|  |  |
| --- | --- |
|  | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования  «Новгородский государственный университет имени Ярослава Мудрого»  ИНСТИТУТ ЭЛЕКТРОННЫХ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ  ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ |

РАЗРАБОТКА ВЕБ ПРИЛОЖЕНИЯ СОИЦАЛЬНАЯ СЕТЬ

Курсовой проект

ПМ.08 ПРОЕКТИРОВАНИЕ ДИЗАЙНА ИНФОРМАЦИОННЫХ СИСТЕМ

Пояснительная записка курсового проекта по специальности

09.02.07 Информационные системы и программирование

|  |
| --- |
| Студент группы 2994  \_\_\_\_\_\_\_\_/ В. В. Левин /  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Руководитель  \_\_\_\_\_\_\_\_\_/ В. Р. Ильин/  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г |

Содержание

[Введение 3](#_Toc165125265)

[1. Общая часть 6](#_Toc165125266)

[1.1 Постановка задачи 6](#_Toc165125267)

[1.2 Обоснование проектных решений 8](#_Toc165125268)

[1.3 Обзор и анализ существующих программных систем 10](#_Toc165125269)

[2 Практическая составляющая работы 12](#_Toc165125270)

[2.1 Информационное моделирование предметной области 12](#_Toc165125271)

[2.2 Реализация 13](#_Toc165125272)

[2.3 Тестирование 16](#_Toc165125273)

[2.4 Руководство оператора 17](#_Toc165125274)

[Заключение 22](#_Toc165125275)

[Список литературы 23](#_Toc165125276)

[Приложения А 24](#_Toc165125277)

[Приложение Б 29](#_Toc165125278)

[Приложение В 35](#_Toc165125279)

[Приложение Г 37](#_Toc165125280)

# Введение

В мире, где каждая минута ценится как драгоценный ресурс, наш образ жизни становится все более динамичным и связанным с цифровой сферой. В современном обществе, на фоне постоянной суеты и ограничений, налагаемых пандемией, сохранение связи с близкими и друзьями становится особенно важным. Социальные сети становятся неотъемлемой частью нашего повседневного общения, предоставляя нам возможность быть на связи в любое время и в любом месте.

Социальные сети продолжают развиваться, ведь каждый день что-то меняется. Нужно только успевать приспосабливаться.

Теперь людям гораздо удобнее разговаривать визуальными символами: это быстрее, эмоциональнее и понятнее.

В таких условиях появляется необходимость в инновационных и удобных платформах для общения, которые учитывают наши потребности в скорости, безопасности и эффективности. “Rustogram” призвана стать ответом на эти потребности, предлагая пользователям уникальный и привлекательный опыт общения в онлайн-среде.

Наша социальная сеть объединяет в себе передовые технологии и новаторский дизайн, чтобы создать место, где каждый момент становится ценным и запоминающимся. “Rustogram” не просто предлагает возможность общения, но и стимулирует творческое самовыражение через мультимедийный контент, текстовые сообщения и эмоциональные реакции.

Вынужденный переход миллионов людей по всему миру в онлайн-пространство во время пандемии существенно повлиял не только на социальные взаимодействия, но и на бизнес. Исследование компании Deloitte показывает, что 77% представителей малого и среднего бизнеса начали или расширили использование цифровых инструментов в период пандемии.

Для “Rustogram” это открывает новые возможности. Взаимодействие в мессенджерах и социальных сетях для бизнеса обладает рядом преимуществ по сравнению с традиционными средствами коммуникации, такими как электронная почта или телефон. Эти платформы обеспечивают быструю и удобную коммуникацию с клиентами, позволяют создавать сообщества и группы для обсуждения продуктов и услуг, а также предоставляют возможность проведения рекламных кампаний и акций прямо внутри приложения.

“Rustogram” стремится стать не только местом для общения и развлечений, но и эффективным инструментом для бизнеса. Мы уверены, что интеграция функционала для бизнеса на нашей платформе поможет пользователям создавать и развивать свои проекты, предприятия и сообщества, делая "Rustogram" неотъемлемой частью их онлайн-присутствия и социальных взаимодействий.

На протяжении этой курсовой работы мы будем исследовать и разрабатывать функционал “Rustogram”, стремясь создать социальную сеть, которая не только соответствует современным требованиям к общению, но и предлагает новые возможности для выражения индивидуальности и социальной связи. Мы рассмотрим ключевые аспекты разработки, начиная с создания привлекательного пользовательского интерфейса и заканчивая реализацией безопасной системы обмена сообщениями. Цель “Rustogram” - не просто предложить новый способ общения, но и вдохновить пользователей на создание связей, выражение идеи и деление моментов своей жизни в непринужденной и вдохновляющей обстановке.

Основные задачи, которые мы ставим перед собой в ходе разработки “Rustogram”, включают:

* создание привлекательного и интуитивно понятного пользовательского интерфейса, который будет обеспечивать удобство и эффективность использования приложения,
* разработать системы аутентификации и авторизации пользователей, чтобы гарантировать безопасность и конфиденциальность персональных данных,
* имплементация функциональности обмена сообщениями, позволяющей пользователям обмениваться текстовыми сообщениями.

В ходе разработки “Rustogram” мы будем использовать современные технологии веб-разработки, такие как HTML, CSS, JavaScript, PHP, библиотеку React, а так же фреймворк Laravel, чтобы обеспечивать высокую производительность и отзывчивость приложения.

# 1. Общая часть

# 1.1 Постановка задачи

1.1.1 Обоснование необходимости разработки

В соответствии с заданием на курсовой, требуется разработать веб-приложение “Rustogram” для улучшения процесса коммуникации и обмена информацией. Существующие социальные сети имеют некоторые ограничения в связи с их региональной особенностью и не полностью удовлетворяют потребности пользователей. “Rustogram” будет предлагать некоторые возможности и решения в области коммуникации.

Целевая аудитория приложения включает широкий круг пользователей, начиная от обычных пользователей, желающих улучшить свой опыт общения, до бизнес-пользователей.

1.1.2 Технико-математическое описание задачи

Для разработки веб-приложения “Rustogram” будет использован набор современных технологий.

Протокол передачи данных HTTP (Hypertext Transfer Protocol) будет использоваться для обмена информацией между клиентом и сервером. Язык разметки гипертекста HTML (Hypertext Markup Language) будет применяться для структрирования содержимого веб-страницы.

Для реализации интерактивности и функциональности приложения будут использованы языки программирования JavaScript и PHP. JavaScript является распространенным языком программирования, который обеспечивает возможность создания динамических и интерактивных элементов пользовательского интерфейса, обработки событий и выполнения ассинхронных запросов к серверу. PHP так же является распространенным языком программирования, который обеспечивает возможность создания серверной логики проекта.

Для стилизации и оформления пользовательского интерфейса будет использован фреймворк Tailwind, который позволяет определить внешний вид элементов веб-страницы.

В качестве библиотеки для разработки пользовательской части веб-приложения будет использован React. React является современной библиотекой, предназначенной для создания эффективных и масштабируемых интерактивных пользовательских интерфейсов. Он обеспечивает удобную работу с компонентами, автоматическую оптимизацию кода и быструю загрузку приложенияю

В качестве фреймворка для разработки серверной части веб приложения будет использован Laravel. Laravel является современным фреймворком, предназначенным для эффективных и масштабируемых решений серверной части приложения. Он обеспечивает удобную работу с базой данных за счет своей орм, а так же быстроту обработки данных на стороне сервера.

В результате использования данных технологий и фреймворка “Rustogram” будет обладать высокой производительностью, отзывчивым пользовательским интерфейсом и современным дизайном.

1.1.3 Характеристики бизнес-процессов

Анализ предметной области позволяет выделить группы пользователей, такие как обычные пользователи, бизнес-пользователи и учебные заведения. Каждая группа пользователей имеет свои особенности и требования к функционалу приложения. Например, бизнес-пользователи могут ожидать возможности управления заказами и клиентской поддержки, в то время как учебные заведения могут требовать функции группового общения и совместной работы над проектами.

1.1.4 Требования к программе

Для работы "Communique app" требуется совместимость с современными операционными системами и браузерами. Веб-приложение должно включать базы данных для хранения данных пользователей и сообщений, а также высокую пропускную способность хостинга для обеспечения быстрого доступа к приложению.

# Обоснование проектных решений

1.2.1 Обоснование выбора языков программирования

При разработке “Rustogram” был выбран комбинированный набор языков программирования, включающий HTML, JavaScript, PHP и SQL. Вот основные причины выбора этих языков:

* HTML (HyperText Markup Language): HTML является основным языком разметки для веб-приложений. Он обеспечивает структурирование и отображение содержимого приложения. HTML позволяет определить различные элементы страницы, такие как заголовки, параграфы, списки, изображения и другие компоненты, что делает его неотъемлемой частью веб-разработки;
* JavaScript является одним из самых распространенных языков программирования для разработки веб-приложений. Он предоставляет возможности для создания динамического и интерактивного пользовательского интерфейса, обработки событий, выполнения асинхронных запросов к серверу и других функциональностей. JavaScript также широко поддерживается веб-браузерами, что делает его идеальным выбором для разработки веб-приложений;
* PHP (Hypertext Preprocessor): PHP - это серверный язык программирования, который широко используется для создания динамических веб-приложений. Он позволяет встраивать PHP-код непосредственно в HTML-страницы, что обеспечивает возможность создания интерактивных и персонализированных веб-приложений. PHP обеспечивает взаимодействие с базами данных, обработку форм, генерацию динамических контентов и другие функциональности, делая его важным инструментом веб-разработки;
* SQL (Structured Query Language) является языком программирования для работы с реляционными базами данных. В "Communique app" SQL используется для хранения и управления данными, связанными с пользовательскими профилями, сообщениями и другими сущностями приложения. SQL обеспечивает эффективное и структурированное взаимодействие с базами данных, позволяя выполнять запросы, изменять данные и обеспечивать целостность данных;

Комбинация HTML, JavaScript, PHP и SQL обеспечивает полный набор инструментов для разработки "Communique app". HTML и JavaScript обеспечивают интерактивность и функциональность пользовательского интерфейса, PHP повышает качество и удобство разработки серверной стороны, а SQL обеспечивает надежное хранение и управление данными приложения. Этот набор языков программирования позволяет создавать мощное и эффективное веб приложение для коммуникации и обмена сообщениями.

1.2.2 Инструментальные средства

В процессе разработки “Rustogram” будет использован IDE PHPStorm, который предоставляет удобную среду разработки для JavaScript/PHP и HTML/CSS. Также могут быть использованы программы для контроля версий (например, Git),база данных MySQL для хранения данных приложения.

1.2.3 Обоснование выбора среды программирования

IDE PHPStorm был выбран в качестве среды разработки, так как он предлагает широкий набор инструментов для разработки веб-приложений на различных языках программирования. PHPStorm обладает удобным интерфейсом, интегрированным отладчиком, поддержкой автодополнения кода и другими полезными функциями, которые значительно упрощают процесс разработки.

1.2.4 Информационное обеспечение

Для работы с "Communique app" использованы сторонние программные средства, такие как Git для контроля версий кода и базы данных MySQL для хранения информации о пользователях, постах и других данных приложения. Кроме того, для создания документации и презентаций использованы программы, такие как Microsoft Word и PowerPoint.

