# ВВЕДЕНИЕ

Данная курсовая работа посвящена разработке инновационного веб-приложения для обмена сообщениями под названием «TwentyThree». Отличительной особенностью проекта является концепция временных чатов по интересам, которые автоматически создаются для активных пользователей ежедневно и существуют ровно 23 часа, после чего безвозвратно удаляются.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Обзор существующих конкурирующих продуктов по теме работы

**Telegram** – популярный кроссплатформенный мессенджер с акцентом на скорость и безопасность. Он поддерживает текстовые и мультимедийные сообщения, групповые чаты до 200 000 участников и каналы «только для чтения»[en.wikipedia.org](https://en.wikipedia.org/wiki/Telegram_(software)#:~:text=match%20at%20L526%20Telegram%20users,They%20support%20flexible%20admin). В 2022 году в Telegram введена функция топиков (тем) – создание подгрупп внутри больших чатов для обсуждения отдельных тем[habr.com](https://habr.com/ru/news/697810/#:~:text=%D0%9D%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B7%D0%B2%D0%BE%D0%BB%D1%8F%D0%B5%D1%82%20%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%B2%D0%B0%D1%82%D1%8C%20%D0%BE%D1%82%D0%B4%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5,%D0%B0%D1%80%D1%81%D0%B5%D0%BD%D0%B0%D0%BB%20%D0%BF%D1%80%D0%B8%D0%B2%D1%8B%D1%87%D0%BD%D1%8B%D1%85%20%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%2C%20%D0%B2%20%D1%82%D0%BE%D0%BC). Это помогает структурировать общение в крупных сообществах, разделяя его по интересам. В Telegram также доступны боты, опросы, платформа каналов и другие инструменты вовлечения. Что касается эфемерности, Telegram предлагает секретные чаты с автоудаляющимися сообщениями и общий таймер самоуничтожения для чатов. В целом Telegram не ориентирован на случайные знакомства, а скорее на общение с уже добавленными в друзья пользователями.

**Discord** – платформа сообществ, изначально предназначенная для геймеров, превратившаяся в универсальный мессенджер. Общение организуется через серверы (удалённые чаты с приглашениями) и каналы по темам: текстовые, голосовые и видеочаты. Каналы структурированы по категориям и темам, что позволяет участникам обсуждать различные аспекты одного сообщества, не «засоряя» основной чат[discord.com](https://discord.com/servers#:~:text=Discord%20servers%20are%20organized%20into,clogging%20up%20a%20group%20chat). В 2021 году для повышения вовлечённости Discord внедрил треды – временные текстовые каналы, которые могут автоматически удаляться через указанное время[en.wikipedia.org](https://en.wikipedia.org/wiki/Discord#:~:text=In%20August%202021%2C%20Discord%20launched,64). Это позволяет создавать кратковременные беседы в пределах сервера. Discord также предлагает **Stage Channels** (аудио-конференции) и стримы. Однако в Discord нет механизма спонтанного соединения незнакомцев: пользователи сами присоединяются к серверам по интересам через поиск или приглашения. Большая часть содержимого – переписки и записи голосовых комнат – сохраняется, что отличает его от полностью эфемерных сервисов.

**Clubhouse** – это социальная аудиоплатформа, в которой пользователи устраивают живые голосовые комнаты по темам. Любой пользователь может мгновенно создать незапланированную комнату или запланировать её заранее[builtin.com](https://builtin.com/articles/clubhouse-business#:~:text=Clubhouse%2C%20an%20invite,the%20conversation%20matches%20their%20expectations). Разговоры протекают в реальном времени, а аудио контент **не сохраняется** и «исчезает» после окончания комнаты[builtin.com](https://builtin.com/articles/clubhouse-business#:~:text=attend%20conference%20calls%20based%20on,any%20kind%20of%20written%20message)[npr.org](https://www.npr.org/2021/06/11/1005304644/social-audio-began-as-a-pandemic-fad-tech-companies-see-it-as-the-future#:~:text=Then%20Twitter%20invited%20her%20to,Audie%20Cornish%20and%20Surgeon%20General). Такой формат создает эффект дефицита: если пользователь не присутствует в комнате в момент беседы, он упускает разговор[builtin.com](https://builtin.com/articles/clubhouse-business#:~:text=At%20any%20given%20moment%2C%20a,a%20lot%20of%20time%20on). Clubhouse стал популярен благодаря экспоненциальному росту вовлеченности за счёт прямого и открытого общения с знаменитостями. Среди особенностей – фокус на контенте, тематические клубы и система приглашений. К недостаткам относятся отсутствие текстового чата, а также ограниченность платформы чисто аудио-форматом.

