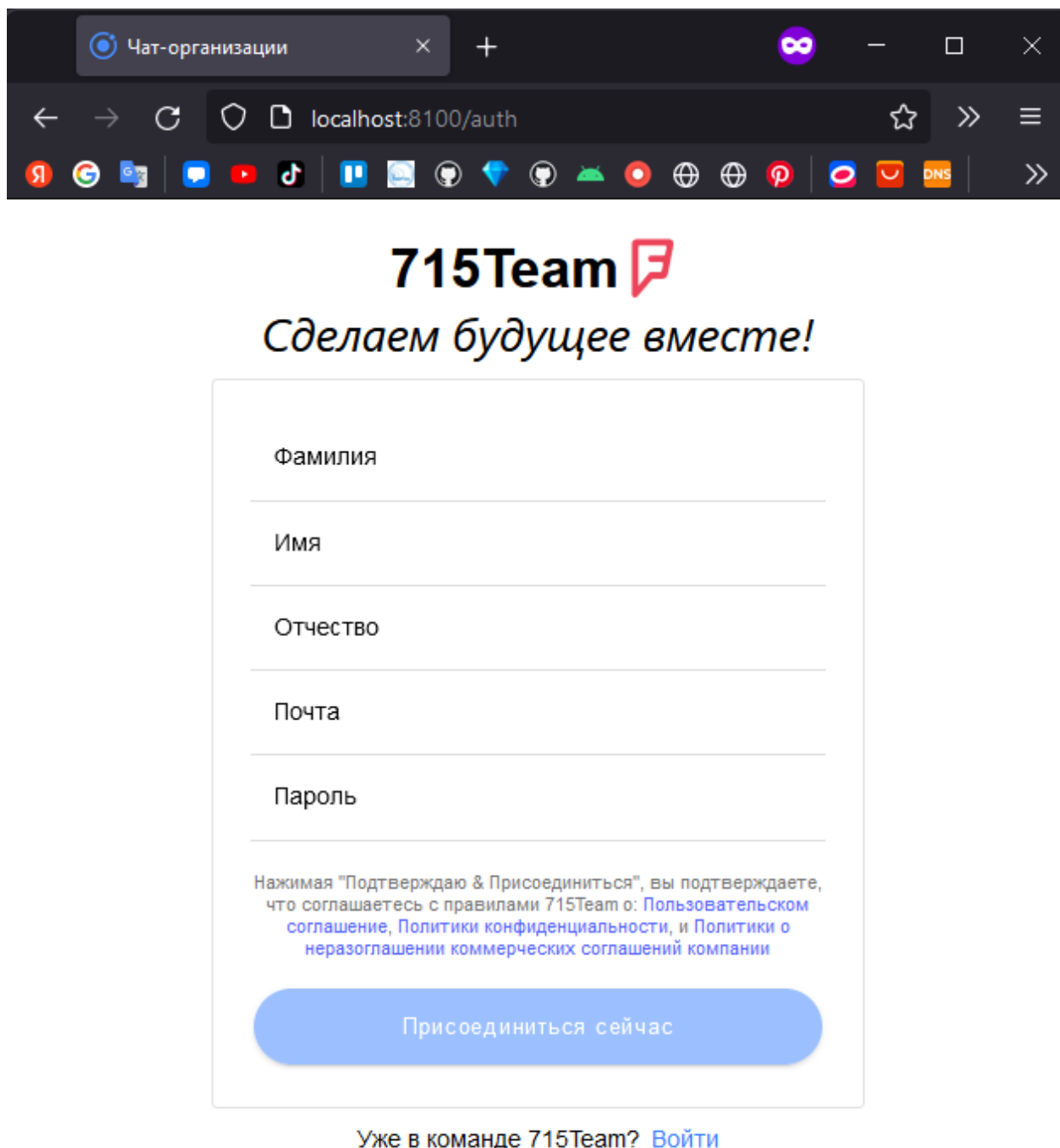


Рисунок 10 – Начальная страница

Для того, чтобы пользователю получить доступ к функционалу сервиса, он должен сперва пройти регистрацию, если он новый сотрудник компании или выполнить вход в соответствующем окне. Хотелось бы отметить, что регистрироваться нужно по специальному почтовому ящику, который предоставляет компания – @teams.ru (Рисунок 11).



The image shows a web browser window with the address bar displaying 'localhost:8100/auth'. The page title is 'Чат-организации'. The main heading is '715Team' with a red speech bubble icon, followed by the tagline 'Сделаем будущее вместе!'. The registration form contains the following fields: 'Фамилия', 'Имя', 'Отчество', 'Почта', and 'Пароль'. Below these fields is a disclaimer: 'Нажимая "Подтверждаю & Присоединиться", вы подтверждаете, что соглашаетесь с правилами 715Team о: Пользовательском соглашении, Политики конфиденциальности, и Политики о неразглашении коммерческих соглашений компании'. At the bottom of the form is a blue button labeled 'Присоединиться сейчас'. Below the form, there is a link: 'Уже в команде 715Team? Войти'.

Рисунок 11 – Форма регистрации

После регистрации пользователь автоматически перенаправляется на страницу главную страницу корпоративного мессенджера, где сразу можно

увидеть посты от других сотрудников, наиболее просматриваемые темы статей, которые публикуются сотрудниками для быстрого доступа к информации (Рисунок 12).

