



Факультет Компьютерных Наук
Департамент Программной Инженерии

Rewind

Приложение для коллективного медиа-потребления

Application for Collective Media Consumption

Выполнили:

Студент группы БПИ228

Рахманов Данила Дмитриевич

Студентка группы БПИ229

Хрулева Александра Денисовна

Руководитель:

Приглашенный преподаватель Департамента Программной Инженерии

Сосновский Григорий Михайлович

Москва 2024

Проблема и актуальность

Цель

Разработать приложение, которое позволит друзьям, членам семьи, коллегам по работе и другим знакомым между собой людям вспомнить яркие моменты их общения

Задачи

- Исследование и анализ аналогов
- Выбор архитектуры приложения
- Реализация сетевого слоя
- Генерация изображения для цитаты
- Добавление участников в группу
- Шифрование ID группы
- Автоматическая генерация тегов для фотографий

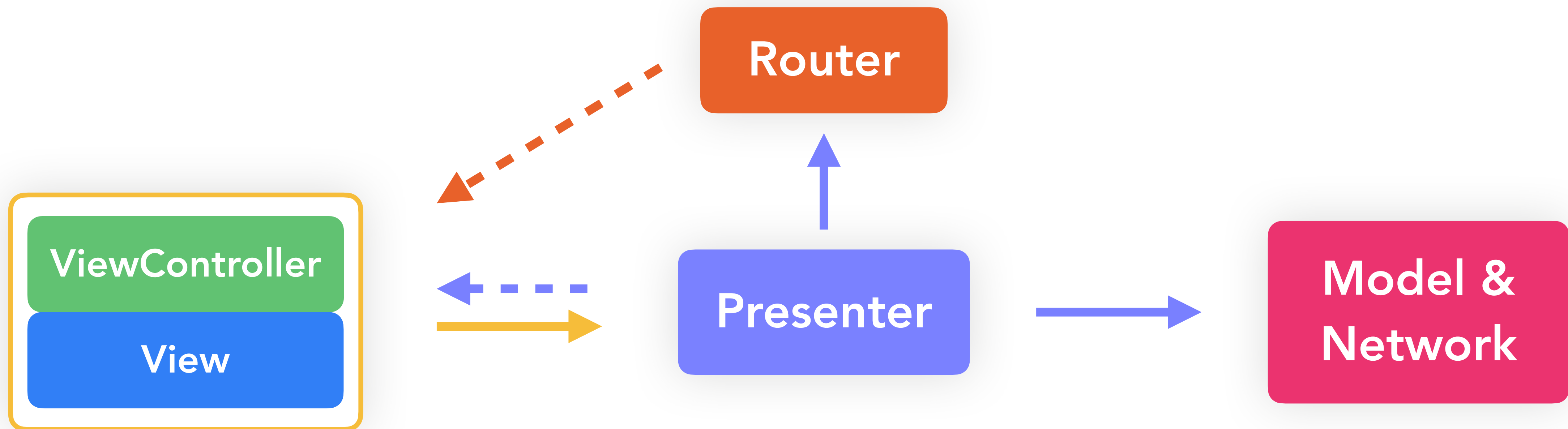
Исследование и анализ аналогов

В ходе исследования были выявлены 3 прямых и 4 косвенных аналога
Сравнительный анализ описан в документации



Выбор архитектуры приложения

MVP + R



Реализация сетевого слоя

NetworkService

```
final class NetworkService
```

Topics

Initializers

```
init()
```

Type Methods

```
static func addMemberToGroup(groupId: Int, userId: Int, completion: (NetworkResponse) -> Void)
```

```
static func addTagToMedia(mediaId: Int, tagTexts: [String], completion: (NetworkResponse) -> Void)
```

```
static func checkEmailToLogin(email: String, completion: (NetworkResponse) -> Void)
```

```
static func createGroup(ownerId: Int, groupName: String, completion: (NetworkResponse) -> Void)
```

В NetworkService
написаны **34** метода

Реализация сетевого слоя

NetworkResponse

```
struct NetworkResponse
```

Topics

Initializers

```
init(success: Bool, statusCode: Int?, message: String?, json: [String : Any]?, jsonArray: [[String : Any]]?)
```

Instance Properties

```
var json: [String : Any]?  
var jsonArray: [[String : Any]]?  
var message: String?  
var statusCode: Int?  
var success: Bool
```

