

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование  
информационных систем

Тетин Илья Николаевич

# Создание мессенджера ВКонтакте на iOS

Курсовая работа

Научный руководитель:  
к. т. н., Брыксин Т. А.

Санкт-Петербург  
2020

# Оглавление

<b>Введение</b>	<b>3</b>
<b>Постановка задачи</b>	<b>4</b>
<b>1. Обзор</b>	<b>5</b>
1.1. Обзор существующих решений . . . . .	5
<b>2. Реализация</b>	<b>6</b>
2.1. Архитектура . . . . .	6
2.1.1. Окна с диалогами и сообщениями . . . . .	7
2.1.2. Отправка запросов . . . . .	8
2.2. Постепенная подгрузка сообщений . . . . .	8
2.3. База данных . . . . .	8
2.3.1. Сохранение диалогов . . . . .	9
2.3.2. Сохранение сообщений . . . . .	9
<b>Результаты</b>	<b>10</b>
<b>Список литературы</b>	<b>11</b>

# Введение

В наше время социальные сети стали неотъемлемой частью нашей жизни. В среднем люди тратят около 2 часов в день на социальные сети.<sup>1</sup> Из-за обилия развлекательной, поверхностной и зачастую ненужной информации время пребывания в социальной сети значительно увеличивается. Такое времяпрепровождение может отрицательно сказываться на нашем здоровье, т.к. большой объем информации нередко утомляет и нагружает нервную систему. Исследования показывают, что чтение большого количества информации способствует ухудшению краткосрочной памяти.<sup>2</sup>

При более эффективном способе использования социальных сетей у нас появляется дополнительное время, а также мы не перегружаем нашу память бесполезной информацией.

В качестве альтернативы мы можем использовать клиенты социальных сетей, где есть только раздел с сообщениями. Ведь в таком случае у нас просто не будет возможности отвлекаться на чтение ненужной информации. Необходимость в таком приложении можно подтвердить тем, что ВКонтакте выпустила мессенджер «VK Messenger», но данная версия была выпущена только для компьютеров.

Несмотря на актуальность проблемы, на рынке iOS приложений все мессенджеры ВКонтакте имеют ряд критических проблем, поэтому было принято решение самостоятельно разработать мобильную версию мессенджера ВКонтакте под iOS.

---

<sup>1</sup><https://www.broadbandsearch.net/blog/average-daily-time-on-social-media>

<sup>2</sup><https://www.sciencedaily.com/releases/2013/01/130109124227.htm>

# Постановка задачи

Целью данной работы является создание клиента ВКонтакте, в котором будет только раздел с сообщениями под iOS. Для достижения цели были поставлены следующие задачи:

1. Изучить программный интерфейс сервисов ВКонтакте<sup>1</sup>.
2. Разработать архитектуру приложения.
3. Разработать графический интерфейс приложения.
4. Реализовать получение и отображение диалогов и сообщений.
5. Реализовать получение новых сообщений с помощью длительных запросов.

---

<sup>1</sup>[https://vk.com/dev/api\\_requests](https://vk.com/dev/api_requests)

# 1. Обзор

## 1.1. Обзор существующих решений

На рынке мобильных приложений iOS есть только 4 клиента ВКонтакте<sup>1-4</sup>, где есть только раздел с сообщениями, но ни один из них нельзя назвать рабочим, так как приложения не адаптированы под новую версию iOS, что делает их бесполезными.

---

<sup>1</sup><https://apps.apple.com/ru/app/id891605076>

<sup>2</sup><https://apps.apple.com/ru/app/id896658711>

<sup>3</sup><https://apps.apple.com/ru/app/id873637150>

<sup>4</sup><https://apps.apple.com/ru/app/id873326902>



### 2.1.1. Окна с диалогами и сообщениями

<b>DialogsViewController</b>
- messagesViewControllers: [MessagesViewController] - conversations: [Conversation] - interactor: DialogsInteractor
- loadSavedConversations(): Void - addNewConversation(conversation: Conversation): Void - displayConversations(conversations: [Conversation]) - displayNewMessages(messages: [Messages]) - deleteConversations(): Void

Рис. 2: Диаграмма класса

<b>Conversation</b>
unreadMessages: Int photoUrlString: String name: String message: String lastMessageId: Int id: Int date: String conversationMessageId: Int

Рис. 3: Представление диалога в приложении

В классе DialogsViewController реализована логика отображения всех бесед и обработки новых событий. Получение новых событий реализовано с помощью длительных запросов<sup>1</sup>. MessagesViewController отвечает за отображение сообщений в диалоге. Для каждого диалога у нас создан отдельный класс MessagesViewController. При получении нового сообщения находим MessageViewController, который представляет диалог, в который пришло новое сообщение, и в этом классе добавляем новое сообщение, а также обновляем поле lastMessageId и увеличиваем значение unreadMessages у этой беседы.

