**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Прикладной информатики**

Направление подготовки **45.03.04 Интеллектуальные системы в гуманитарной сфере**

Образовательная программа[**Языковые модели и искусственный интеллект**](https://isu.ifmo.ru/pls/apex/f?p=2143:13:116963507483892::NO::EP_UCHEB_YEAR,EP_MEGAFACULTY,EP_FACULTY,EP_CATHEDRA,EP_GROUP,EP_FO,EP_KVAL,EP_UCHEB_PLAN:2024/2025,-1,725,-1,K3160,-1,-1,117277)

**К У Р С О В О Й   П Р О Е К Т**

Тема: «Приложение для работы с экспортом из Telegram»

Обучающийся: Белов Александр Павлович К3160

Санкт-Петербург 2024

**СОДЕРЖАНИЕ**

1 ВВЕДЕНИЕ………………………………………………..3

1.1 Актуальность рассматриваемой темы…………………3

1.2 Цель проекта…………………………………………….3

1.3 Задачи проекта…………………………………………..3

2 РАБОТА НАД ПРОЕКТОМ……………………………...4

2.1 Суть проекта……………………………………………..4

2.2 Процессы работы над всем проектом………………….4

2.2.1 Исследование целевой аудитории……………………5

2.2.2 Определение функционала…………….……………...5

2.3 Мои задачи……………..………………………………..5

2.3.1 Создание моделей данных……………………………6

2.3.2 Реализация отображения динамических данных……8

2.4 Анализ моей работы……………………..……………...11

2.5 Взаимодействие с командой……………………..……..13

2.6 Взаимодействие с руководителем…..………………….13

3 ЗАКЛЮЧЕНИЕ………..…………………………………..13

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ………..15

1. **ВВЕДЕНИЕ**

**1.1 Актуальность рассматриваемой темы**

Telegram — один из самых популярных мессенджеров в мире, используемый миллионами людей для общения, обмена файлами и хранения информации. Однако, несмотря на возможности Telegram для экспорта чатов, данный процесс может быть неудобным и требовать дополнительных инструментов для удобной работы с большими объемами данных.

Мобильное приложение для работы с экспортированными данными из Telegram существенно упрощает процессы архивирования, анализа и миграции данных для дальнейшего пользования с ними.

**1.2 Цель проекта**

Цель проекта - создать прототип мобильного приложения для работы с экспортом из Telegram, которое позволит пользователям эффективно организовывать, анализировать и взаимодействовать с экспортированными данными, такими как чаты, сообщения, медиафайлы, и ссылки.

**1.3 Задачи проекта**

Для достижения поставленной цели проект был разделен на несколько ключевых этапов и задач:

Создать макет экрана загрузки данных.

Разработать модели данных для хранения информации.

Сверстать экран для загрузки данных.

Реализовать механизм парсинга данных формата HTML.

Сверстать экран для отображения загруженных данных.

Реализовать функционал загрузки данных в приложение.

Разработать механизм отображения динамических данных в реальном времени.

Провести тестирование и отладку приложения.

1. **РАБОТА НАД ПРОЕКТОМ**
   1. **Суть проекта**

Проект заключался в разработке мобильного приложения, предназначенного для обработки экспортированных данных из Telegram. Telegram предоставляет возможность выгружать чаты и переписки в формате HTML, что удобно для сохранения важной информации. Однако не даёт полной возможности для её фильтрации, анализа и полноценного функционала для взаимодействия с ней. Разработанное приложение позволяет экспортировать данные в форматах JSON, CSV, PDF, обеспечивая удобный доступ и возможность дальнейшего использования данных для анализа, архивирования или обмена. Упрощённый интерфейс способствует быстрой адаптации к приложению, а также доступность для пользователей без технических знаний.

* 1. **Процессы работы над всем проектом**

Процесс каждого этапа на протяжении всей работы над проектом курировался его руководителем: Черным Яном Дмитриевичем. Каждую неделю, на общем собрании Ян назначал участникам задания, которые необходимо было выполнить до следующей встречи. На этих встречах обсуждался общий прогресс всей работы, и формулировался график последующих задач для того, чтобы успеть в установленные временные рамки. Такой подход способствовал эффективному взаимодействию внутри команды, позволяя каждому быть вовлеченным в общий процесс и своевременно корректировать свою работу в зависимости от обсужденных результатов. Работа выполнялась на языке программирования Flatter, в связи с прежним опытом работы руководителя проекта и общим отсутствием опыта работы с каким-либо языком программирование прежде у нескольких человек в команде. В случае возникновения трудностей или необходимости дополнительных пояснений, Ян всегда был готов предоставить помощь и направить нас в нужное русло.