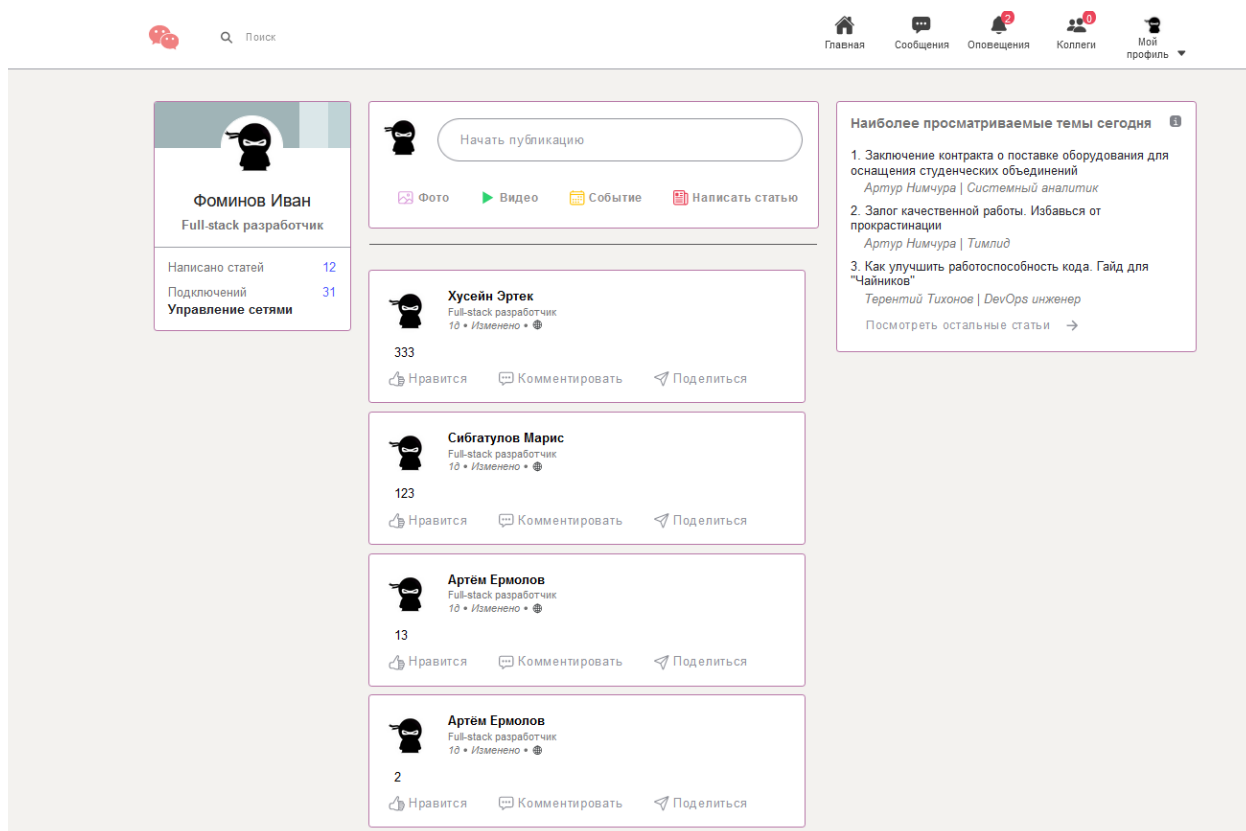


Рисунок 12 – Главная страница сервиса

Как только пользователь входит или регистрирует аккаунт, его профиль приобретает специальную цветовую гамму, которая закреплена за специальными ролями. В данном случае представлен пользователь с ролью user, вместе с ней существует роль директора – headmaster, а также администратор сервиса (Рисунок 13)