<b>MessagesViewController</b>
- messagesViewControllers: [MessagesViewController] - Messages: [Message] - conversation: Conversation - interactor: MessagesInteractor
- loadSavedMessages(): Void - addNewMessages(messages: [Messages]) - deleteLastMessageFromData(): Void

Рис. 4: Диаграмма класса

---

<sup>1</sup>[https://vk.com/dev/using\\_longpoll](https://vk.com/dev/using_longpoll)

### 2.1.2. Отправка запросов

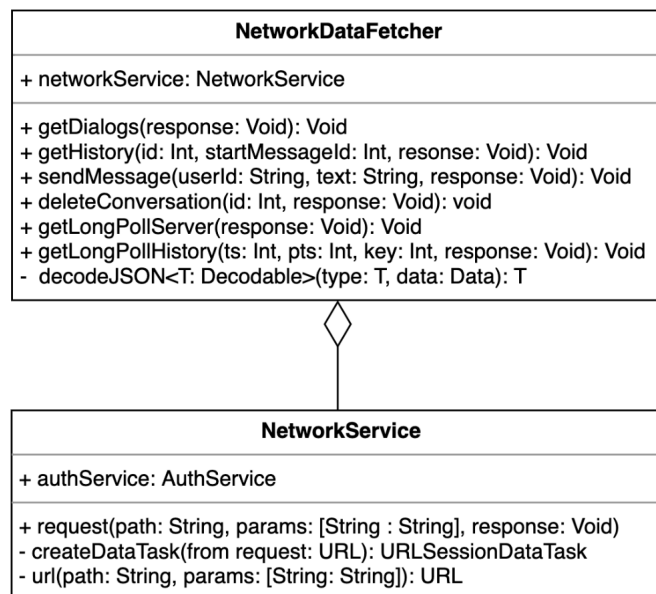


Рис. 5: Диаграмма классов отвечающих составление и отправку запросов

За составление и отправку запросов отвечают классы **NetworkDataFetcher** и **NetworkService**. Отправка запросов осуществляется асинхронно, что позволяет открывать окно, до загрузки данных. В качестве ответа на запрос приходят данные в формате JSON, которые с помощью метода `decodeJSON`, преобразовываем в удобный нам формат. В качестве параметра к каждому методу мы передаем функцию, которая будет вызывать метод у **Presenter**, который подготовит эти данные к отображению.

## 2.2. Постепенная подгрузка сообщений

В целях экономии ресурсов пользователя было решено реализовать постепенную подгрузку сообщений. При пролистывании половины экрана, отправляется запрос на получение следующих сообщений.

## 2.3. База данных

В целях экономии ресурсов пользователя и возможности использовать приложение в офлайн режиме было решено реализовать возмож-



ность сохранение диалогов и сообщений в базе данных.

### **2.3.1. Сохранение диалогов**

При запуске приложения мы загружаем из базы данных сохранённые диалоги и после этого отправляем запрос на сервер ВКонтакте на получение 20 диалогов. Это необходимо в связи с тем, что, пока мы не заходили в приложение, могли появиться новые беседы, и, не отправляя запрос, мы просто не сможем узнать о том, что появилась новая беседа.

### **2.3.2. Сохранение сообщений**

При открытии диалога мы сравниваем id последнего сохраненного сообщения с id последнего сообщения в диалоге. Если они отличаются, тогда мы отправляем запрос на получение новых сообщений.

В каждом диалоге будем хранить только 20 последних сообщений. При получении нового сообщения мы удаляем из базы данных последнее в списке сообщение и сохраняем новое.

# Результаты

В ходе выполнения данной работы были достигнуты следующие результаты:

1. Получен опыт работы с программным интерфейсом сервисов Вконтакте.
2. Получен опыт разработки iOS приложений.
3. Разработана архитектура приложения.
4. Разработан графический интерфейс приложения.
5. Реализовано получение и отображение диалогов и сообщений.
6. Реализовано получение новых сообщений с помощью длительных запросов.

## Список литературы

- [1] SwiftBook. Изучаем Основы Core Data. — 2018. — URL: <https://swiftbook.ru/contents/core-data-for-beginners/>.
- [2] SwiftBook. Курс по GCD (Grand Central Dispatch). — 2018. — URL: <https://swiftbook.ru/contents/grand-central-dispatch-in-swift4>.
- [3] Yandex. Видео лекции Разработка под iOS. — 2019. — URL: <https://yandex.ru/online-courses/ios-course>.