# 1.3 Обзор и анализ существующих программных систем

В рамках курсового проекта был проведен сбор и анализ существующих программных систем, которые предлагают функциональность коммуникации, развлечений и обмена сообщениями, схожую с “Rustogram”. Были выявлены следующие аналоги, которые были подвергнуты сравнительному анализу:

* Instagram является одной из самых популярных социальных сетей, предоставляющих возможность загрузки постов, историй, видео, обмена сообщениями и другими функциями. Он обладает широкой аудиторией и хорошей интеграцией со смартфонами, что делает его удобным инструментом для коммуникции;
* X ещё одна популярная социальная сеть, которая предлагает весь тот же функционал, что и у Instagram за исключением того, что в X также можно публиковать посты без изображений. Он известен своей политической тематикой, а так же фокусировкой на безопасность пользователя;
* ВКонтакте предоставляет широкий спектр функциональности, включая загрузку медиа-контента, обмен сообщениями, создание групп и многое другое. Обладает большой популярностью в русскоязычном сегменте интернета;
* Facebook - одна из крупнейших и наиболее популярных социальных сетей, предоставляющая широкий спектр функциональности, включая создание профилей, загрузку фотографий и видео, обмен сообщениями, создание событий и групп, а также рекламные возможности для бизнеса;
* Snapchat - платформа для обмена мгновенными сообщениями и мультимедийным контентом, который автоматически удаляется через определенное время после просмотра. Snapchat известен своими фильтрами для фотографий и видео, а также функциями историй, чатов и создания собственного контента.

Проведенный анализ позволил выделить достоинства и недостатки указанных аналогов. Некоторые из них могут обладать ограничениями в функциональности, сложностью использования, отсутствием определенных возможностей или недостаточной безопасностью. “Rustogram” нацелена на преодоление этих ограничений и предлагает уникальные особенности, специально разработанные для обеспечения более эффективной и удобной коммуникации.

Таким образом, проведенный обзор существующих программных систем позволил определить преимущества и уникальные особенности, которые предлагает разрабатываемое веб-приложение “Rustogram”.

# 2 Практическая составляющая работы

# 2.1 Информационное моделирование предметной области

Курсовой проект наличие базы данных со следующими таблицами:

* users(id, name, email, password, avataer, bio, birthday, gender, is\_online, last\_online, is\_admin, email\_verified\_ar, created\_at, updated\_at);
* posts(id, user\_id, image\_path, video\_path, description, created\_at, updated\_at);
* posts\_comments(id, user\_id, post\_id, content, created\_at, updated\_at);
* posts\_likes(id, user\_id, post\_id, created\_at, updated\_at);
* chat(id, name, created\_at, updated\_at);
* messages(id, chat\_id, owner\_id, user\_id, message, created\_at, updated\_at);
* users\_chats(id, chat\_id, owner\_id, user\_id, created\_at, updated\_at);
* hashtags(id, name, user\_id, post\_id, created\_at, updated\_at);
* more\_posts(id, image\_path, video\_path, post\_id, created\_at, updated\_at);
* reels(id, video\_path, description, user\_id, created\_at, updated\_at);
* reels\_likes(id, user\_id, reels\_id, created\_at, updated\_at);
* reels\_comments(id, user\_id, reels\_id, content, created\_at, updated\_at);
* stories(id, user\_id, image\_path, video\_path, description, created at, updated\_at);
* looked\_stories(id, user\_id, story\_id, is\_looked, created\_at, updated\_at);
* subscriptions(id, user\_id, owner\_id, created\_at, updated\_at);

Диаграмму классов, а также диграммы IDEF0 и IDEF1X можно посмотреть на картинках А.3 – А.5.

# 2.2 Реализация

2.2.1 Обоснование и описание выбора состава технических и программных средств

Для разработки проекта “Rustogram” был осуществлен тщательный выбор состава технических и программных средств, учитывая требования проекта к функциональности, производительности и удобству разработки.

* React был выбран в качестве основной библиотеки для разработки пользовательского интерфейса. Эта библиотека для JavaScript позволяет разрабатывать динамические и интерактивные веб-приложения, используя компонентный подход. Его преимущества включают в себя повторное использование компонентов, простоту управления состоянием и удобство отладки. React также обладает обширным сообществом разработчиков и большим количеством сторонних библиотек и инструментов, что делает его мощным инструментом для разработки современных веб-приложений;
* Для создания серверной части проекта был выбран фреймворк Laravel. Laravel - это PHP фреймворк, который обеспечивает мощные инструменты для разработки веб-приложений, включая работу с базами данных, маршрутизацию, аутентификацию и многое другое. Его преимущества включают в себя простоту и удобство в использовании, высокую производительность и расширяемость;
* Для стилизации пользовательского интерфейса был выбран Tailwind CSS. Этот инструмент предоставляет набор готовых к использованию CSS-классов, которые позволяют быстро и гибко создавать стили для элементов интерфейса. Tailwind CSS основан на подходе “utility-first”, что делает его особенно удобным для быстрой разработки и поддержки веб-приложений.

Дополнительно, были использованы следующие библиотеки для React:

* react-slick: для создания каруселей и слайдеров;
* framer-motion: для добавления анимаций и переходов между компонентами;
* react-custom-scrollbars: для кастомизации скроллбаров в приложении;
* react-lottie-player: для встраивания анимаций в формате Lottie в приложение;
* axios: для выполнения HTTP-запросов к серверу;
* slick-carousel: для добавления дополнительных возможностей каруселей и слайдеров;
* tailwind-merge: для более гибкой работы с классами Tailwind CSS;
* video-react: для встраивания видео в приложение с удобным интерфейсом и возможностями управления.

Этот комплекс технических и программных средств был выбран с учетом требований к проекту "Rustogram" и цели обеспечить высокое качество, производительность и удобство использования для конечных пользователей.

2.2.2 Практическая реализация алгоритмов

Рассмотрим практическую реализацию алгоритмов на примере функции createChat ответственная за создание чата между пользователями. Эта функция, помимо своей прямой задачи, представляет собой уникальный пример применения алгоритмов в реальном веб-приложении. Изначально происходит проверка пользователя, которому предназначен запрос на создание чата. Этот этап обеспечивает корректность ввода данных и гарантирует, что создание чата возможно только для существующего пользователя. Затем происходит поиск информации о втором пользователе на основе переданного идентификатора. Этот шаг необходим для определение, с кем будет создан чат. После этого происходит проверка существующего чата, на этом этапе функция проверяет наличие уже существующего чата между текущим пользователем и найденным пользователем. Это позволяет избежать создание дублирующихся чатов. Затем происходит создание нового чата, если чата между пользователями не существует, функция создает новый и присваивает ему уникальное имя, отображающее имена обоих участников. В конце происходит запись в базу данных информации, связывающую пользователей с новым чатом. Это обеспечивает последующую возможность взаимодействия пользователей в рамках созданного чата. Блок-схему функции можно посмотреть ниже на рисунке А.1.

Функция getAllMessagesFromChat.

На вход функции передается три переменных:

* request,
* chatId,
* userId.

Request отвечает за содержание запроса с пользовательской стороны, в ней храниться пользователь, который передаёт этот запрос, а так-же информация о дате и времени. В переменной chatId храниться уникальный идентификатор чата, к которому обращён этот запрос. А в переменной userId хряниться уникальный идентификатор второго пользователя чата. После того как функция получила все необходимы параметры, происходит взятие текущего пользователя из переменной request, это нужно для дальнейшей выборки чата, чтобы исключить возможности, что чат не будет найден. Затем происходит выборка второго пользователя чата. После выборки пользователей происходит выборка чата по двум уникальным идентификаторам пользователей. Если чат не будет найден, то произойдет выход из функции с ошибкой. Тоже самое произойдет если и один из пользователей не будет найден. В противном случае произойдет выборка всех сообщений из текущего чата и отправка их на пользовательскую сторону. Блок-схему функции можно посмотреть ниже на рисунке А.2.

2.2.3 Реализация интерфейса

Интерфейс был разработан с использованием известной библиотеки React. За основу была взята структура страниц, которые в проекте называются представлениями, компонентов, а так же медиа файлов. Были использованы библиотеки такие как:

* react-slick: для создания каруселей и слайдеров;
* framer-motion: для добавления анимаций и переходов между компонентами;
* react-custom-scrollbars: для кастомизации скроллбаров в приложении;
* react-lottie-player: для встраивания анимаций в формате Lottie в приложение;
* axios: для выполнения HTTP-запросов к серверу;
* slick-carousel: для добавления дополнительных возможностей каруселей и слайдеров;
* tailwind-merge: для более гибкой работы с классами Tailwind CSS;
* video-react: для встраивания видео в приложение с удобным интерфейсом и возможностями управления. Пользовательский интерфейс можно увидеть на рисунках Б.1-Б.10.

2.2.4 Реализация серверной части приложения

Для разработки серверной части приложения был выбран PHP фреймворк Laravel, который обеспечивает удобство и эффективность веб-разработки. В ходе работы был использован принцип REST (Representational State Transfer) для построения архитектуры веб-сервисов, обеспечивая гибкость, масштабируемость и удобство взаимодействия с клиентской частью приложения. Laravel предоставляет мощные инструменты для создания RESTful API, что позволяет управлять ресурсами приложения с помощью стандартных HTTP методов, таких как GET, POST, PUT и DELETE. Это обеспечивает прозрачное взаимодействие между клиентом и сервером, а также упрощает разработку и поддержку кода. Использование Laravel в сочетании с принципом REST позволило создать надёжную и эффективную серверную часть приложения, готовую к масштабированию и обеспечивающую высокую производительность.

# 2.3 Тестирование

Тестирование было осуществлено путём создания тест кейсов, которые можно увидеть в приложении Г.5-Г.10, а также средствами PHPUnit, с результатами выполнения тестов можно ознакомиться на рисунках Г.4, с кодом функций тестирования можно ознакомиться на рисунках Г.1-Г.3.

# 2.4 Руководство оператора

2.4.1 Назначение программы

Социальная сеть была реализована для коммуникации людей разных возрастов и находящихся на разных расстояниях, путём передачи друг-другу текстовых сообщениях в окне чата, а также созданием постов.

Целевая аудитория приложения люди разных возрастов и разных профессий.

2.4.2 Условия выполнения

Для работы социальной сети необходим персональный компьютер с установленным (любым) браузером, обновленным до последней доступной версии и поддерживаемый javaScript. Последняя версия браузера Google Chrome - 114.0.5735.199.

2.4.2.1 Работа пользователя

Для того чтобы начать пользоваться социальной сетью пользователь должен открыть установленный браузер на персональном компьютере, после этого перейти в адресную строку и ввести ссылку с доменным именем приложения, затем пользователь попадает на страницу авторизации, где он может авторизоваться в существующий аккаунт в приложении или перейти на страницу регистрации, если пользователь не помнит свой пароль, он может восстановить его на странице восстановления пароля, которую он может найти под формой авторизации. После успешной авторизации (или регистрации с последующей авторизацией), пользователь попадает в свою персональную ленту, где отображаются посты пользователей на которых он подписан (если пользователь только зарегистрировался, то его перенаправит на странице до заполнения информации, где ему будет нужно заполнить свою дату рождения и гендер, а также по желанию добавить аватар и биографию). Путем нажатия на сайдбаре слова поиск, пользователю будет предложено найти посты другого человека, путем нажатия на одноименное поле поиск, он введёт букву v и будут выданы посты пользователей, найденных на букву v. Дальше пользователь может перейти в свой профиль, путём нажатия на сайдбаре слова Профиль, где пользователь сможет добавить свои истории, а так же увидеть своих подписчиков, посты и редактировать свой профиль, путём нажатия на кнопку редактировать и заполнения соответствующих полей. Пользователь также может общаться с другими пользователями в мессенджере, для этого ему нужно перейти на сайдбаре в сообщения и в поле поиск, найти пользователя с кем тот хочет начать общения, когда поиск выдаст результат, нажать на интересующего пользователя и написать своё первое сообщения в соответствующее поле. При общении в чатах, очень часто люди допускают ошибки в своих сообщениях, потому что торопятся, для этого в чате предусмотрено редактирование сообщений, а также их полное удаление, пользователь может нажать правой кнопкой мыши по сообщению, откроется модальная коробка с предложенными действиями. Так же пользователь может удалить чат со своим собеседником, нажав правой кнопкой мыши по сообщению, откроется модальное окно, где будет предложено два действия: удалить и отменить, при нажатии на кнопку удалить, удаляются все сообщения из чата и сам чат, как на стороне пользователя, так и из базы данных в целом. При нажатии кнопки отменить, закрывается модальное окно и можно спокойно продолжать свое общение. Пользователь так же имеет возможность посмотреть короткие видео, а также их создать, которые находятся на вкладке шортсы, где находятся уже созданные короткие видео и кнопка для создания персональных.