**2.2.1 Исследование целевой аудитории**

Перед началом разработки была проведена работа по изучению целевой аудитории. На основе этого исследования был определен основной круг пользователей:

1. **Обычные пользователи** — люди, заинтересованные в сохранении важной переписки и медиафайлов из Telegram. Также нуждающиеся в простом и адаптивном интерфейсе за неимением технических знаний.
2. **Исследователи и аналитики данных —** специалисты, которым требуется проводить анализ данных, полученных из чатов, например, для обучения моделей машинного обученияили создания отчётов.
3. **Юридические лица, предприниматели —** люди, для которых требуется приложение с архивацией деловых переписок, а также для подведения отчётности.
4. **IT-специалисты —** для эффективной интеграции чатов из телеграмма в другие системы, работы с большими объёмами данных.

**2.2.2 Определение функционала**

Функциональные требования включают в себя**:**

* Загрузку чатов из экспортированных файлов (HTML, JSON, CSV, PDF).
* Вывод списка чатов с возможностью фильтрации, по ключевым словам, дате и типу файлов.
* Интерактивный интерфейс для просмотра сообщений с медиафайлами и ссылками.
* Механизм поиска по сообщениям и экспортированным данным.
* Сохранение данных в различных форматах (JSON, PDF, CSV).
* Настройка структуры экспорта (включение/исключение авторов, дат, типов контента).
* Логирование экспортированных данных в журнал операций

**2.3 Мои задачи**

В рамках проекта мне были назначены следующие задачи:

1. **Создание моделей данных** — создание классов для хранения данных из экспортированных чатов.
2. **Реализация отображения динамических данных** — разработка механизма отображения сообщений в чате.

**2.3.1 Создание моделей данных**

Для реализации этой задачи Ян дал мне задачу по изучению основных принципов объектно-ориентированного программирования (ООП), включая создание классов и абстрактных классов. Каждая модель данных должна была содержать методы fromJson и toJson, чтобы обеспечивать возможность сериализации и десериализации данных.

Для этого я:

* Изучил структуру экспортированных файлов (JSON).
* Выделил сущности данных (сообщения, медиафайлы, чаты).
* Создал соответствующие классы в приложении для обработки данных.