Реализация сетевого слоя

processStringResponse(data:response:error:)

```
static func processStringResponse(  
    data: Data?,  
    response: URLResponse?,  
    error: Error?  
) -> NetworkResponse
```

processJSONResponse(data:response:error:)

```
static func processJSONResponse(  
    data: Data?,  
    response: URLResponse?,  
    error: Error?  
) -> NetworkResponse
```

processJSONArrayResponse(data:response:error:)

```
static func processJSONArrayResponse(  
    data: Data?,  
    response: URLResponse?,  
    error: Error?  
) -> NetworkResponse
```

Генерация изображения для цитаты

QuoteImageGenerator

```
final class QuoteImageGenerator
```

Topics

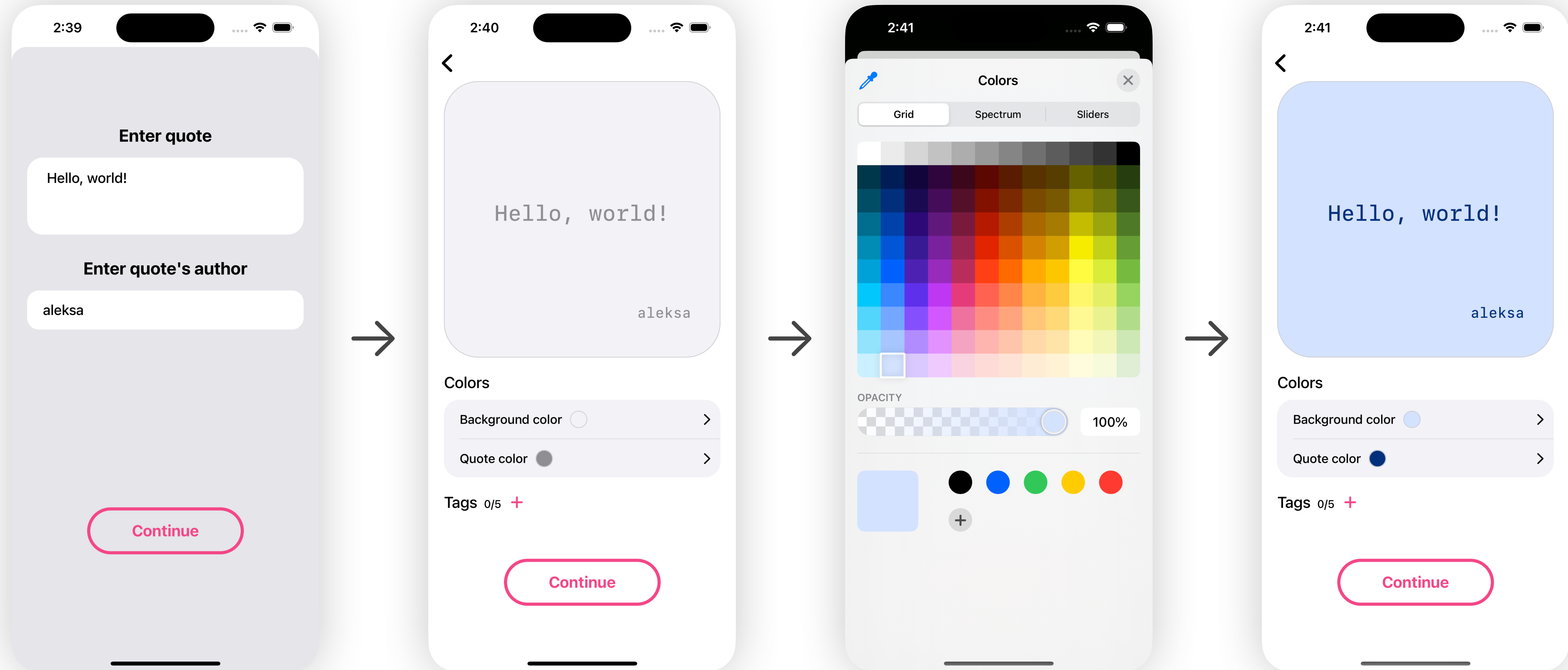
Initializers

```
init()
```

Type Methods

```
static func convertToImage(quote: String, author: String, backgroundColor: UIColor, text  
Color: UIColor) -> UIImage?
```