После того как пользователь устанет общаться, он может выйти из аккаунта путем нажатия кнопки выйти или просто закрыть вкладку социальной сети, в этом случае все данные пользователя обнуляться как на сервере, так и на стороне клиента, но при этом сам пользователь не удалиться из базы данных, как и чаты, которые он создал вместе с сообщениями.

2.4.2.2  Работа администратора

Администратор в социальной сети может удалять истории, короткие видео, а также посты, это сделано во избежание публикации наготы, насилия, рекламы террористов и так далее.

2.4.2.3 Техника безопасности при работе на компьютере

К работе допускаются лица:

* прошедшие вводный инструктаж по охране труда;
* прошедшие обучение безопасным приемам и методам труда по программе, утвержденной руководителем предприятия (работодателем), разработанной на основе Типовой программы, и прошедшие проверку знаний, в том числе по электробезопасности;
* прошедшие курс обучения на персональном компьютере с использованием конкретного программного обеспечения;
* прошедшие инструктаж по охране труда на конкретном рабочем месте по данной инструкции.
* При эксплуатации персонального компьютера на работника могут оказывать действие следующие опасные и вредные производственные факторы:
* повышенный уровень электромагнитных излучений;
* повышенный уровень статического электричества;
* пониженная ионизация воздуха;
* статические физические перегрузки;
* перенапряжение зрительных анализаторов;

Работник обязан:

* выполнять работу, которая определена его должностной инструкцией;
* содержать в чистоте рабочее место;
* соблюдать режим труда и отдыха в зависимости от продолжительности, вида и категории трудовой деятельности;
* соблюдать меры пожарной безопасности.

Рабочие места с компьютерами должны размещаться таким образом, чтобы расстояние от экрана одного видеомонитора до тыла другого было не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, преимущественно слева.

Оконные проемы в помещениях, где используются персональные компьютеры, должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Рабочая мебель для пользователей компьютерной техникой должна отвечать следующим требованиям:

* высота рабочей поверхности стола должна регулироваться в пределах 680 – 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм;
* рабочий стол должен иметь пространство для ног высотой не менее 600 мм, глубиной на уровне колен не менее 450 мм и на уровне вытянутых ног не менее 650 мм;
* рабочий стул должен быть подъемно - поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также - расстоянию спинки от переднего края сиденья;
* рабочее место должно быть оборудовано подставкой для ног, имеющей ширину, не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов; поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм;
* рабочее место с персональным компьютером должно быть оснащено легко перемещаемым пюпитром для документов.

Для нормализации аэроионного фактора помещений с компьютерами необходимо использовать устройства автоматического регулирования ионного режима воздушной среды.

# Заключение

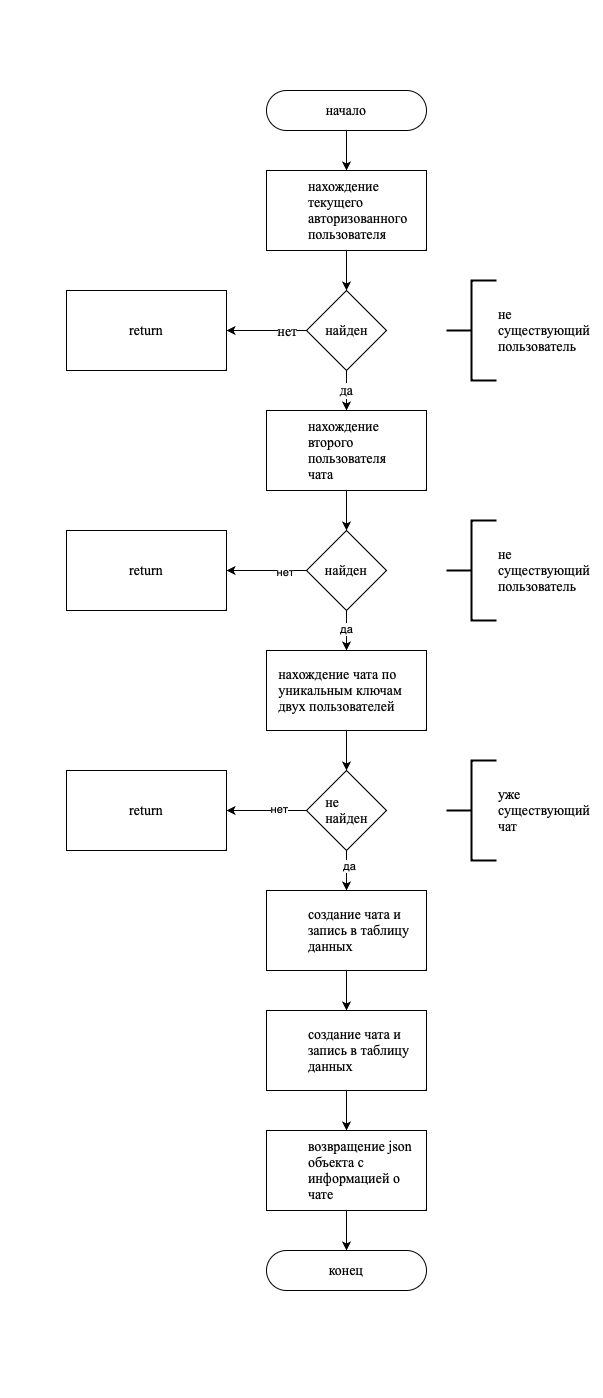
Результатом курсовой работы стала разработка социальной сети, котоая позволяет поддерживать связь с людьми и миром, как на расстоянии, так и удаленно. Социальная сеть позволяет пользователям общаться текстовыми сообщениями, редактировать и удалять их, так же находить людей через поле поиск, в последствии создавать с ними чат (диалог), удалять пользователей из списка чатов, а также редактировать собственный профиль, имя пользователя и картинку пользователя, А так же смотреть посты, истории, короткие видео, профили других пользователей, редактировать и создавать собственные посты, истории, истории, короткие видео.

Курсовая работа выполнена с точки зрения функционала серверной и пользовательской части, но дизайн до конца не перенесён в приложение, а также не до конца сделаны хештеги для постов, в дальнейшем дизайн сайта будет улучшаться, а также дополняться новым функционалом с учётом критики экспертов.

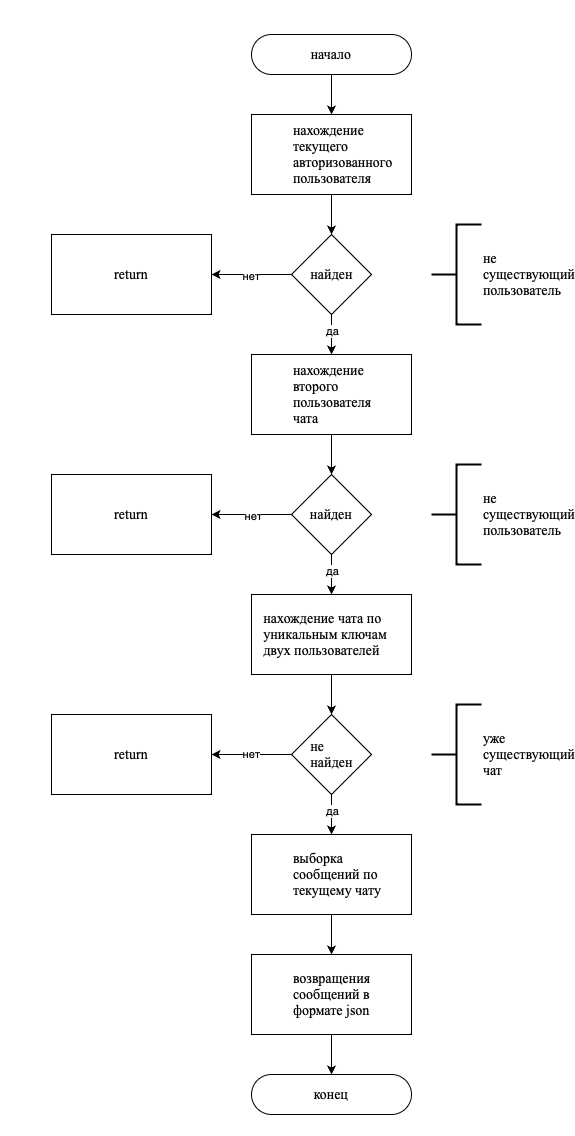
# Список литературы

1. Официальная документация по языку программирования javaScript [электронный ресурс] URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> . Дата обращения к ресурсу: 2 апреля 2024 года.
2. Официальная документация по языку разметки HTML [Электронный ресурс] URL: <https://developer.mozilla.org/ru/docs/Web/HTML> . Дата обращения к ресурсу: 2 апреля 2024 года.
3. Официальная документация по языку программирования PHP [электронный ресурс] URL: [https://www.php.net](https://www.php.net/) . Дата обращения к ресурсу: 2 апреля 2024 года.
4. Официальная документация Laravel [электронный ресурс] URL: [https://laravel.com](https://laravel.com/) . Дата обращения к ресурсу: 2 апреля 2024 года.
5. Официальная документация React [электронный ресурс] URL: <https://react.dev> . Дата обращения к ресурсу: 2 апреля 2024 года.
6. Официальная документация Tailwind [электронный ресурс] URL: [https://tailwindcss.com](https://tailwindcss.com/) . Дата обращения к ресурсу: 2 апреля 2024 года.
7. Официальная документация MySQL [электронный ресурс] URL: <https://dev.mysql.com/doc/> . Дата обращения к ресурсу: 2 апреля 2024 года.