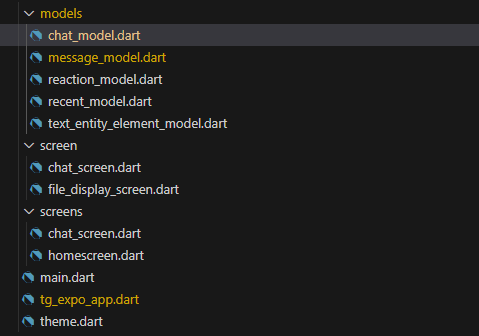


Рисунок 1. Обобщенный вид всех моделей данных

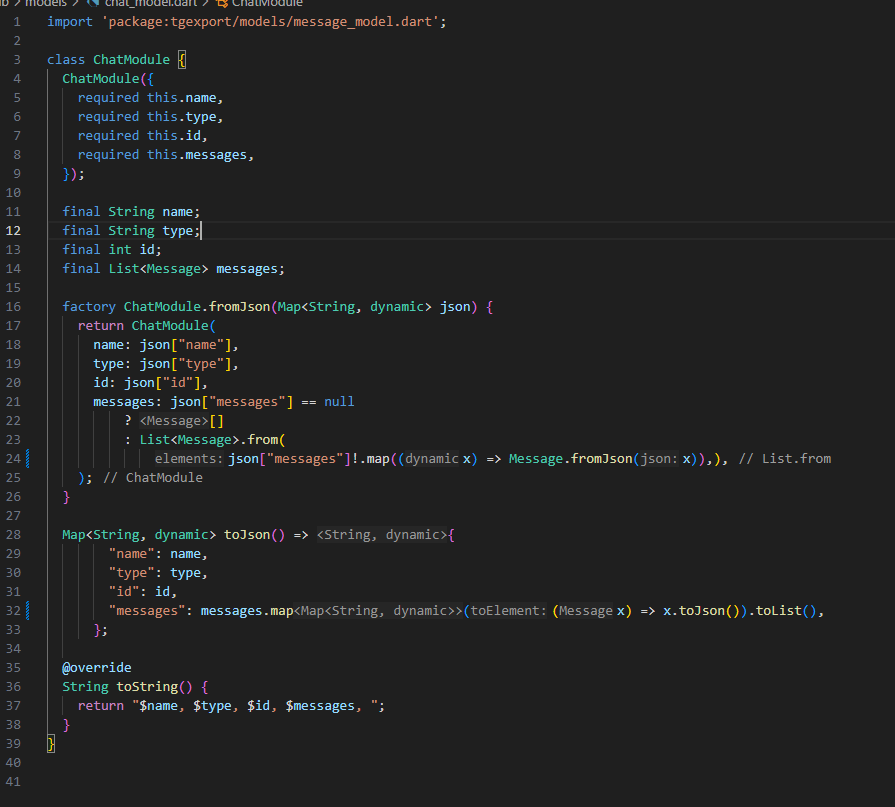


Рисунок 2. Модель класса ChatModule

Каждая модель данных в проекте включает определённый тип информации и выполняет конкретную роль в обработке данных, полученных из экспортированных чатов.

Так, например, **ChatModel** отвечает за хранение и обработку информации, связанной с чатами. Эта модель включает в себя все необходимые атрибуты, такие как идентификаторы чатов, их название, описание и список участников.

**ReactionModel** отвечает за хранение реакций пользователей на сообщения, например, лайков, смайликов или других типов взаимодействий, предоставляемых Telegram.

Важной деталью было интегрировать методы **fromJson** и **toJson** для каждой модели данных, что обеспечило удобство и гибкость при работе с данными, экспортированными из Telegram. Метод **fromJson** позволяет преобразовывать данные, полученные в формате JSON, в объекты, которые можно использовать в приложении. Это особенно важно для парсинга экспортированных файлов Telegram, которые обычно имеют структуру JSON. С помощью этого метода можно автоматически и корректно извлекать данные из файла, сохранять их в объектной форме и далее работать с ними, используя объектно-ориентированные принципы.