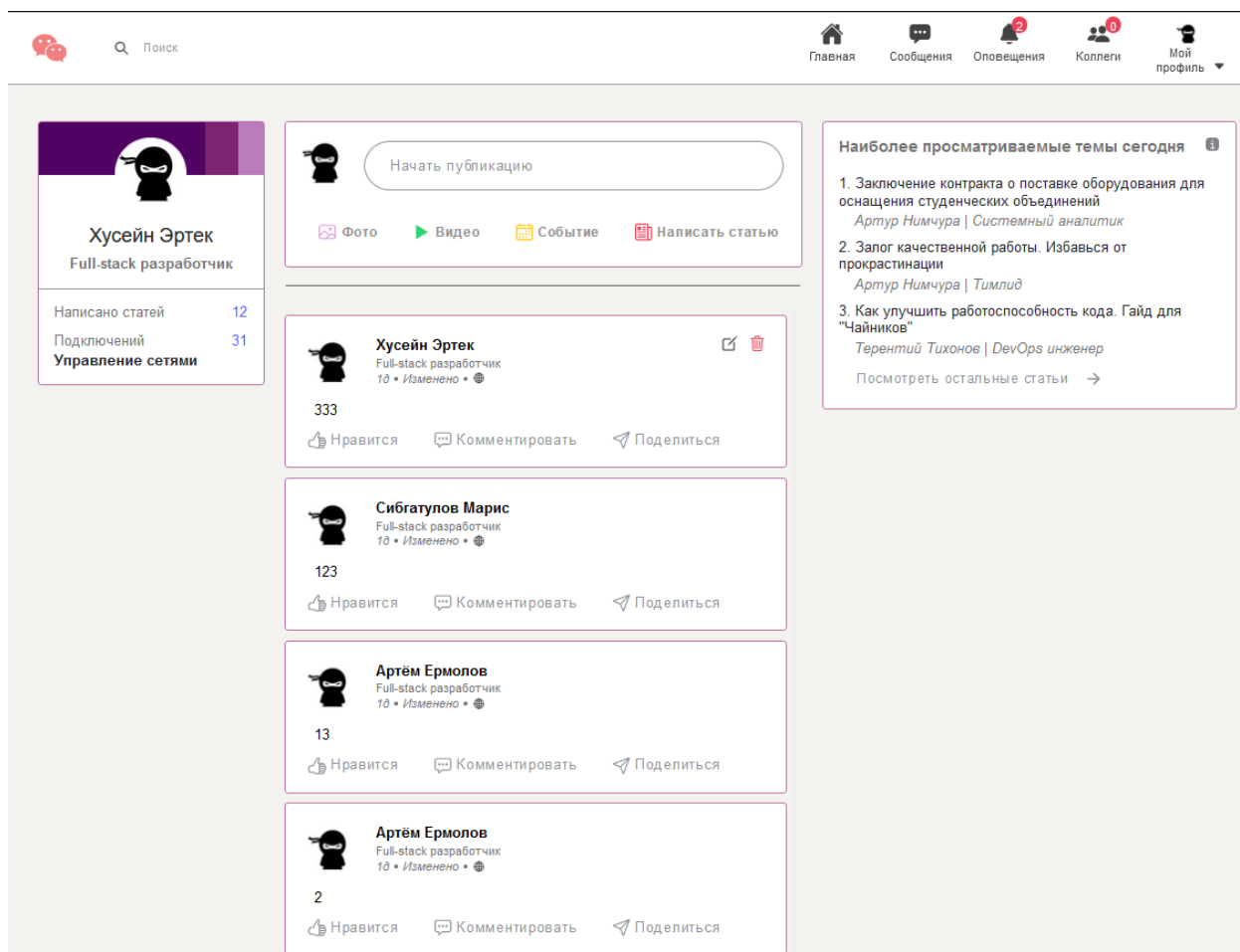


Рисунок 13 – Главная страница сервиса со стороны администратора

Если же пользователю необходимо получить доступ к приложению удалённо через мобильное устройство (Рисунок 14).

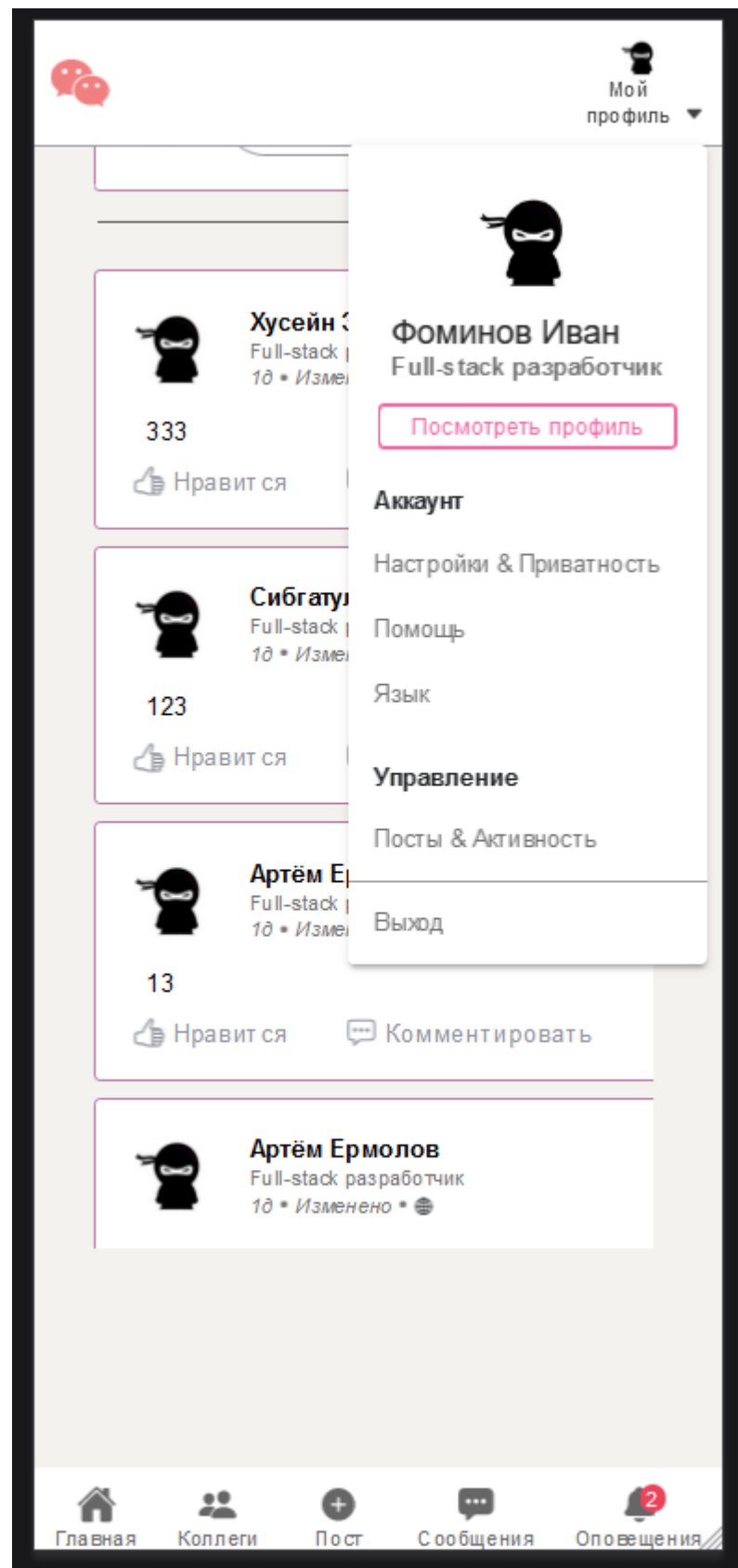


Рисунок 14 – Главная страница со стороны мобильного устройства

Пользователю доступны следующие возможности:

- Просмотр существующих постов
- Создание, редактирование и удаление своих постов
- Возможность добавить другого пользователя в свою сеть для переписывания
- Начать чат с другим сотрудником, который принял заявку

Просмотр существующих постов осуществляется на основной странице. На ней же иной сотрудник может создать свой пост, отредактировать его, если то потребуется, а также удалить (Рисунок 15-18).