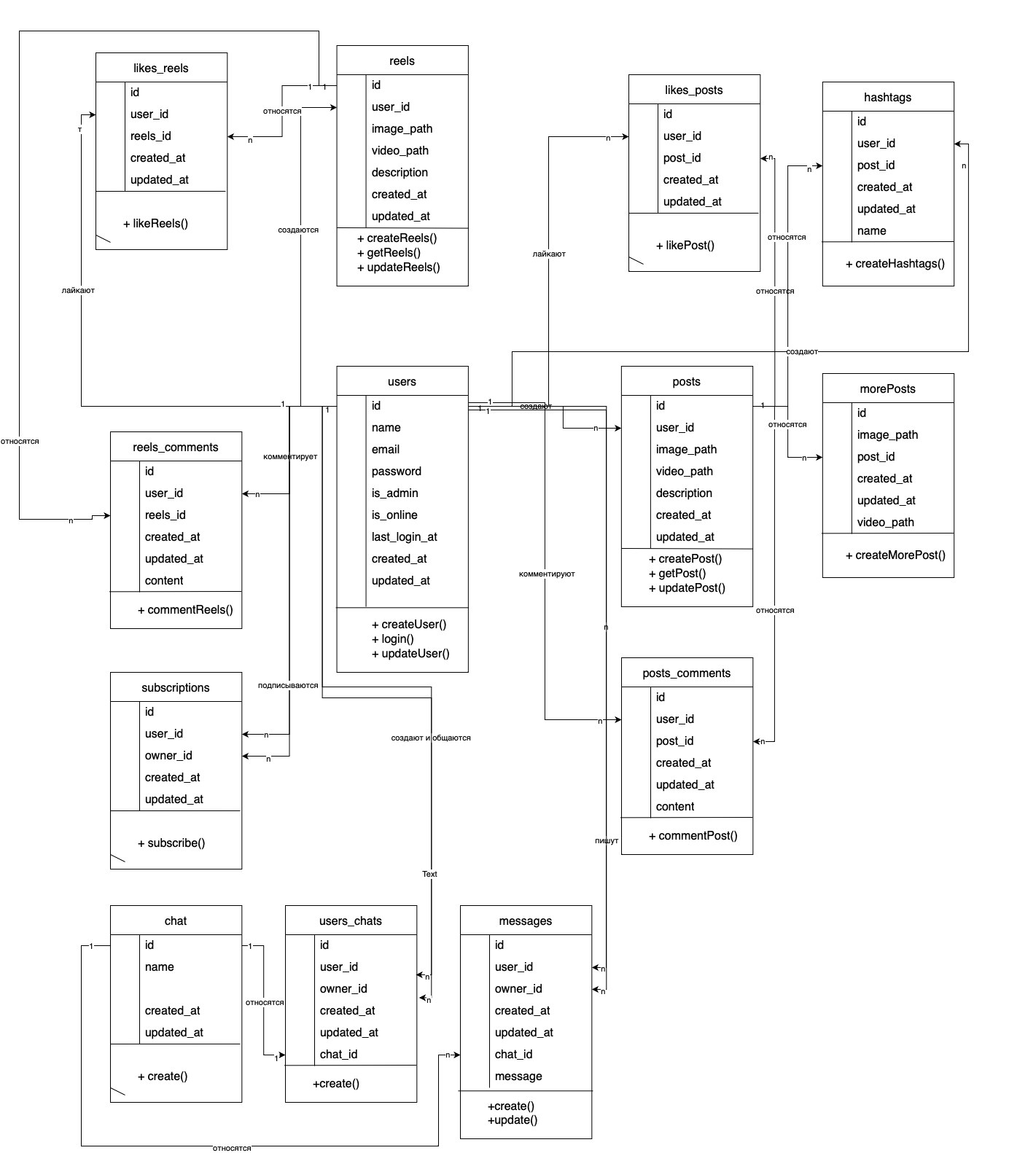
# Приложения А



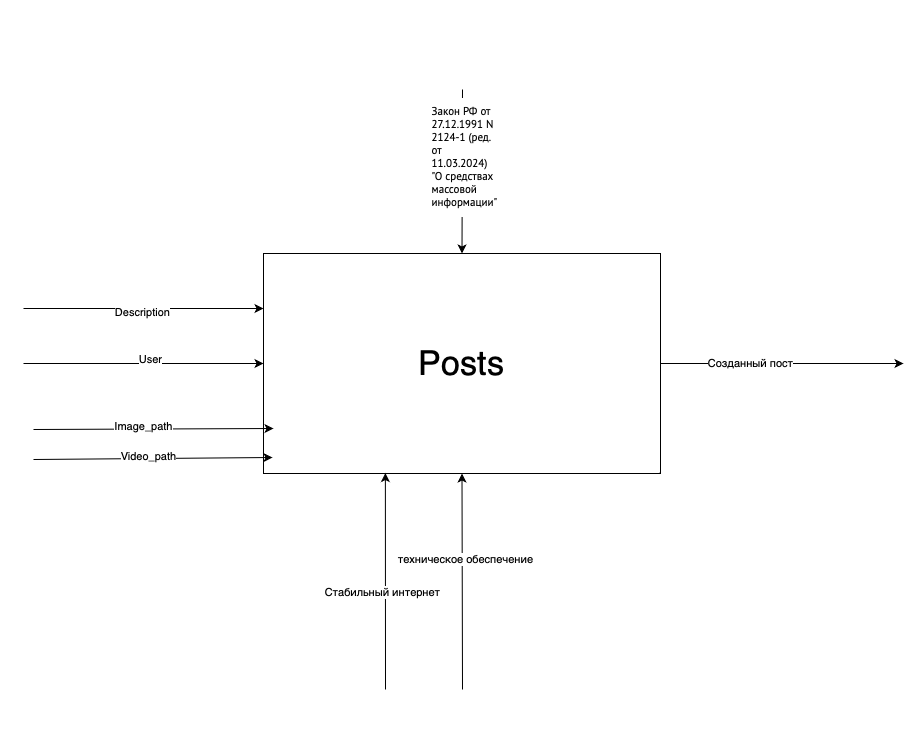
А.1 – Блок-схема функции startChat



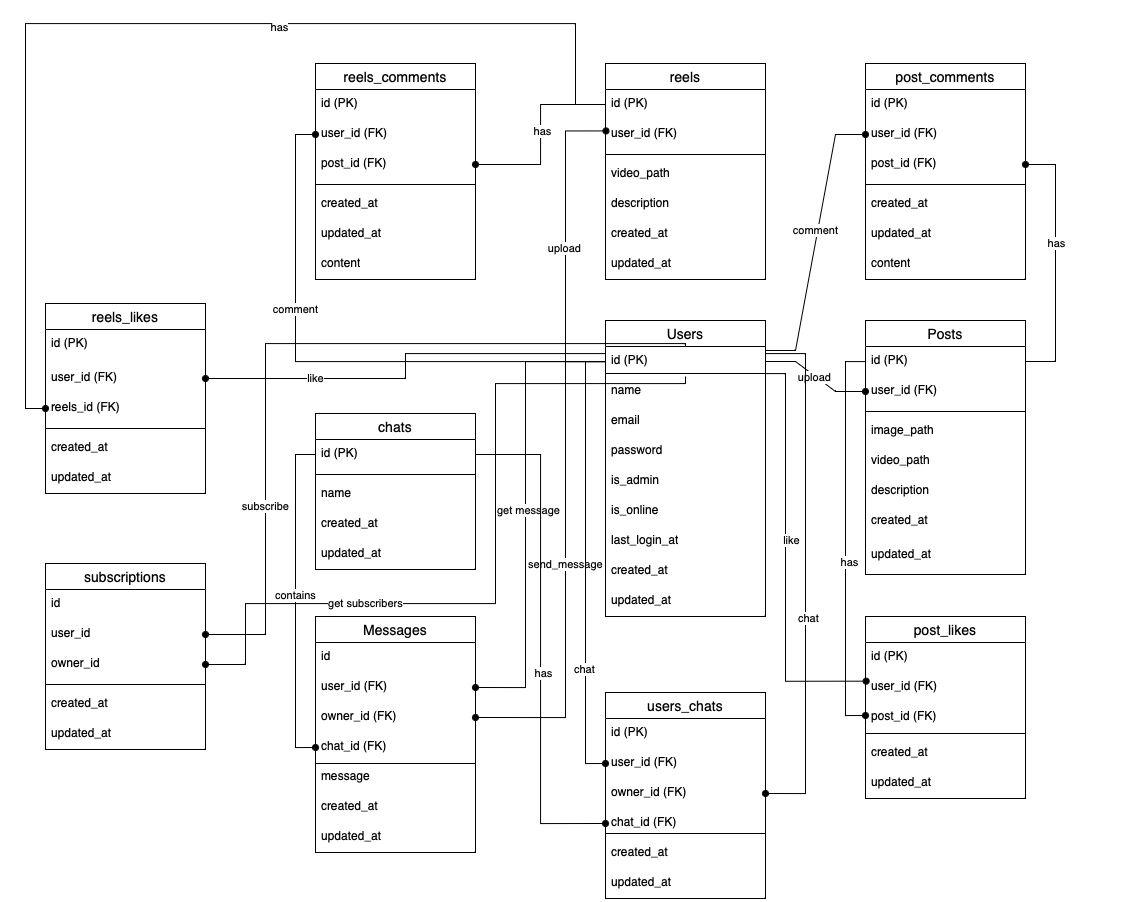
А.2 – Блок-схема функции getAllMessagesFromChat