## 1.2 Сравнительный анализ

**Telegram:** плюс – крупные тематические сообщества и каналы (до 200 000 участников)[en.wikipedia.org](https://en.wikipedia.org/wiki/Telegram_(software)" \l ":~:text=match%20at%20L526%20Telegram%20users,They%20support%20flexible%20admin" \t "_blank), удобные инструменты (боты, опросы, каналы новостей), новая функция *топиков* для субчатов по узким темам[habr.com](https://habr.com/ru/news/697810/" \l ":~:text=%D0%9D%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B7%D0%B2%D0%BE%D0%BB%D1%8F%D0%B5%D1%82%20%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%B2%D0%B0%D1%82%D1%8C%20%D0%BE%D1%82%D0%B4%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5,%D0%B0%D1%80%D1%81%D0%B5%D0%BD%D0%B0%D0%BB%20%D0%BF%D1%80%D0%B8%D0%B2%D1%8B%D1%87%D0%BD%D1%8B%D1%85%20%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%2C%20%D0%B2%20%D1%82%D0%BE%D0%BC" \t "_blank). Минус – отсутствие механизма случайного объединения незнакомцев в группы: пользователь сам выбирает, к каким чатам вступать.

**Discord:** плюс – сообщества по интересам с иерархией каналов (текстовых и голосовых)[discord.com](https://discord.com/servers#:~:text=Discord%20servers%20are%20organized%20into,clogging%20up%20a%20group%20chat); поддержка временных тредов (temporary threads), исчезающих через время[en.wikipedia.org](https://en.wikipedia.org/wiki/Discord#:~:text=In%20August%202021%2C%20Discord%20launched,64); интерактивные функции (голосовые комнаты, стриминг, реакции, боты, роли). Эффективен для постоянных сообществ, легок поиск по темам. Минус – случайных встреч сервис не предлагает: нужно самостоятельно присоединяться к сообществам. Платформа может быть перегружена для новичков, требует модерирования больших серверов и не стимулирует частое возвращение ради новых групп.

**Clubhouse:** плюс – живые тематические аудио чаты: комнаты создаются спонтанно, темы задают участники, а содержание разговора исчезает сразу после окончания. Это обеспечивает высокий уровень вовлеченности: пользователи не хотят пропускать незапланированные дискуссии. Минус – только голосовой формат (не подходит тем, кто предпочитает текст); сложность поиска новых контактов без истории; отсутствие сохранения чатов делает контент недоступным после завершения; требуется активный модератор и зачастую приглашения. Clubhouse хорош для нетворкинга и фриланс-мероприятий, но не для постоянного общения внутри статичных тематических групп.

## 1.3 Вывод по итогам анализа существующих конкурентов

Анализ показывает, что на современном рынке отсутствует мессенджер с механикой ежедневной автоматической смены групповых чатов по интересам и строго ограниченным временем существования каждой комнаты. Решения типа Telegram и Discord обеспечивают тематические сообщества, но участники остаются в них по своему выбору, а чаты сохраняются бессрочно.

# 2 АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ «TwentyThree»

# Для обеспечения стабильной работы, высокой производительности и удобства разработки, мы выбрали следующий стек технологий: Java с использованием Spring Framework для бэкенда, React для фронтенда и PostgreSQL в качестве системы управления базами данных.

## 2.1. Бэкенд: Java с использованием Spring Framework

Java в сочетании с Spring Framework (в частности, Spring Boot) является мощным и проверенным решением для создания корпоративных и высоконагруженных веб-приложений.

Обоснование выбора:

* Надежность и масштабируемость Java: Java — это зрелый, стабильный и высокопроизводительный язык программирования, идеально подходящий для создания масштабируемых серверных приложений. Его виртуальная машина (JVM) обеспечивает кроссплатформенность и эффективное управление ресурсами, что критично для приложения, обрабатывающего большое количество одновременных подключений и данных.
* Экосистема Spring Framework: Spring Framework предоставляет комплексную и модульную платформу для разработки Java-приложений. Он упрощает разработку, предлагая решения для:
  + Внедрения зависимостей (Dependency Injection): Упрощает тестирование и повышает модульность кода.
  + Аспектно-ориентированного программирования (AOP): Позволяет централизованно управлять сквозными функциями, такими как логирование и безопасность.
  + Spring Boot: Значительно ускоряет процесс разработки и развертывания, минимизируя необходимость в ручной конфигурации. Это позволяет быстро создавать готовые к производству микросервисы, что идеально подходит для итеративной разработки.
  + Обработки данных: Spring Data упрощает взаимодействие с базами данных, а Spring WebFlux (для реактивного программирования) может быть использован для эффективной обработки большого количества параллельных запросов, что важно для чат-приложения.
  + Безопасности: Spring Security предоставляет мощные и гибкие механизмы аутентификации и авторизации, что является приоритетом для любого мессенджера.
* Поддержка микросервисов: Spring Boot отлично подходит для архитектуры микросервисов, что дает гибкость в масштабировании отдельных компонентов приложения по мере необходимости. Это позволяет эффективно управлять такими функциями, как создание чатов, управление пользователями и обработка сообщений.
* Большое сообщество и обширная документация: Java и Spring имеют одно из самых больших и активных сообществ разработчиков, а также обширную документацию и множество ресурсов, что обеспечивает легкость поиска решений и поддержки.