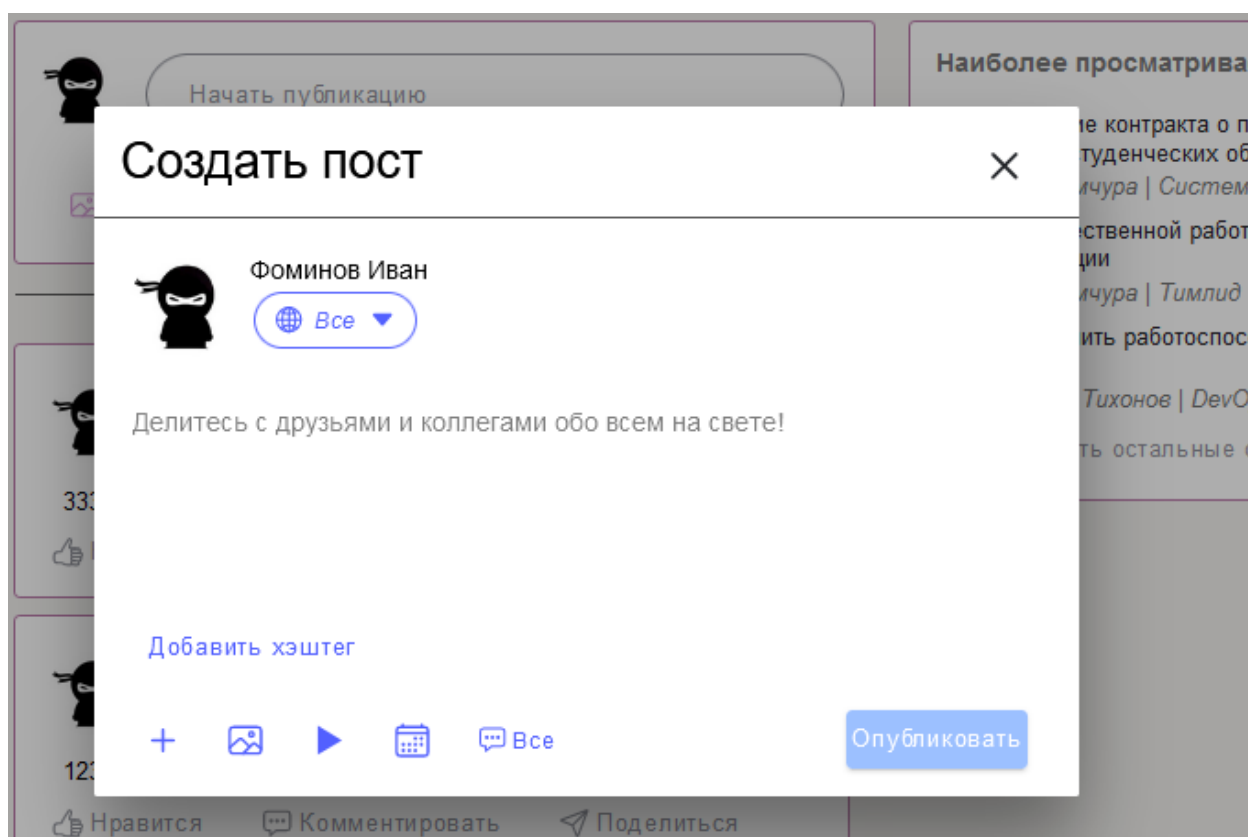


Рисунок 15 – Создание своего поста

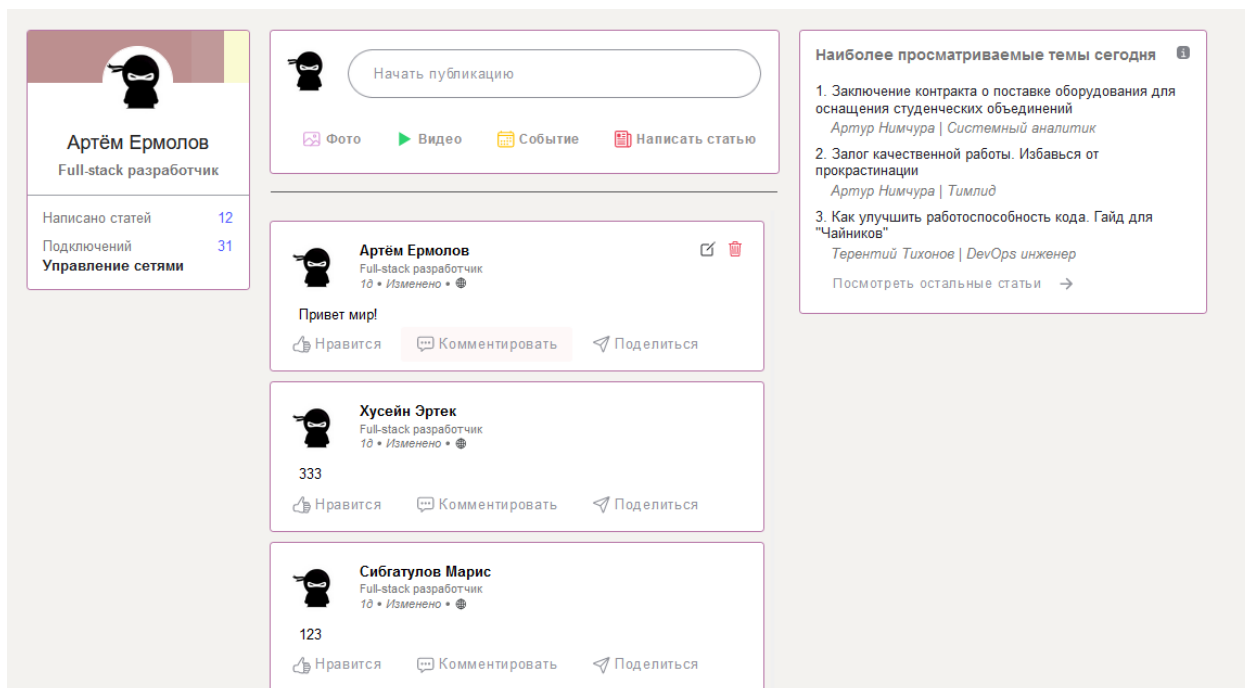


Рисунок 16 – Опубликованный пост

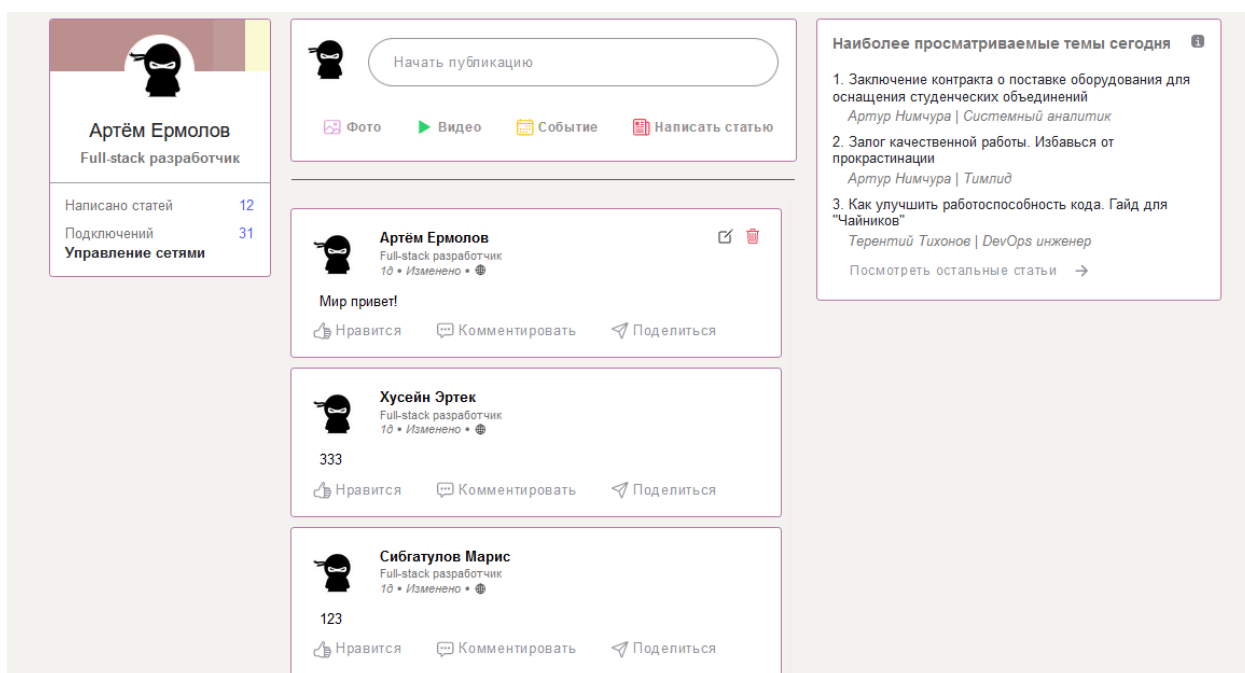


Рисунок 17 – Изменение поста

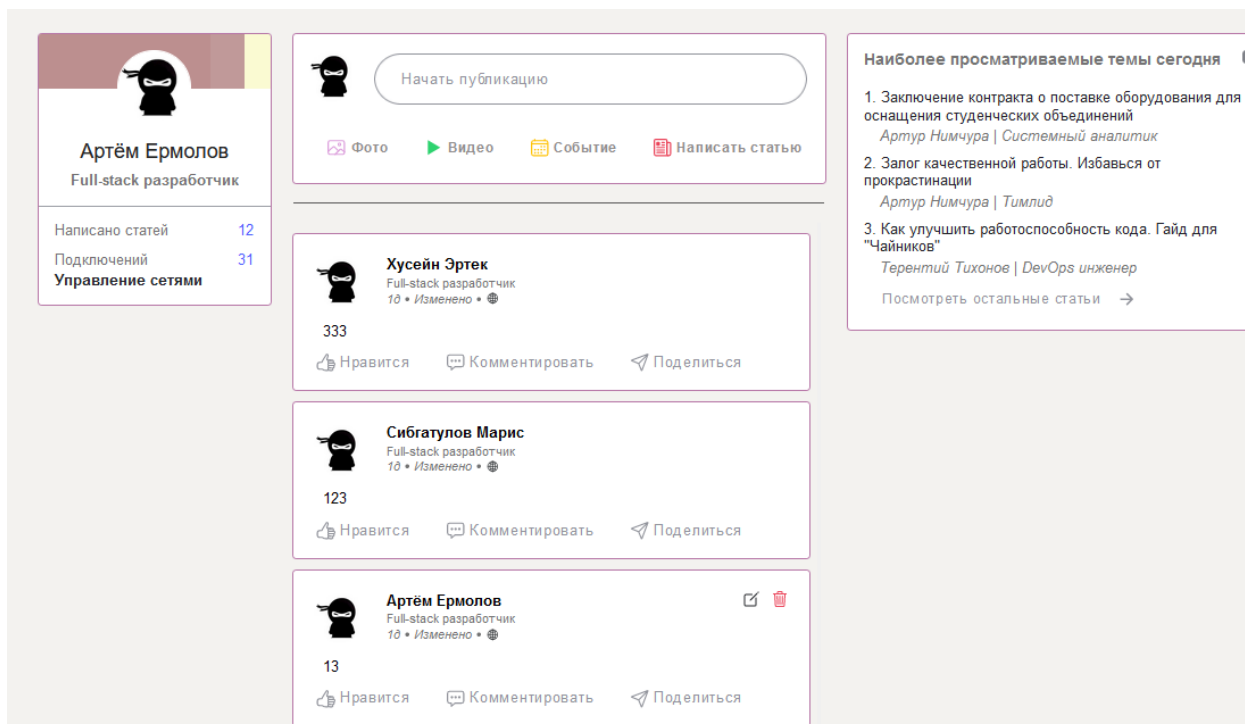


Рисунок 18 – Удаление поста

После того, как новый пользователь подключился к сети компании, его можно найти по посту, кликнув по его имени. Происходит пересылка на профиль сотрудника, где можно нажать кнопку «Подключить», чтобы добавить его в сеть и начать общение (Рисунок 19-21).