А.3 – Диаграмма классов

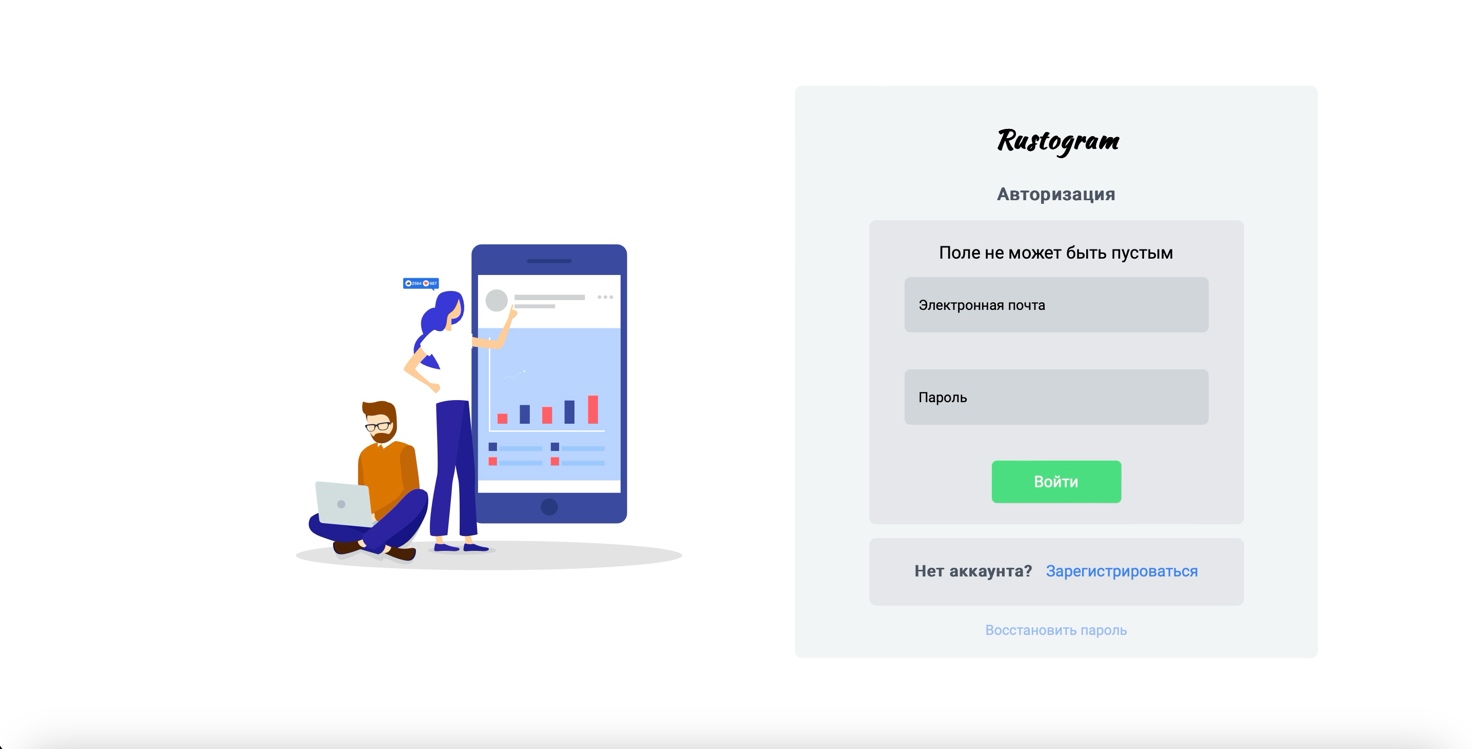


А.4 – IDEF0 диаграмма

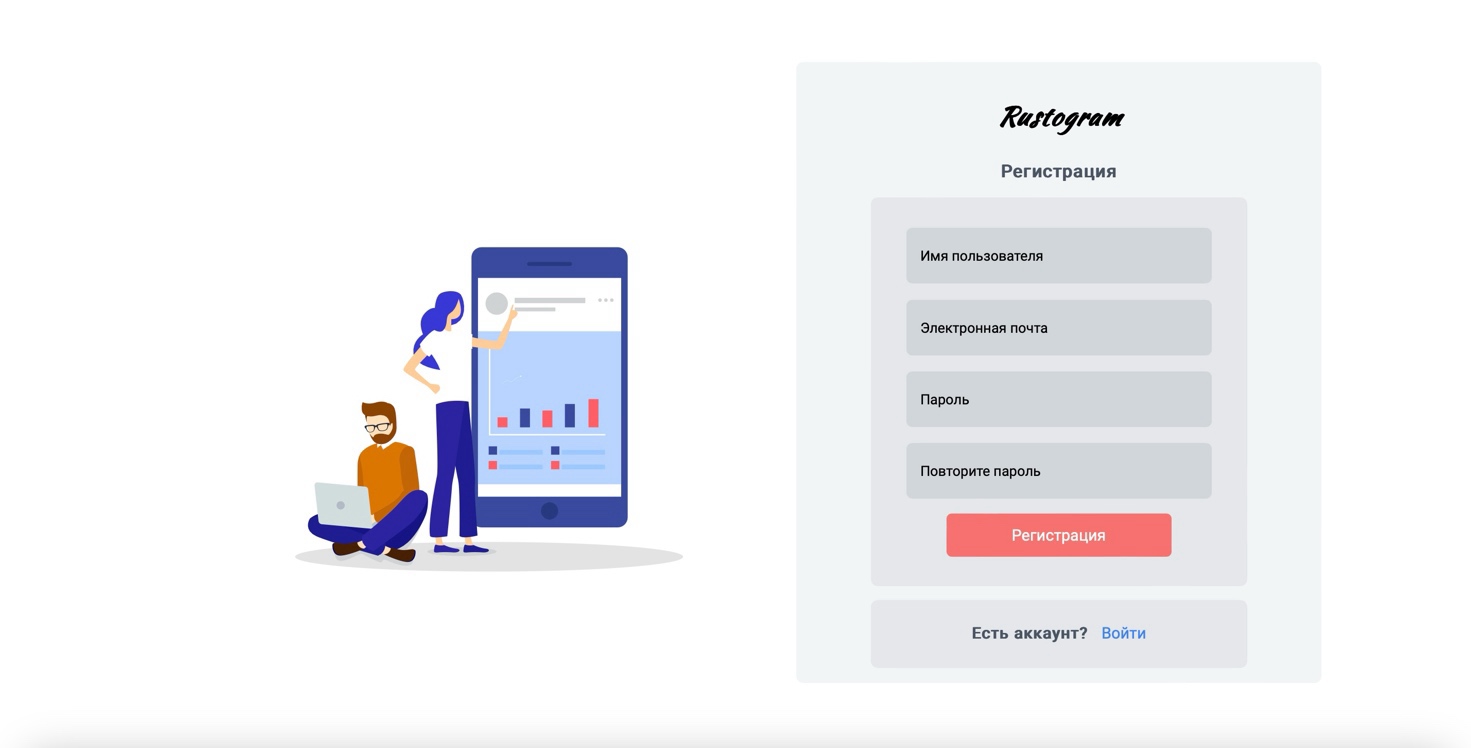


А.5 – IDEF1X диаграмма

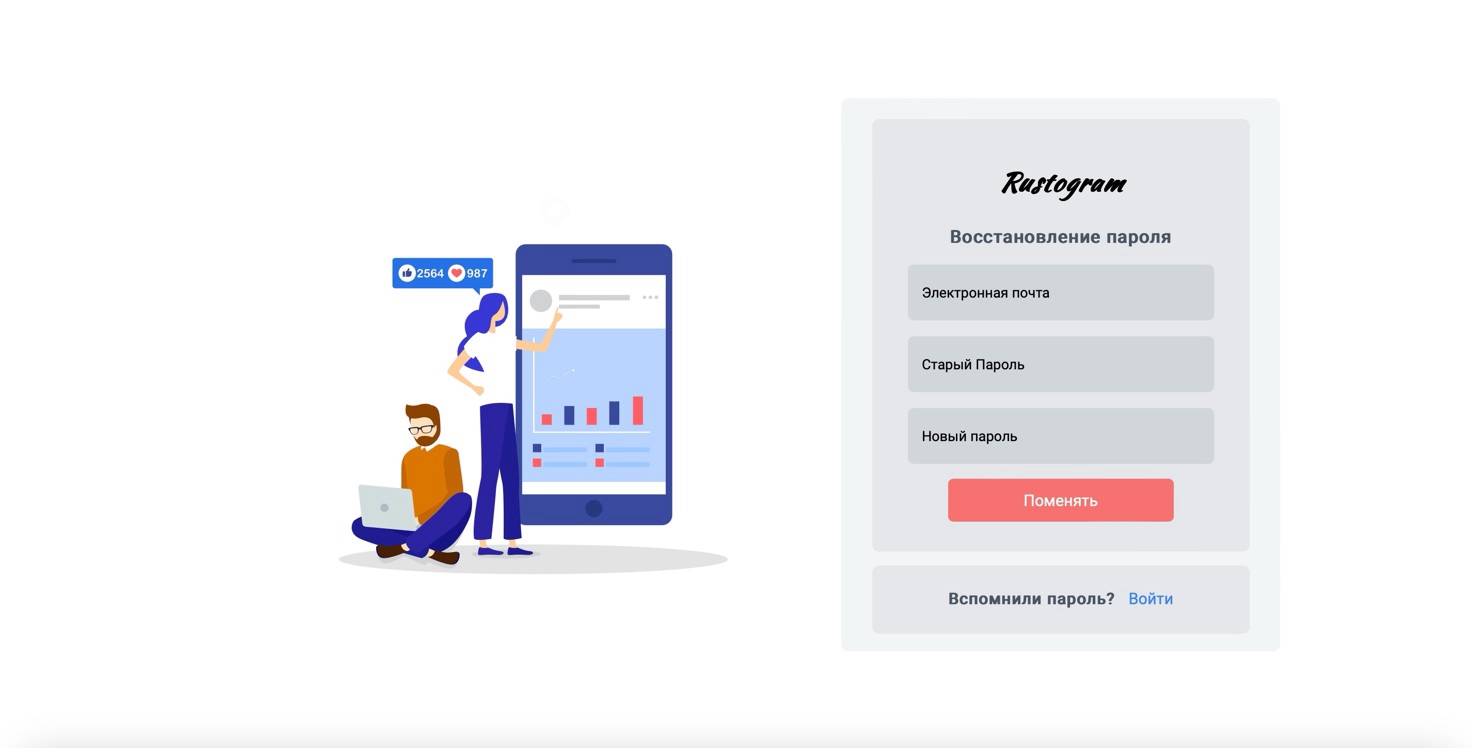
# Приложение Б



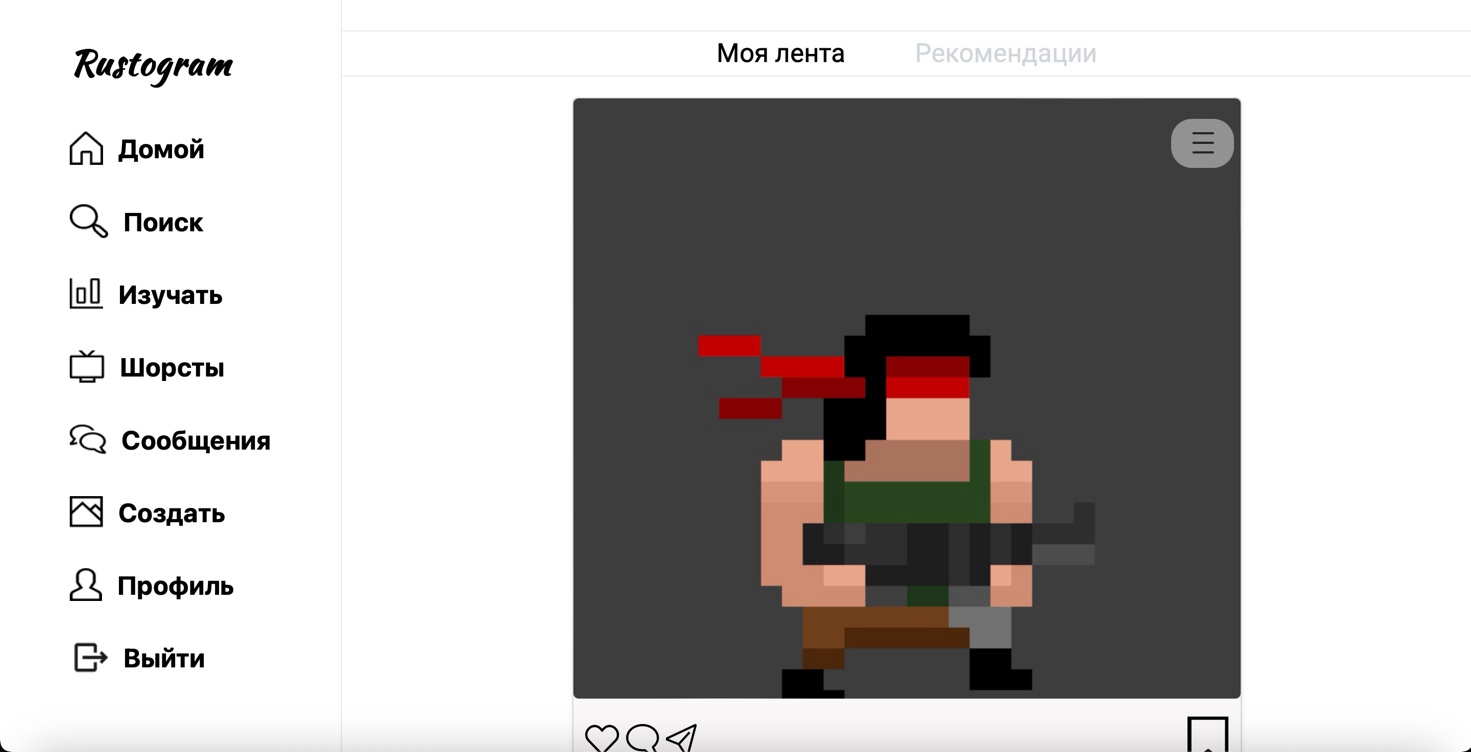