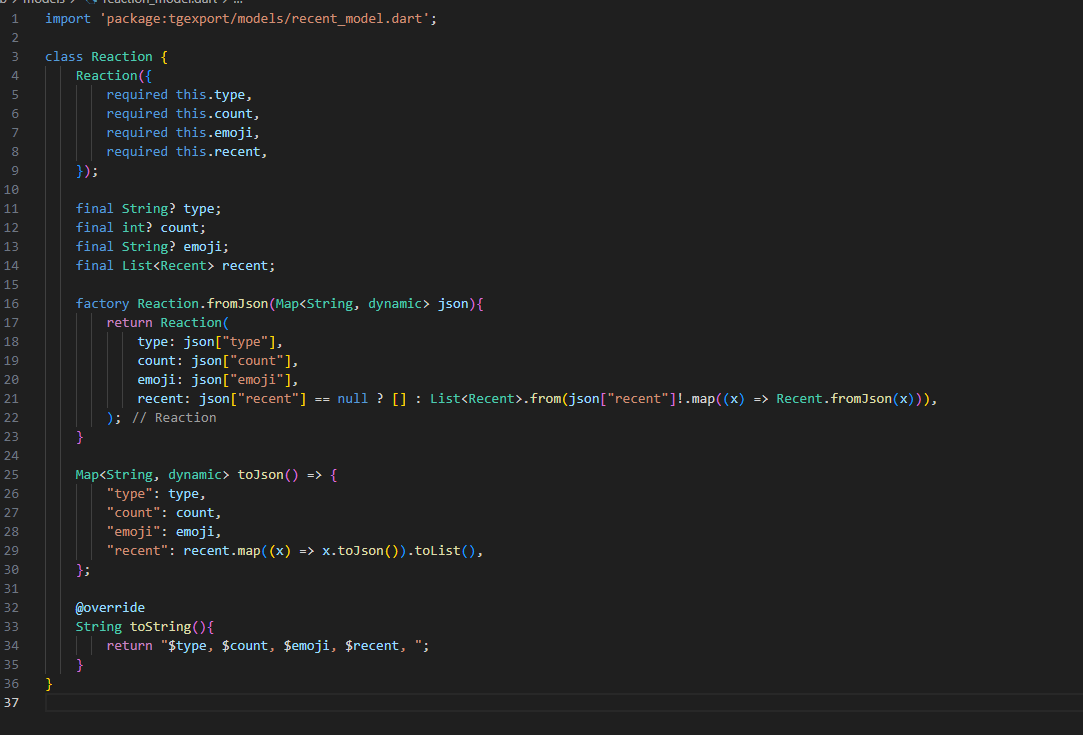


Рисунок 3. Модель класса Reaction

**2.3.2 Реализация отображения динамических данных**

Реализация отображения динамических данных является одной из ключевых задач в мобильной разработке, особенно в приложениях, работающих с большими объемами информации, как в случае с экспортированными чатами Telegram. В данном проекте для отображения данных использовался фреймворк **Flutter**, который предоставляет мощные инструменты для работы с динамическими списками, а также для создания отзывчивых и удобных интерфейсов.

В этом проекте динамическими данными являются сообщения в чатах, которые могут поступать в приложение по мере загрузки и обновления данных. Каждое сообщение может быть связано с медиафайлами, такими как изображения, видео или документы, а также с реакциями пользователей, которые могут изменяться в реальном времени. Задача заключалась в том, чтобы эффективно отображать все эти данные и обновлять интерфейс в ответ на изменения.  
Одним из основных инструментов, что я использовал для отображения динамических данных, является виджет **ListView**. Этот виджет идеально подходит для отображения списков, в том числе длинных и динамически обновляющихся списков.

В Flutter ListView может быть использован как для статических, так и для динамических данных. Для отображения сообщений чата мы использовали **ListView.builder**, который позволяет эффективно создавать элементы списка по мере необходимости. Это особенно важно для приложений, которые могут работать с большими объемами данных, так как этот подход позволяет значительно снизить нагрузку на систему.



Рисунок 4. Виджет ListView

Реализация отображения динамических данных в проекте, включая сообщения, медиафайлы и реакции, потребовала использования множества инструментов и методов для обеспечения плавного и отзывчивого интерфейса. Использование **ListView.builder** и **StreamBuilder** позволило эффективно отображать и обновлять данные, а обработка медиафайлов и реакций предоставила пользователю интуитивно понятный и богатый функционал для взаимодействия с чатом.

**2.4 Анализ моей работы**

Прежде чем приступить к работе над проектом, у меня не было практического опыта работы с **Flutter** и разработкой в целом. В начале работы над проектом я столкнулся с множественными трудностями, так как пришлось осваивать не только сам фреймворк, но и целый ряд связанных с ним технологий и концепций. Например, мне нужно было быстро изучить **виджеты Flutter**, управление состоянием, асинхронное программирование и работу с потоками данных. Я также столкнулся с необходимостью работы с **JSON**-форматом данных и взаимодействием с сетью. Ни с чем подобным ранее я не встречался, от того было огромное множество вопросов, на которые я постепенно находил ответы, хоть и не на все.

Так как основная часть работы в проекте связана с динамическим отображением данных, мне пришлось освоить такие темы, как **работа с потоками данных** с помощью **StreamBuilder**, оптимизация производительности с использованием **Lazy Loading**, а также взаимодействие с медиафайлами в реальном времени. Эти темы потребовали от меня не только изучения документации, но и регулярной практики, чтобы понять, как наилучшим образом интегрировать их в структуру приложения.

Из-за недостаточного опыта работы с фреймворком и программированием в целом, мне часто приходилось обращаться к руководителю проекта, Яну Черному, с просьбой разъяснить те или иные моменты. Ян всегда был готов помочь, объясняя не только конкретные технические вопросы, но и общее направление разработки. Он направлял меня на верный путь, объяснял, как правильно работать с различными инструментами, и помогал разобраться в тонкостях программирования на языке **Dart** и работы с **Flutter**.