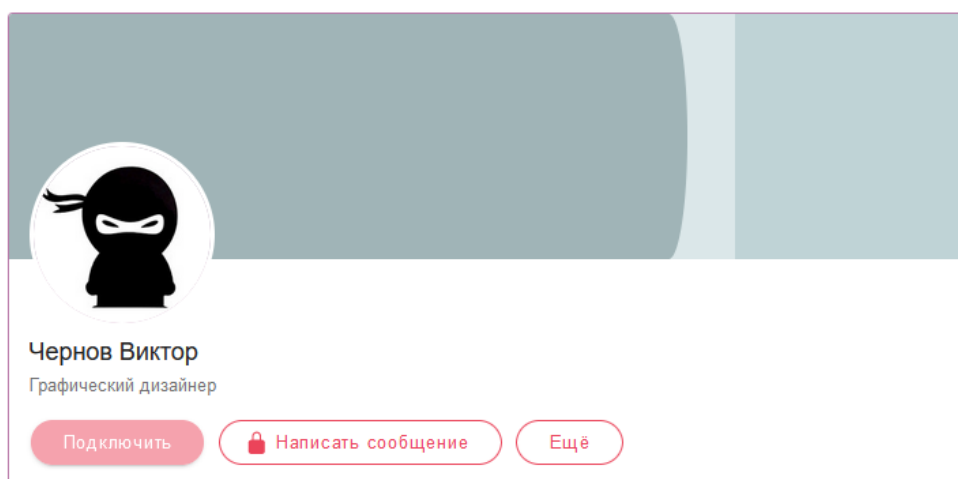


Рисунок 19 – Отправка запроса на подключение к единой сети другому сотруднику

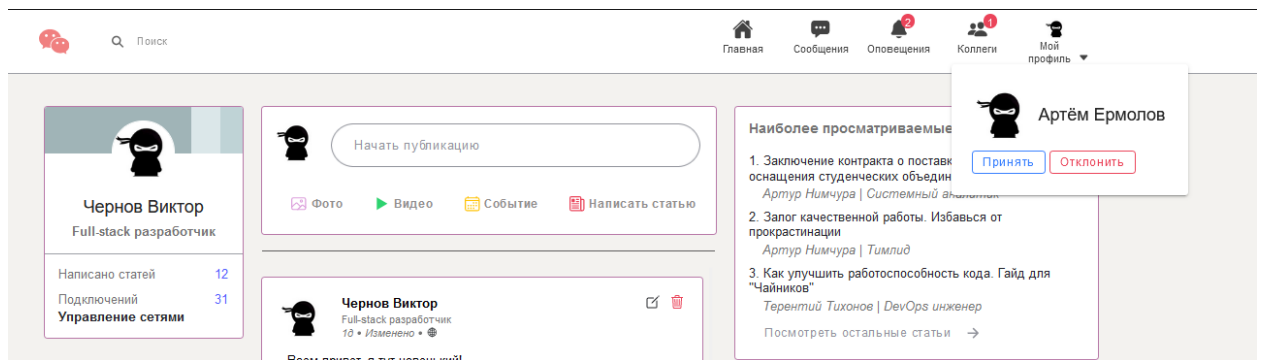


Рисунок 20 – У другого сотрудника отобразилось запрос

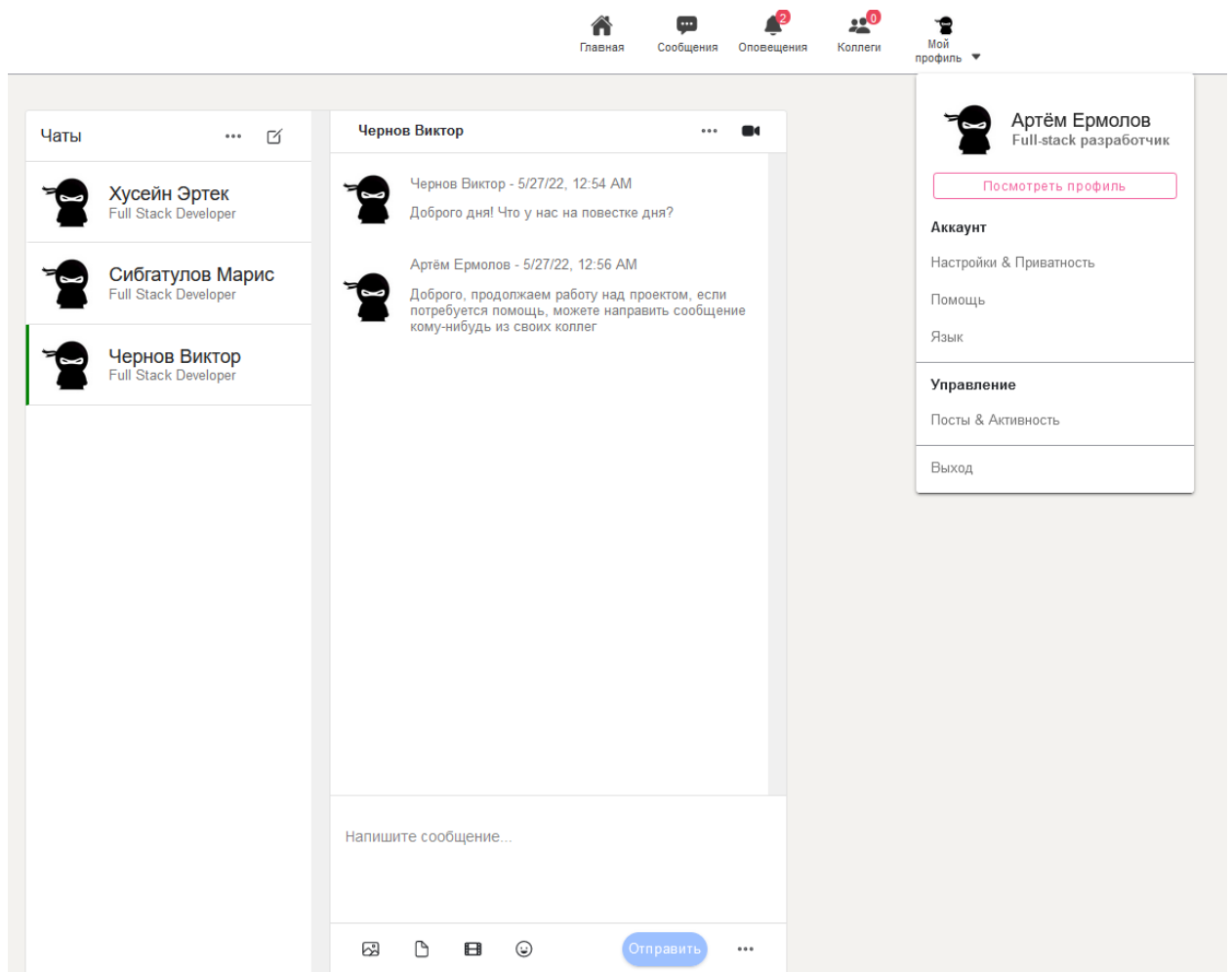


Рисунок 21 – Пример переписки между двумя сотрудниками

4.2 Структура базы данных

Все таблицы базы данных были созданы и связаны между собой автоматически при помощи моделей данных в NestJS [17].

Описание содержимого базы данных (Таблица 1):

- 1) users – учётные записи пользователей
- 2) feed_post – посты всех пользователей
- 3) message – сообщения пользователей
- 4) request – отправленные запросы всех пользователей друг другу
- 5) active_conversation – данные об активном чате

Таблица 1 – содержимое базы данных веб-приложения

Сущность	Атрибут	Описание
users	_id	Уникальный идентификатор сотрудника
	firstName	Имя сотрудника
	lastName	Фамилия сотрудника
	middleName	Отчество сотрудника
	imagePath	Путь до аватара пользователя
	email	Почта сотрудника
	password	Пароль сотрудника (хранится в хешированном виде)
	role	Роль сотрудника
feed_post	_id	Уникальный идентификатор поста
	body	Время отправки сообщения
	createdAt	Дата опубликования поста
	authorId	Уникальный идентификатор автора поста
message	id	Уникальный идентификатор сообщения
	message	Текст сообщения

	createdAt	Дата отправки сообщения
	userId	Уникальный идентификатор сотрудников в чате
	conversationId	Уникальный идентификатор беседы
request	id	Уникальный идентификатор запроса
	status	Статус запроса
	creatorId	Уникальный идентификатор отправителя запроса
	receiverId	Уникальный идентификатор получателя запроса
active_conversation	id	Уникальный идентификатор открытого диалога
	socketId	Уникальный идентификатор сокета
	userId	Уникальный идентификатор сотрудников в чате
	conversationId	Уникальный идентификатор беседы

4.3 Выводы к четвертой главе

После выполнения всех поставленных задач – удалось реализовать и оформить пользовательский интерфейс веб-приложения, содержащий систему регистрации и авторизации, главную страницу для всех сотрудников, возможность опубликовывать и редактировать посты, а также совершать переписку, однако, некоторая часть функционала не была реализована в связи с изменением структур некоторых пакетов, которые позволяют устанавливать фото профиля и совершать отправку документов в чате. Также была описана структура базы данных.