Б.1 – Страница авторизации в приложении



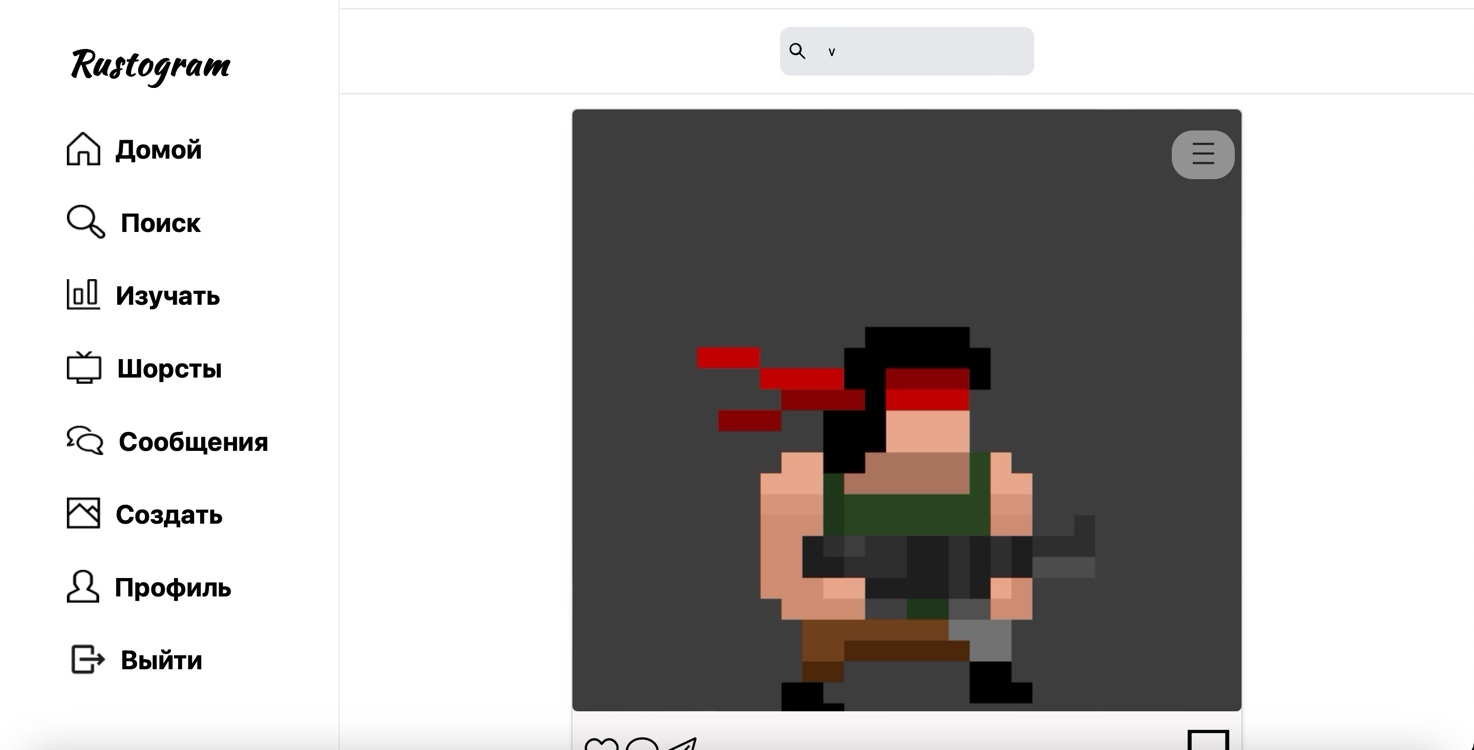
Б.2 – Страница регистрации в приложении



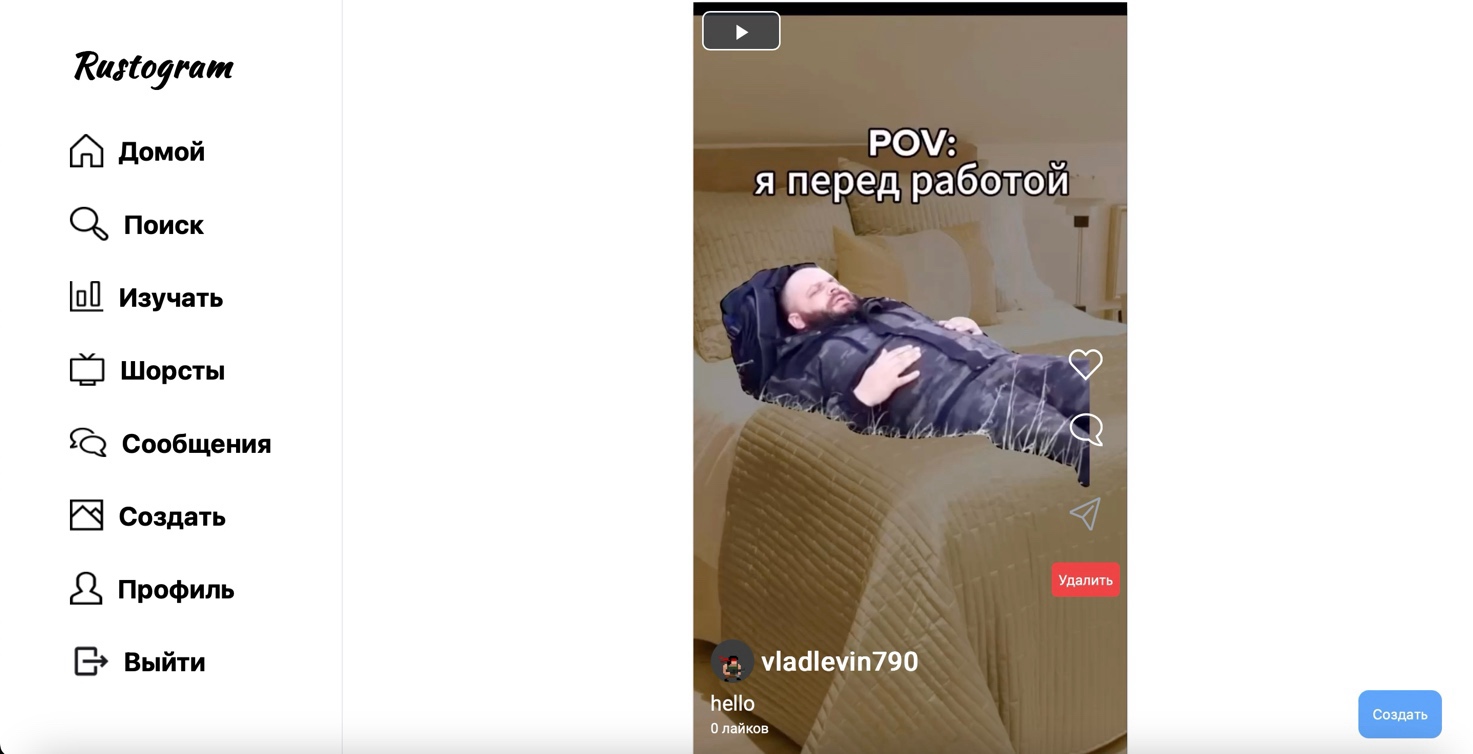
Б.3 – Страница Восстановления пароля в приложении