Генерация изображения для цитаты



Добавление участников в группу

JoinLinkService

```
final class JoinLinkService
```

Topics

Initializers

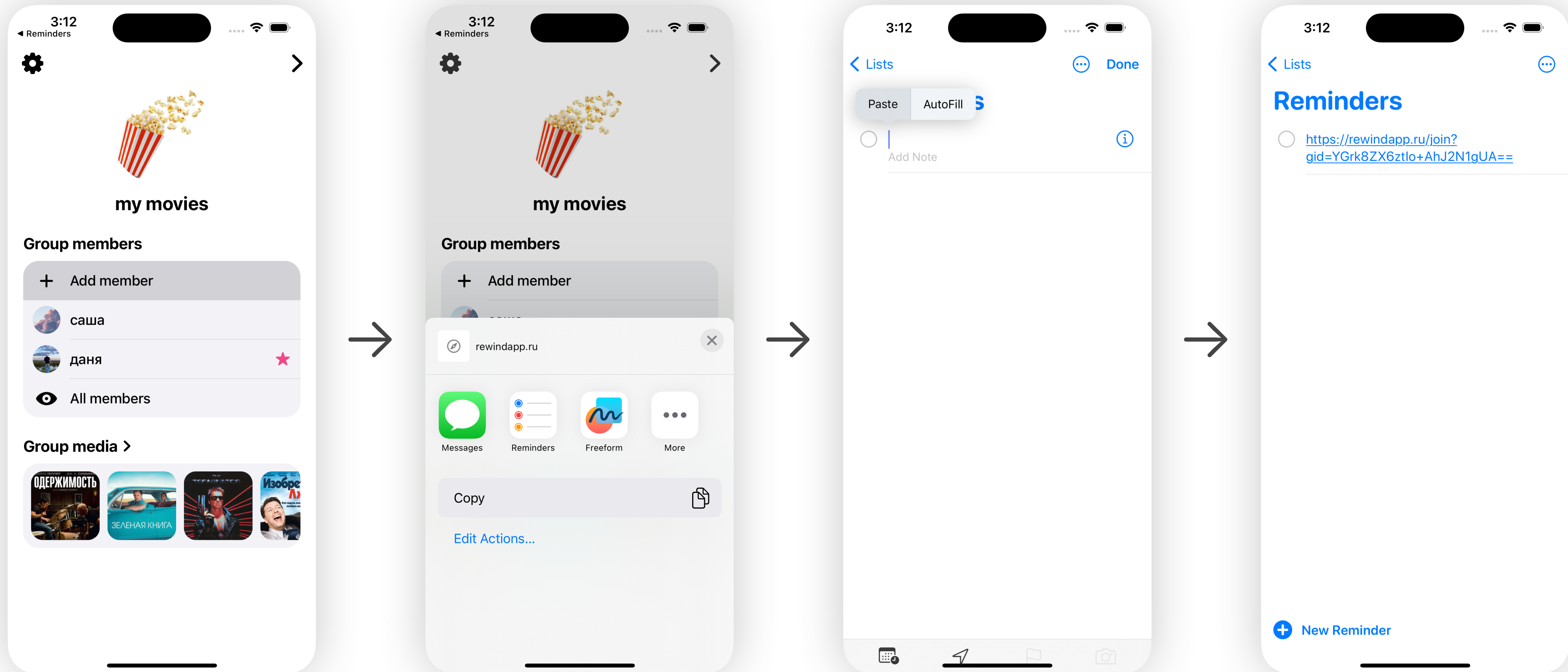
```
init()
```

Type Methods

```
static func createJoinLink(groupId: Int) -> URL?
```

```
static func decrypt(encryptedString: String) -> Int?
```

Добавление участников в группу



Шифрование ID группы

- ✗ <https://rewindapp.ru/join?gid=106>
- ✓ <https://rewindapp.ru/join?gid=jRPYGMxCg1YMqDa+63oXSA==>

Используемый алгоритм шифрования — AES (Advanced Encryption Standard)

```
private static func encrypt(data: Data, key: String) -> Data? {
    let keyData = key.data(using: .utf8)!
    let inputData = data as NSData
    let encryptedData = NSMutableData(length: Int(inputData.length) + kCCBlockSizeAES128)!
    let keyLength = size_t(kCCKeySizeAES128)
    let operation = CCOperation(kCCEncrypt)
    let algorithm = CCAAlgorithm(kCCAlgorithmAES)
    let options = CCOptions(kCCOptionPKCS7Padding)

    var numBytesEncrypted: size_t = 0

    let cryptStatus = CCCrypt(operation, algorithm, options, (keyData as NSData).bytes, keyLength, nil,
                              inputData.bytes, inputData.length, encryptedData.mutableBytes, encryptedData.length, &numBytesEncrypted)

    if cryptStatus == kCCSuccess {
        encryptedData.length = Int(numBytesEncrypted)
        return encryptedData as Data
    }

    return nil
}
```

Автоматическая генерация тегов

Framework

Core ML

Integrate machine learning models into your app.

iOS 11.0+

iPadOS 11.0+

macOS 10.13+

Mac Catalyst 13.0+

tvOS 11.0+

watchOS 4.0+

visionOS 1.0+

Overview

Use Core ML to integrate machine learning models into your app. Core ML provides a unified representation for all models. Your app uses Core ML APIs and user data to make predictions, and to train or fine-tune models, all on a person's device.