5 Интеграция с GitHub и облачной платформой

5.1 Репозиторий GitHub

Для сохранения состояний проекта была выбрана система контроля версий Git и хранилище пользовательских репозиториях GitHub.

Ссылка на репозиторий – https://github.com/Ri7vaI/Course_Work_RKSP

В связи со сложившейся ситуацией в мире интеграция с облачной платформой Heroku на данный момент недоступна.

5.2 Выводы к пятой главе

Удалось загрузить версию проекта на GitHub для дальнейшей доработки и дополнения его функциональности.

6 Проверка минимально жизнеспособного продукта

6.1 MVP

Тестовые данные – ручной ввод.

Была проверена функциональность каждой сущности и функции, которое присутствует в приложении:

- Регистрация и авторизация – созданы несколько пользователей для проверки работоспособности системы. Также возможность переписываться могут лишь те пользователи, которые подключены к одной сети. Публиковать и редактировать посты могут лишь зарегистрированные сотрудники. Каждый сотрудник может удалить или изменить только свой пост
- Проверена база данных на корректность сохранения и обновления информации, содержащейся в ней, а также форматов, в которых хранятся эти данные.

6.2 Выводы к шестой главе

Благодаря проверке минимально жизнеспособного продукта удалось убедиться в правильности работы основных аспектов веб-приложения, а также доработать оставшуюся часть функционала до рабочего состояния.

Заключение

В ходе выполнения курсовой работы были получены навыки создания веб-приложений, реализации слоя серверной и клиентской логики, слоя логики базы данных. Также, был пройден полный цикл разработки от анализа требований до внедрения и поддержки приложения.

Были выполнены конечные цели работы:

1. Создано веб-приложение с применением технологий PostgreSQL, AngularJS, NestJS и Node.js.
2. Реализован удобный интерфейс доступа к различным частям веб-приложения.
3. Внедрены возможности добавления и редактирования записей из базы данных.
4. Добавлена система регистрации и авторизации.
5. Настроен вывод постов на главной странице, а также возможность манипулировать ими.
6. Настроена система отправки и принятия от других сотрудников, настроено общение в чатах.

Все условия и задачи курсовой работы были выполнены в полном объеме и корректно работают согласно требованиям.

Список использованных источников

1. Социальная сеть, как инструмент преобразования человека и общества [Электронный ресурс]: Режим доступа: https://ru.wikipedia.org/wiki/Корпоративный_мессенджер, свободный (Дата последнего обращения 05.05.22)
2. WhatsApp [Электронный ресурс]: Режим доступа: <https://www.whatsapp.com/?lang=ru>, свободный (Дата последнего обращения 05.05.22)
3. Telegram [Электронный ресурс]: Режим доступа: <https://ru.wikipedia.org/wiki/Telegram>, свободный (Дата последнего обращения 05.05.22)
4. Официальный сайт PostgreSQL [Электронный ресурс]: Режим доступа: <https://www.postgresql.org/>, свободный (Дата последнего обращения: 10.05.22)
5. Официальный сайт AngularJS [Электронный ресурс]: Режим доступа: <https://angularjs.org/>, свободный (Дата последнего обращения: 12.05.22)
6. NestJS официальный сайт [Электронный ресурс]: Режим доступа: <https://nestjs.com/>, свободный (Дата последнего обращения: 12.05.22)
7. Node.js официальный сайт [Электронный ресурс]: Режим доступа: <https://nodejs.org/en/about/>, свободный (Дата последнего обращения: 10.05.22)
8. Microsoft VS Code официальный сайт [Электронный ресурс]: Режим доступа: <https://code.visualstudio.com/>, свободный (Дата последнего обращения: 05.05.22)

9. Статья на сайте Web-Proger [Электронный ресурс]: Режим доступа: <http://web.spt42.ru/index.php/chto-takoe-typescript/>, свободный (Дата последнего обращения: 12.05.22)

10. MDN Web Docs Java Script [Электронный ресурс]: Режим доступа: <https://developer.mozilla.org/ru/docs/Web/JavaScript>, свободный (Дата последнего обращения: 12.05.22)

11. Статья на сайте web-creator «PostgreSQL – объектно-реляционная система управления базами данных» [Электронный ресурс]: Режим доступа: <https://web-creator.ru/articles/postgresql/>, свободный (Дата последнего обращения: 10.05.22)

12. Что такое PostgreSQL? [Электронный ресурс]: Режим доступа: http://www.sai.msu.su/~megera/postgres/talks/what_is_postgresql.html/, свободный (Дата последнего обращения: 10.05.22)

13. NestJS – тот самый, настоящий бэкенд на nodejs [Электронный ресурс]: Режим доступа: <https://habr.com/ru/post/439434/>, свободный (Дата последнего обращения: 15.05.22)

14. Паттерн MVC [Электронный ресурс]: Режим доступа: <https://metanit.com/web/nodejs/7.1.php/>, свободный (Дата последнего обращения 17.04.22)

15. Диаграмма вариантов использования [Электронный ресурс]: Режим доступа: <https://habr.com/ru/post/566218/>, свободный (Дата последнего обращения: 25.04.2022)

16. Телеграмм в цифрах: в каких странах работает мессенджер и как переживает блокировки [Электронный ресурс]: Режим доступа: <https://daily.afisha.ru/infoporn/22599-telegram-v-cifrah-v-kakih-stranah-rabotaet-messendzher-i-kak-perezhivaet-blokiriovki/>, свободный (Дата последнего

обращения: 30.04.2022)

17. Database | NestJS [Электронный ресурс]: Режим доступа: <https://docs.nestjs.com/techniques/database/>, свободный (Дата последнего обращения: 13.05.2022)