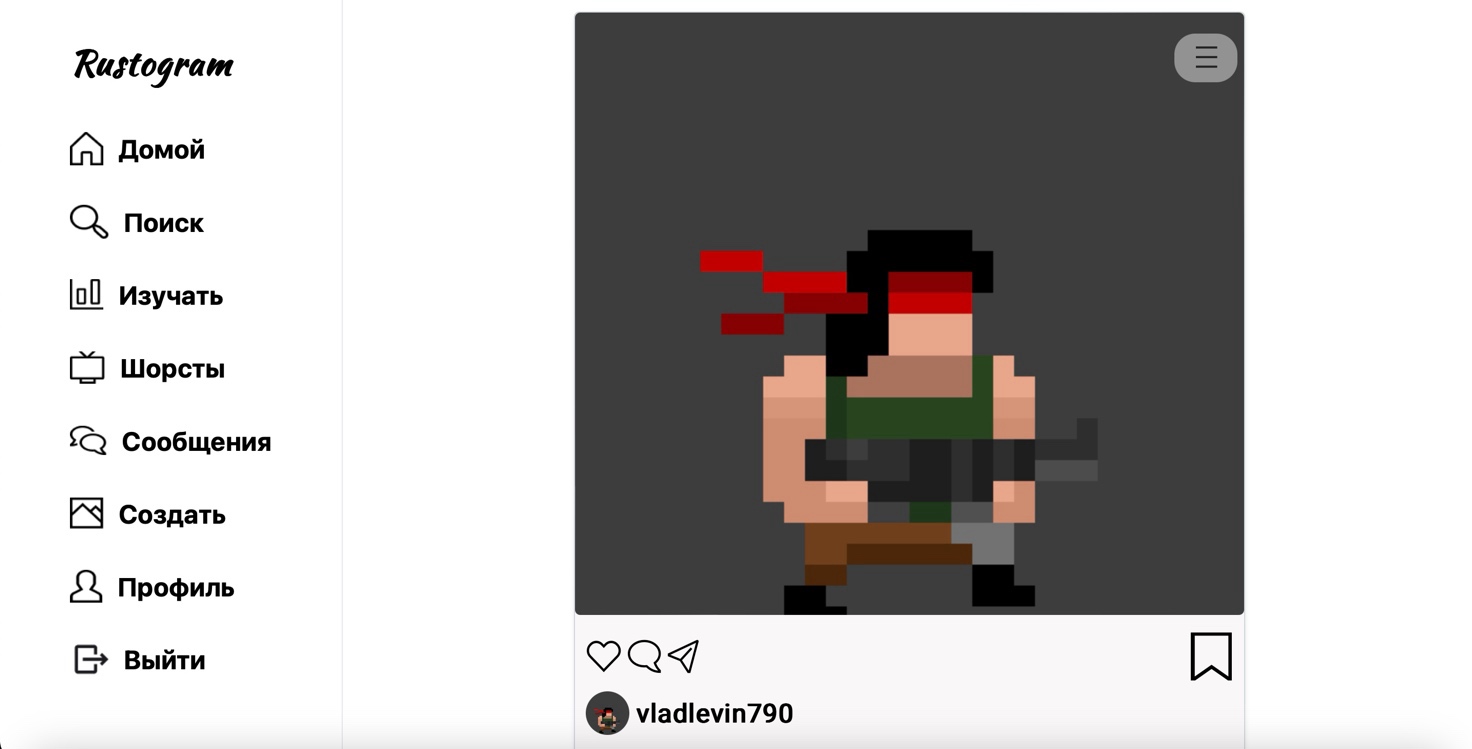
Б.4 – Страница постов в приложении



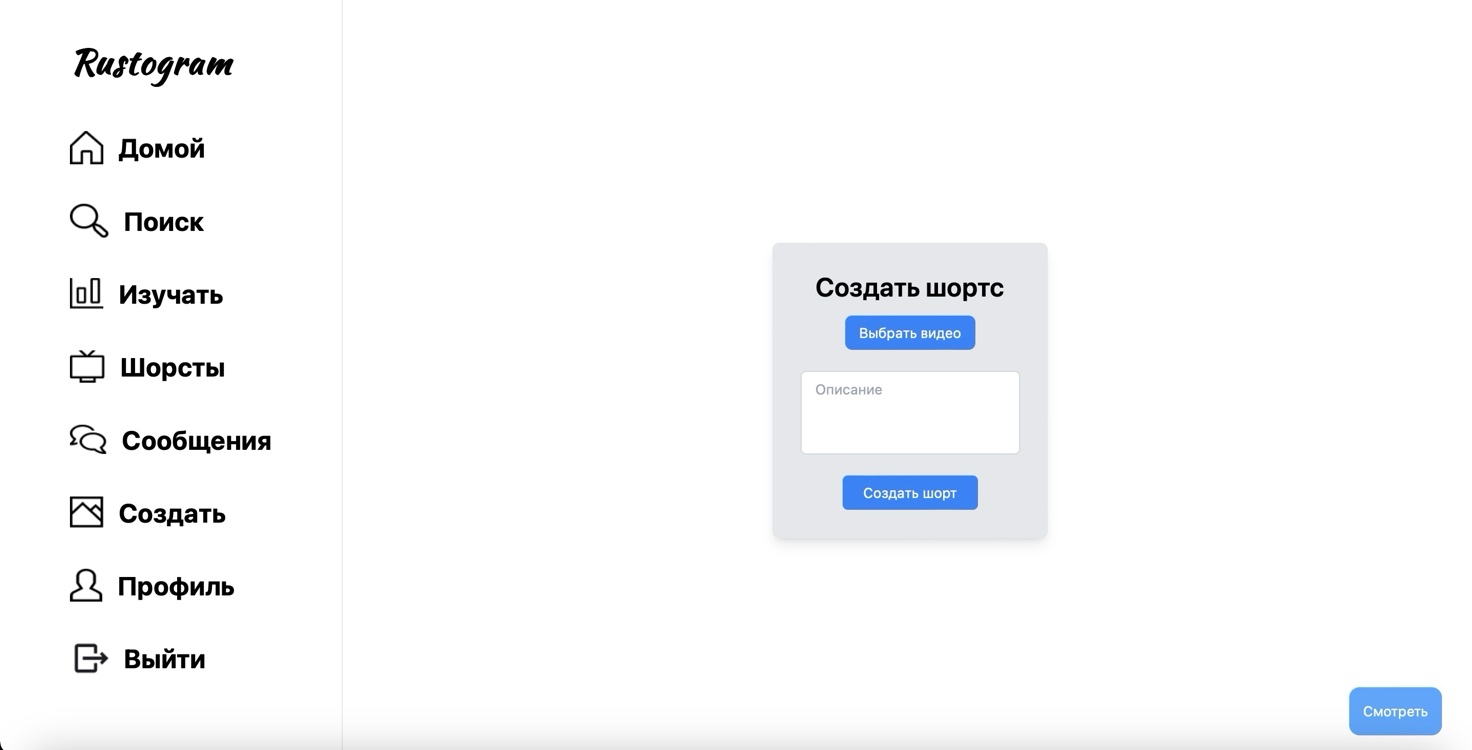
Б.5 – Страница поиска в приложении



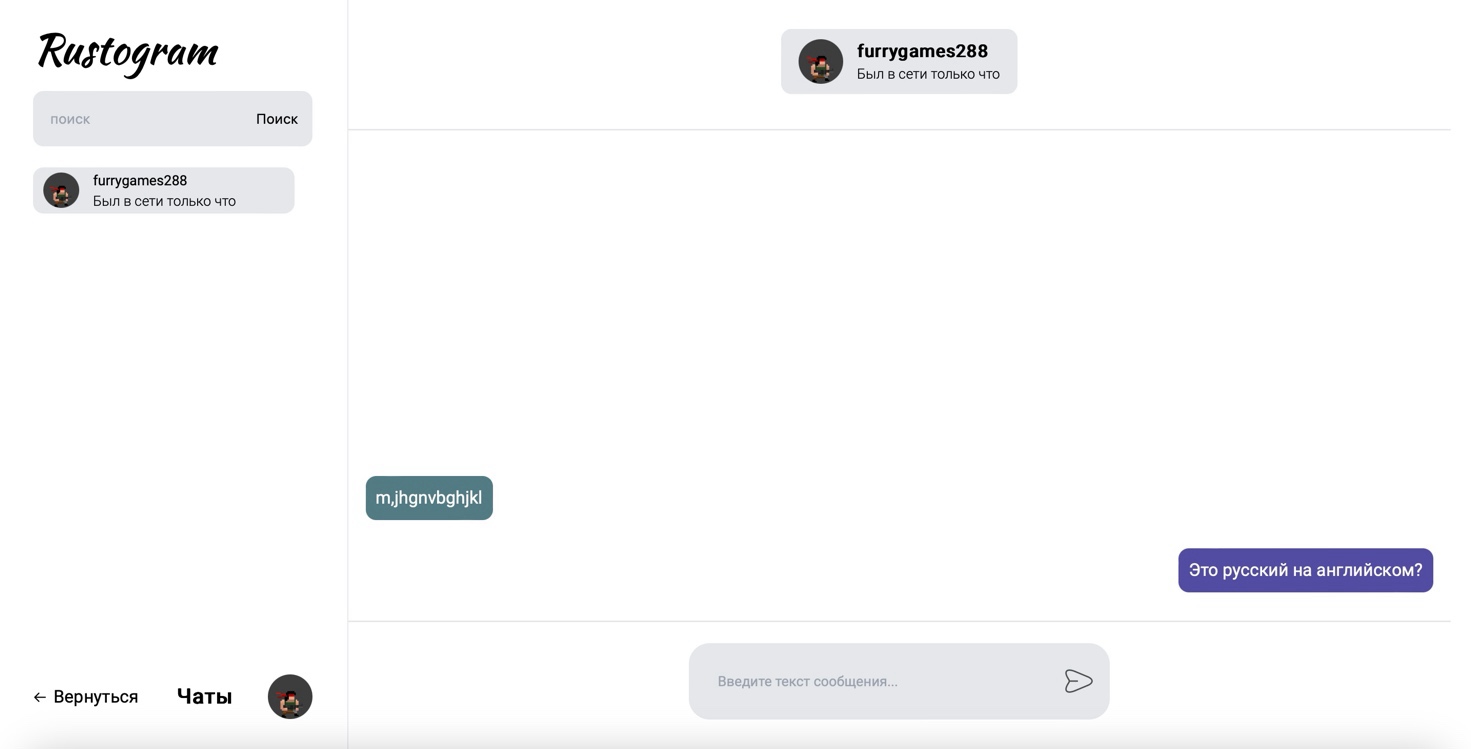
Б.6 – Страница коротких видео в приложении



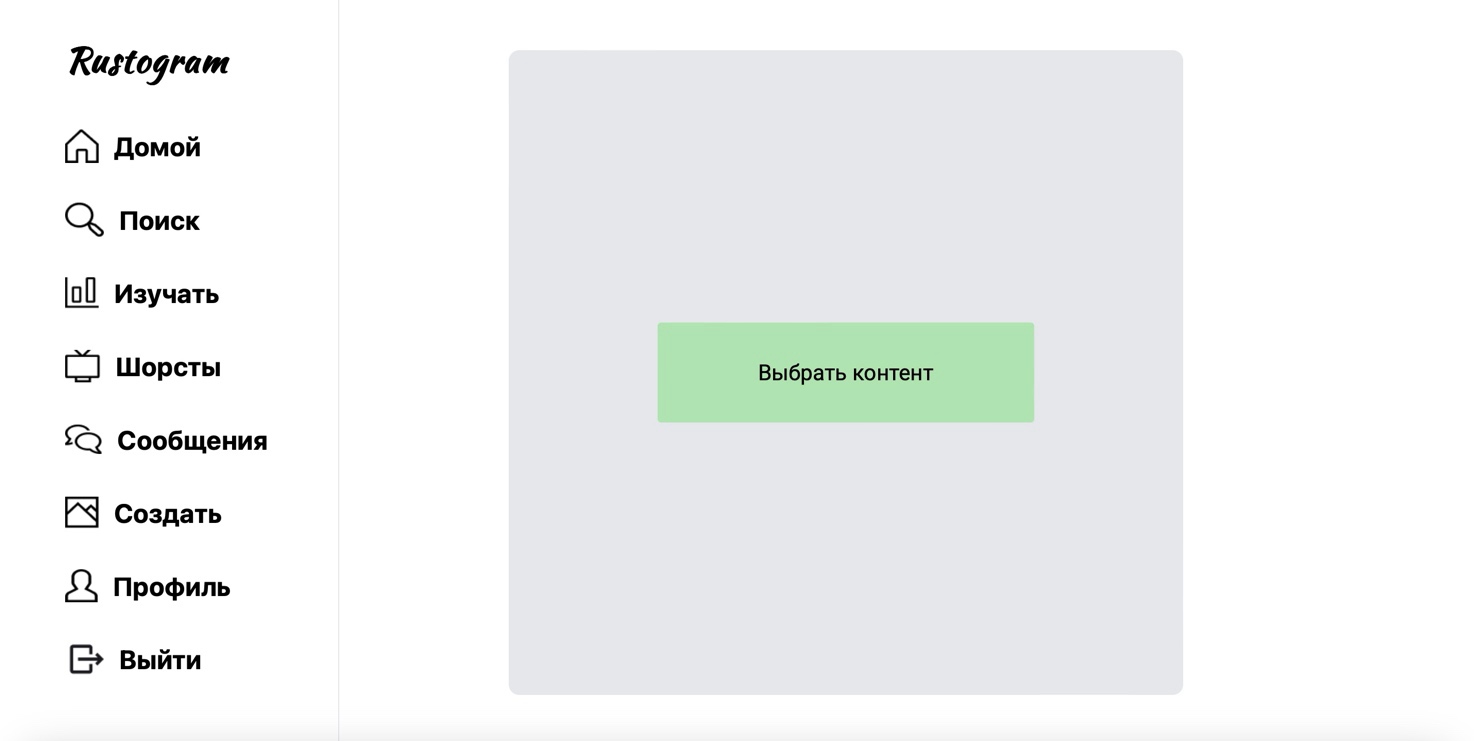
Б.7 – Страница изучать в приложении



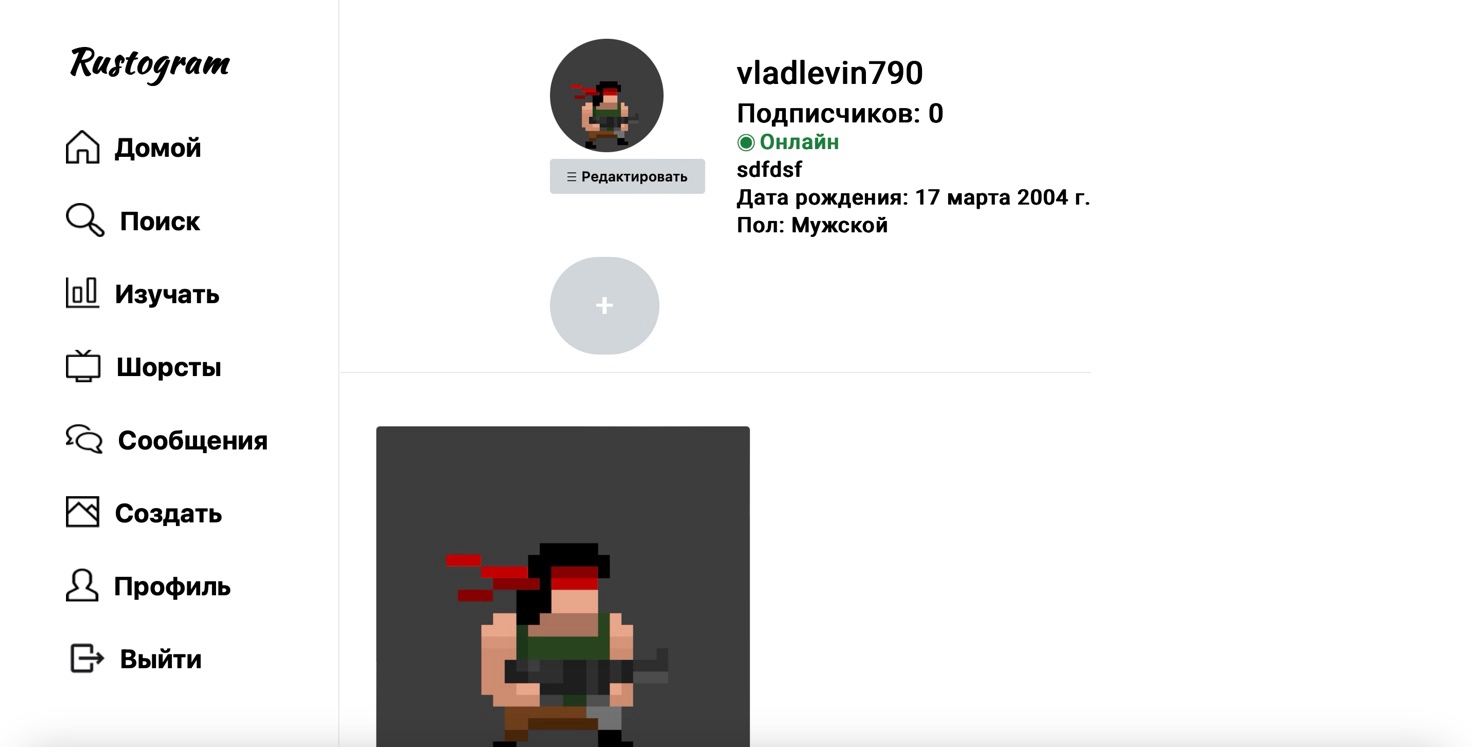
Б.8 – Страница создания коротких видео в приложении



Б.8 – Страница чатов в приложении



Б.9 – Страница создания постов в приложении



Б.10 – страница профиля в приложении

# Приложение В

public function createChat($request, $userId)  
{  
 try {  
 $user = $request->user();  
 $secondUser = User::*FindOrFail*($userId);  
  
 if (!$secondUser) {  
 return response()->json([‘error’ => ‘User not found’], 404);  
 }  
 if (!$user) {  
 return response()->json([‘error’ => ‘User not found’], 404);  
 }  
  
 $existingChat = Users\_chats::*where*(function ($query) use ($user, $secondUser) {  
 $query->where(‘owner\_id’, $user->id)  
 ->where(‘user\_id’, $secondUser->id);  
 })->orWhere(function ($query) use ($user, $secondUser) {  
 $query->where(‘owner\_id’, $secondUser->id)  
 ->where(‘user\_id’, $user->id);  
 })->first();  
  
 if ($existingChat) {  
 return response()->json([‘error’ => ‘Chat already exists’], 400);  
 }  
  