Одной из ключевых задач, с которой я столкнулся, было создание моделей данных для чатов, сообщений, медиафайлов и реакций. Это важная часть проекта, так как правильная структура моделей данных позволяет обеспечить корректную работу приложения и эффективно организовать взаимодействие с внешними данными, такими как экспортированные чаты Telegram.  
 В процессе создания моделей я столкнулся с рядом технических сложностей:

* Необходимость тщательно продумать структуру данных, чтобы они соответствовали данным Telegram и могли легко адаптироваться к изменениям в структуре экспортируемых файлов.
* Важность правильного использования коллекций, таких как списки и карты (list, map), для хранения вложенных данных, например, для списка сообщений в чате или для хранения реакций на сообщения.
* Требование к реализации методов **fromJson** и **toJson** таким образом, чтобы они корректно обрабатывали все типы данных, включая вложенные объекты и массивы. Этот процесс потребовал внимательности и учета всех возможных нюансов при извлечении и записи данных.

Другой важной задачей, которая была мне поручена, являлась реализация отображения динамических данных. В условиях работы с экспортированными данными из Telegram, которые могут быть весьма объемными и изменяться в реальном времени, необходимость отображать данные динамически — это не просто требование, а важный аспект, который напрямую влияет на опыт пользователя.

Сложности, с которыми я столкнулся при реализации отображения динамических данных:

* Работа с большим количеством данных в реальном времени, что требует эффективной обработки и отображения.
* Обработка медиафайлов различных типов, что потребовало написания специализированных виджетов для каждого типа медиа.
* Обеспечение плавности и отзывчивости интерфейса при большом количестве сообщений и вложений.

Еще одной задачей было корректное отображение медиафайлов, что потребовало написания кастомных виджетов для их обработки и отображения. Я изучил различные способы работы с медиафайлами в Flutter и использовал сторонние библиотеки для работы с видео и аудио.  
 Моя работа над проектом была значительным шагом в развитии моих технических навыков и профессиональных компетенций. Задачи, поставленные передо мной, позволили мне не только углубить знания в области **Flutter**, но и научиться работать с данными, отображать их динамически и эффективно взаимодействовать с медиафайлами.

**2.5 Взаимодействие с командой**

Взаимодействие с командой было построено на регулярных встречах и обсуждениях прогресса. Мы делились результатами работы, обсуждали возникающие проблемы и решали их совместно. Каждому участнику были назначены задачи в зависимости от его опыта и квалификации. Присутствовало и совместное выполнение особо сложных задач.

**2.6 Взаимодействие с руководителем**

Наш руководитель, Ян Черный, всегда был на связи и готов помочь, отвечая на все возникающие вопросы. Каждый день, сверяясь с гитхабом, он проверял наши результаты, указывая на ошибки. Помимо этого, Ян регулярно организовывал персональные созвоны с каждым из нас, чтобы помочь разобраться с техническими аспектами работы. Он также присылал обучающие видео и материалы, что позволяло нам усваивать новые знания в удобном формате. Кроме того, он не ограничивался только официальными встречами, а часто проводил дополнительные созвоны в середине недели, чтобы помочь нам с возникающими трудностями и улучшить качество работы.

В течение рабочей недели, Ян периодически спрашивал об процессе выполнения задачи. Руководитель явно был заинтересован в выполнении работы и имел чёткое представление конечного варианта продукта, и смог внятно объяснить его всем участникам. Но, случались моменты, когда участники не успевали делать задачи и происходил застой работы, из-за чего работа и затянулась аж до последних дней сдачи проекта.

**3. ЗАКЛЮЧЕНИЕ**

В результате проделанной работы был создан прототип мобильного приложения для работы с экспортированными данными из Telegram. Приложение позволяет загружать и отображать чаты, фильтровать сообщения по различным критериям и экспортировать данные в удобном формате.

Разработка этого приложения позволила мне углубить знания в области разработки с использованием Flutter, а также в области работы с данными в формате JSON и их обработке. Познакомила с работой в команде с разными, как по возрасту, так и по опыту людьми.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* + 1. Официальная документация по Flutter: <https://flutter.dev>
    2. Статья по объектно-ориентированному программированию: <https://www.tutorialspoint.com/dart/dart_object_oriented.htm>
    3. Описание формата JSON: <https://www.json.org/json-en.html>
    4. Пособие по Dart <https://metanit.com/dart/tutorial/>
    5. Видеоуроки <https://youtu.be/rZcTaRU7AAw?si=eVQbmqR4Oq3ZQ8Sk>