Core ML model




Core ML



Your app

Автоматическая генерация тегов

 Developer

News

Discover

Design

Develop

Distribute

Support

Account

Machine Learning

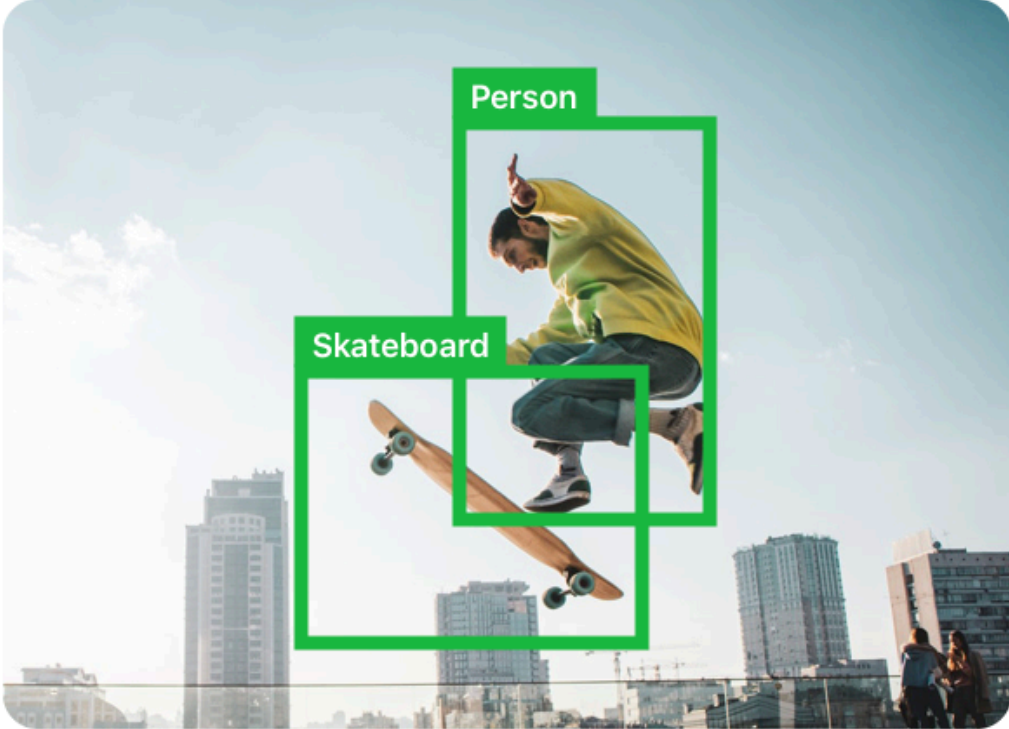
OverviewCore MLCreate MLAPIsModelsResources

Core ML Models

Build intelligence into your apps using machine learning models from the research community designed for Core ML.

ImagesText

Models are in Core ML format and can be integrated into Xcode projects. You can select different versions of models to optimize for sizes and architectures.



YOLOv3

Object Detection

Locate and classify 80 different types of objects present in a camera frame or image.

[View Models and Code Sample](#)