 $chat = new Chat();  
 $chat->name = ‘Chat between ‘ . $user->name . ‘ and ‘ . $secondUser->name;  
 $chat->save();  
  
 $userChat = new Users\_chats();  
 $userChat->chat\_id = $chat->id;  
 $userChat->owner\_id = $user->id;  
 $userChat->user\_id = $secondUser->id;  
 $userChat->save();  
  
 return $chat;  
 } catch (\Exception $e) {  
 return [‘Success’=>false,’Message’=>’Error with creating chat’];  
 }  
}

Листинг В.1 – Метод createChat

public function getAllMessagesFromChat($request, $chatId, $userId)  
{  
 try {  
 $user = $request->user();  
 $secondUser = User::*findOrFail*($userId);  
 $userChat = Users\_chats::*where*(function ($query) use ($user, $secondUser) {  
 $query->where('owner\_id', $user->id)  
 ->where('user\_id', $secondUser->id);  
 })->orWhere(function ($query) use ($user, $secondUser) {  
 $query->where('owner\_id', $secondUser->id)  
 ->where('user\_id', $user->id);  
 })->where('chat\_id', $chatId)  
 ->first();  
 if (!$userChat) {  
 return response()->json(['error' => 'Not Found'], 404);  
 }  
 if(!$user) {  
 return response()->json(['error' => 'Unauthorized'],401);  
 }  
 if(!$secondUser) {  
 return response()->json(['error' => 'Unauthorized'],401);  
 }  
 $messages = Messages::*where*('chat\_id', $chatId)->get();  
 return $messages;  
 } catch (\Exception $e) {  
 return ['Success'=>false, 'Message'=>'Error with gitting all messages from this chat'];  
 }  
}

Листинг В.2 – Метод getAllMessagesFromChat

# Приложение Г



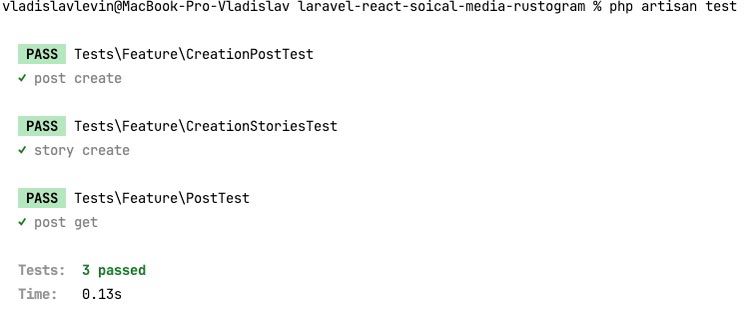
Г.1 – Тест получения поста



Г.2 – Тест создания поста



Г.3 – Тест создания истории



Г.4 – Результаты тестов

|  |  |
| --- | --- |
| Поле | Описание |
| 1 | 2 |
| Название проекта | Rustogram |
| Рабочая версия | 1.0 |
| Имя тестирующего | Levin Vladislav |
| Дата(ы) теста | 26.04.2024 |
| Тестовый пример # | TC\_UI\_1 |
| Приоритет тестирования (Низкий/Средний/Высокий) | Высокий |
| Заголовок/название теста | Регистрация пользователя |
| Краткое изложение теста | После того, как пользователь зарегистрировался, в базе данных появляется поле с данными пользователя |
| Этапы теста | 1. Перейти на страницу регистрации 2. Заполнить поле «Имя» 3. Заполнить поле email 4. Заполнить поле «Пароль» 5. Заполнить поле «Подтверждение пароля» 6. Нажать на кнопку «Регистрация» 7. Вывести данные из таблицы users |
| Тестовые данные | Email пользователя: [vladlevin790@gmail.com](mailto:vladlevin790@gmail.com)  Пароль пользователя: 1234567+f |
| Ожидаемый результат | В таблице users появилось поле с новым пользователем с почтой [vladlevin790@gmail.com](mailto:vladlevin790@gmail.com) |
| Фактический результат | В таблице users появилось поле с новым пользователем с почтой [vladlevin790@gmail.com](mailto:vladlevin790@gmail.com) |
| Предварительное условие | Пользователь должен иметь стабильное интернет-соединение |
| Статус (Зачет/Незачет) | Зачёт |

Г.5 – Тест-кейс №1. Тестирования регистрации пользователя

|  |  |
| --- | --- |
| Поле | Описание |
| Название проекта | Rustogram |
| Рабочая версия | 1.0 |
| Имя тестирующего | Levin Vladislav |
| Дата(ы) теста | 26.04.2024 |
| Тестовый пример # | TC\_UI\_2 |
| Приоритет тестирования (Низкий/Средний/Высокий) | Средний |
| Заголовок/название теста | Изменение пола пользователя в профиле |
| Краткое изложение теста | Когда пользователь в своём профиле меняет поле «Пол» в модальном окне, данные в его профиле изменяются |
| Этапы теста | 1. Авторизоваться в системе под своим логином и паролем 2. Нажать на кнопку «Изменить» 3. В выпадающем списке в поле «Пол» выбрать опцию «мужской» 4. Нажать кнопку «Подтвердить» |
| Тестовые данные | Email пользователя: [vladlevin790@gmail.com](mailto:vladlevin790@gmail.com)  Пароль пользователя: 1234567+f |
| Ожидаемый результат | В профиле в поле «Пол» указан «мужской» |
| Фактический результат | В профиле в поле «Пол» указан «мужской» |
| Предварительное условие | Пользователь должен иметь стабильное интернет-соединение и его данные есть в базе данных |
| Постусловие |  |
| Статус (Зачет/Незачет) | Зачёт |
| Примечания/комментарии |  |

Г.6 – Тест-кейс №2. Тестирование изменения пола пользователя в профиле

|  |  |
| --- | --- |
| Поле | Описание |
| Название проекта | Rustogram |
| Рабочая версия | 1.0 |
| Имя тестирующего | Levin Vladislav |
| Дата(ы) теста | 26.04.2024 |
| Тестовый пример # | TC\_UI\_3 |
| Приоритет тестирования (Низкий/Средний/Высокий) | Средний |
| Заголовок/название теста | Изменение дня рождения пользователя в профиле |
| Краткое изложение теста | Когда пользователь в своём профиле меняет поле «Дата рождения» в модальном окне, данные в его профиле изменяются |
| Этапы теста | 1. Авторизоваться в системе под своим логином и паролем 2. Нажать на кнопку «Изменить» 3. Заполнить поле «Дата рождения» на «17.03.2004» 4. Нажать кнопку «Подтвердить» |
| Тестовые данные | Email пользователя: [vladlevin790@gmail.com](mailto:vladlevin790@gmail.com)  Пароль пользователя: 1234567+f |
| Ожидаемый результат | Поле «Дата рождения» в профиле заполнена данными «17.03.2004» |
| Фактический результат | Поле «Дата рождения» в профиле заполнена данными «17.03.2004» |
| Предварительное условие | Пользователь должен иметь стабильное интернет-соединение и его данные есть в базе данных |
| Постусловие |  |
| Статус (Зачет/Незачет) | Зачёт |
| Примечания/комментарии |  |

Г.7 – Тест-кейс №3. Тестирование изменения даты рождения в профиле пользователя

|  |  |
| --- | --- |
| Поле | Описание |
| Название проекта | Rustogram |
| Рабочая версия | 1.0 |
| Имя тестирующего | Levin Vladislav |
| Дата(ы) теста | 26.04.2024 |
| Тестовый пример # | TC\_UI\_4 |
| Приоритет тестирования (Низкий/Средний/Высокий) | Высокий |
| Заголовок/название теста | Создание поста |
| Краткое изложение теста | Когда пользователь заходит на страницу создание поста и создаёт там новый пост, при переходе на профиль пользователя, появляется созданный пост |
| Этапы теста | 1. Авторизоваться в системе под своим логином и паролем 2. Перейти на страницу создания поста, нажав на кнопку «Создать» на левом сайдбаре 3. Нажать на кнопку «Создать пост» 4. Выбрать изображение 5. В поле описание ввести «Мой новый пост» 6. Нажать кнопку «Создать» 7. Перейти в профиль пользователя |
| Тестовые данные | Email пользователя: [vladlevin790@gmail.com](mailto:vladlevin790@gmail.com)  Пароль пользователя: 1234567+f |
| Ожидаемый результат | В профиле пользователя появляется новый пост с описанием «Мой новый пост» |
| Фактический результат | В профиле пользователя появляется новый пост с описанием «Мой новый пост» |
| Предварительное условие | Пользователь должен иметь стабильное интернет-соединение и его данные есть в базе данных |
| Постусловие |  |
| Статус (Зачет/Незачет) | Зачёт |

Г.8 – Тест-кейс №4. Тестирование создания поста

|  |  |
| --- | --- |
| Поле | Описание |
| Название проекта | Rustogram |
| Рабочая версия | 1.0 |
| Имя тестирующего | Levin Vladislav |
| Дата(ы) теста | 26.04.2024 |
| Тестовый пример # | TC\_UI\_5 |
| Приоритет тестирования (Низкий/Средний/Высокий) | Высокий |
| Заголовок/название теста | Создание истории |
| Краткое изложение теста | При создании истории пользователем, его история появляется в его профиле |
| Этапы теста | 1. Авторизоваться в системе под своим логином и паролем 2. Нажать на кнопку «Профиль» в левом сайдбаре 3. Нажать на кнопку «+» 4. Выбрать изображение 5. В поле описание ввести «Моя новая история» 6. Нажать кнопку «Создать» |
| Тестовые данные | Email пользователя: [vladlevin790@gmail.com](mailto:vladlevin790@gmail.com)  Пароль пользователя: 1234567+f |
| Ожидаемый результат | В профиле пользователя появляется новая история с описанием «Моя новая история» |
| Фактический результат | В профиле пользователя появляется новая история с описанием «Моя новая история» |
| Предварительное условие | Пользователь должен иметь стабильное интернет-соединение и его данные есть в базе данных |
| Постусловие |  |
| Статус (Зачет/Незачет) | Зачёт |
| Примечания/комментарии |  |

Г.9 – Тест-кейс №5. Тестирование создания истории

|  |  |
| --- | --- |
| Поле | Описание |
| Название проекта | Rustogram |
| Рабочая версия | 1.0 |
| Имя тестирующего | Levin Vladislav |
| Дата(ы) теста | 26.04.2024 |
| Тестовый пример # | TC\_UI\_5 |
| Приоритет тестирования (Низкий/Средний/Высокий) | Высокий |
| Заголовок/название теста | Создание чата |
| Краткое изложение теста | Пользователь переходит на страницу с чатами, в строке поиск вводит nickname пользователя, с которым хочет начать чат, после создания чата он появляется на странице чата у пользователя и у его собеседника |
| Этапы теста | 1. Авторизоваться в системе под своим логином и паролем 2. Нажать на кнопку «Чаты» в левом сайдбаре 3. В поле поиск ввести «paulos9932» 4. Выбрать пользователя с nickname «paulos9932» |
| Тестовые данные | Email пользователя: [vladlevin790@gmail.com](mailto:vladlevin790@gmail.com)  Пароль пользователя: 1234567+f  Nickname собеседника: paulos9932 |
| Ожидаемый результат | На странице с чатами появляется пустая переписка с пользователем paulos9932 |
| Фактический результат | На странице с чатами появляется пустая переписка с пользователем paulos9932 |
| Предварительное условие | Пользователь должен иметь стабильное интернет-соединение, данные пользователя и данные его собеседника есть в базе данных |
| Постусловие |  |
| Статус (Зачет/Незачет) | Зачёт |
| Примечания/комментарии |  |

Г.10 – Тест-кейс №6. Тестирование создания чата