## 2.2 Фронтенд: React

Выбор: React — это популярная JavaScript-библиотека для создания пользовательских интерфейсов, разработанная Facebook.

Обоснование выбора:

* Компонентный подход: React основан на компонентном подходе, что позволяет создавать сложные пользовательские интерфейсы из небольших, переиспользуемых и независимых компонентов. Это значительно упрощает разработку, тестирование и поддержку кода, особенно для динамичного интерфейса чата.
* Виртуальный DOM: React использует виртуальный DOM, который оптимизирует процесс обновления пользовательского интерфейса. Вместо прямого манипулирования реальным DOM, React сначала обновляет виртуальное представление, а затем эффективно синхронизирует изменения с реальным DOM. Это обеспечивает высокую производительность и плавность работы интерфейса, что критично для интерактивного чат-приложения.
* Декларативный подход: Декларативный стиль React делает код более предсказуемым и легким для понимания. Разработчик описывает, как должен выглядеть интерфейс в различных состояниях, а React берет на себя заботу о его обновлении.
* Большое и активное сообщество: React имеет огромное сообщество разработчиков, что означает доступность большого количества библиотек, инструментов, учебных материалов и готовых решений. Это ускоряет разработку и обеспечивает поддержку в случае возникновения проблем.
* Однонаправленный поток данных: Упрощает отладку и управление состоянием приложения, что важно для поддержания консистентности данных в чате.
* Поддержка мобильной разработки (React Native): Хотя для данного проекта не указана мобильная версия, знание React открывает путь к кроссплатформенной мобильной разработке с помощью React Native в будущем, что является дополнительным преимуществом.

## 2.3 База данных: PostgreSQL

PostgreSQL — это мощная, открытая, объектно-реляционная система управления базами данных.

Обоснование выбора:

* Надежность и целостность данных (ACID): PostgreSQL полностью соответствует принципам ACID (Atomicity, Consistency, Isolation, Durability), что гарантирует надежность и целостность данных. Это крайне важно для мессенджера, где потеря или искажение сообщений недопустимы.
* Масштабируемость: PostgreSQL способен эффективно обрабатывать как небольшие, так и очень большие объемы данных и высокие нагрузки. Он поддерживает различные методы масштабирования, включая репликацию и шардинг, что позволяет приложению расти вместе с количеством пользователей.
* Расширяемость: PostgreSQL известен своей расширяемостью. Он позволяет добавлять новые типы данных, функции, операторы, агрегаты, индексные методы и языки программирования. Это дает гибкость для реализации специфических требований проекта.
* Поддержка JSONB: Встроенная поддержка типа данных JSONB позволяет хранить и эффективно запрашивать полуструктурированные данные (например, метаданные чатов или пользовательские настройки), сочетая преимущества реляционной и NoSQL моделей.
* Открытый исходный код и активное сообщество: Будучи проектом с открытым исходным кодом, PostgreSQL имеет большое и преданное сообщество, которое постоянно улучшает его, предоставляет поддержку и разрабатывает новые функции. Это снижает затраты на лицензирование и обеспечивает долгосрочную жизнеспособность.
* Гибкость схем: Хотя PostgreSQL является реляционной базой данных, его возможности по работе с JSONB позволяют хранить более гибкие структуры данных, что может быть полезно для хранения информации о чатах и их интересах.

# 3 ЦЕЛЬ И ЗАДАЧИ

Основной целью данной курсовой работы является проектирование и разработка прототипа веб-приложения «TwentyThree», демонстрирующего ключевой функционал создания и управления временными чатами, а также обеспечивающего стабильное взаимодействие между пользователями. При этом будет использован современный стек технологий, включающий Java с Spring Framework для бэкенда, React для фронтенда и PostgreSQL в качестве системы управления базами данных, с акцентом на обоснование выбора и эффективное применение данных инструментов.

# Задачи:

# Провести анализ существующих решений.

# Определить целевую аудиторию

# Анализ требований и проектирование архитектуры.

# Проектирование и реализация базы данных. Разработать схему базы данных PostgreSQL для эффективного хранения информации о пользователях, чатах, сообщениях и их интересах. Реализовать механизмы взаимодействия бэкенда с базой данных, используя Spring Data JPA или другие подходящие инструменты.

# Разработка серверной части (бэкенд). Реализовать RESTful API с использованием Java и Spring Boot для управления пользователями, чатами и сообщениями. Разработать логику автоматического создания временных чатов на основе интересов пользователей и их удаления по истечении 23 часов. Интегрировать механизмы аутентификации и авторизации с использованием Spring Security для обеспечения безопасности данных.

# Разработка клиентской части (фронтенд). Создать пользовательский интерфейс веб-приложения с использованием React, обеспечивающий интуитивно понятное взаимодействие. Реализовать функционал отображения активных чатов, присоединения к ним, отправки и получения сообщений в реальном времени. Обеспечить адаптивный дизайн для корректного отображения приложения на различных устройствах.

# Тестирование и отладка. Провести модульное, интеграционное и системное тестирование разработанных компонентов для выявления и устранения ошибок.