Автоматическая генерация тегов

YOLOv3Model

```
final class YOLOv3Model
```

Topics

Initializers

```
init(model: VNCoreMLModel)
```

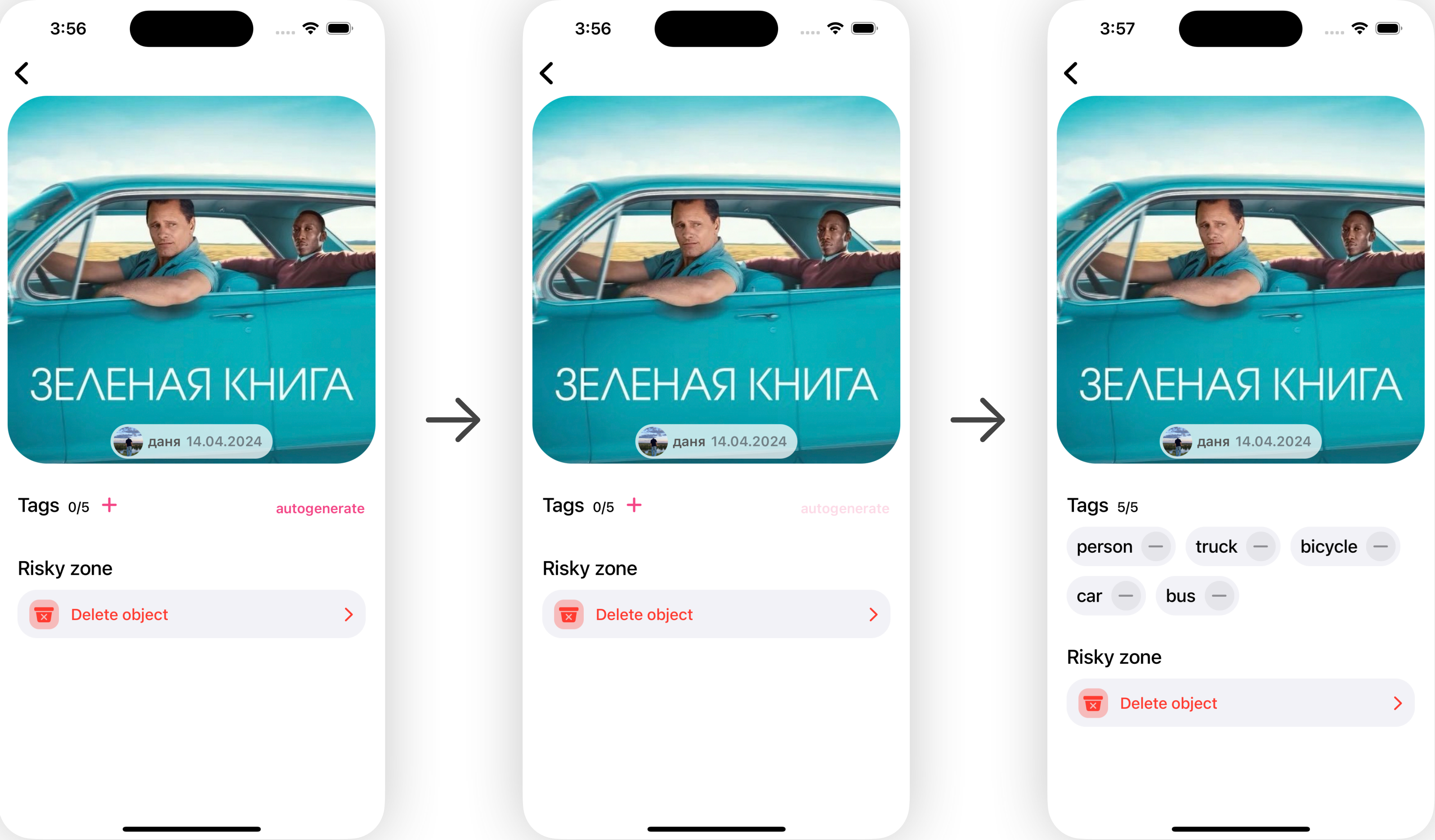
Instance Properties

```
let model: VNCoreMLModel
```

Instance Methods

```
func performObjectDetection(with: Data, completion: ([VNRecognizedObjectObservation]?,  
Error?) -> Void)
```


Автоматическая генерация тегов



Результаты

Выполнены следующие задачи:

- Исследование и анализ аналогов
- Выбор архитектуры приложения
- Реализация сетевого слоя
- Генерация изображения для цитаты
- Добавление участников в группу
- Шифрование ID группы
- Автоматическая генерация тегов для фотографий

Планы на будущее

- Добавить в проект виджеты
- Реализовать возможность добавления видео в галерею
- Реализовать возможность выбора отображаемой области фото и видео
- Добавить различные темы заднего фона
- Реализовать навигацию между экранами с помощью свайпов

Демонстрация

Спасибо за внимание!



Факультет Компьютерных Наук
Департамент Программной Инженерии

Rewind

Приложение для коллективного медиа-потребления

Application for Collective Media Consumption

Выполнили:

Студент группы БПИ228

Рахманов Данила Дмитриевич

Студентка группы БПИ229

Хрулева Александра Денисовна

Руководитель:

Приглашенный преподаватель Департамента Программной Инженерии

Сосновский Григорий Михайлович

Москва 2024