

SWIFTUI ДЛЯ БОЛЬШОГО B2C ПРОДУКТА

спойлер: было весело



Полина Скалкина
iOS разработчик

📍 @bypolinafox

DOUBLETAPP.

DOUBLETAPP.

+

Яндекс  Путешествия

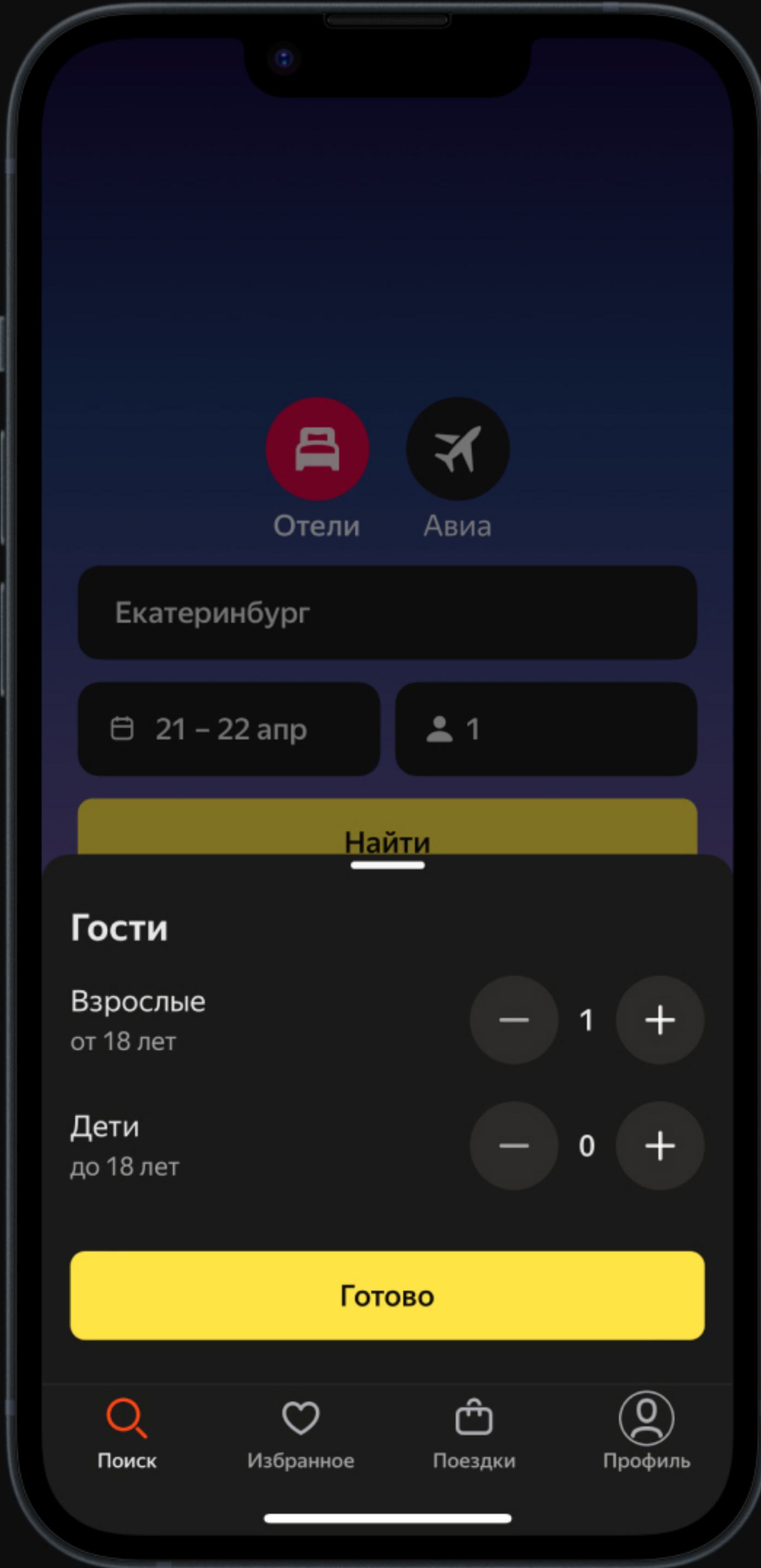
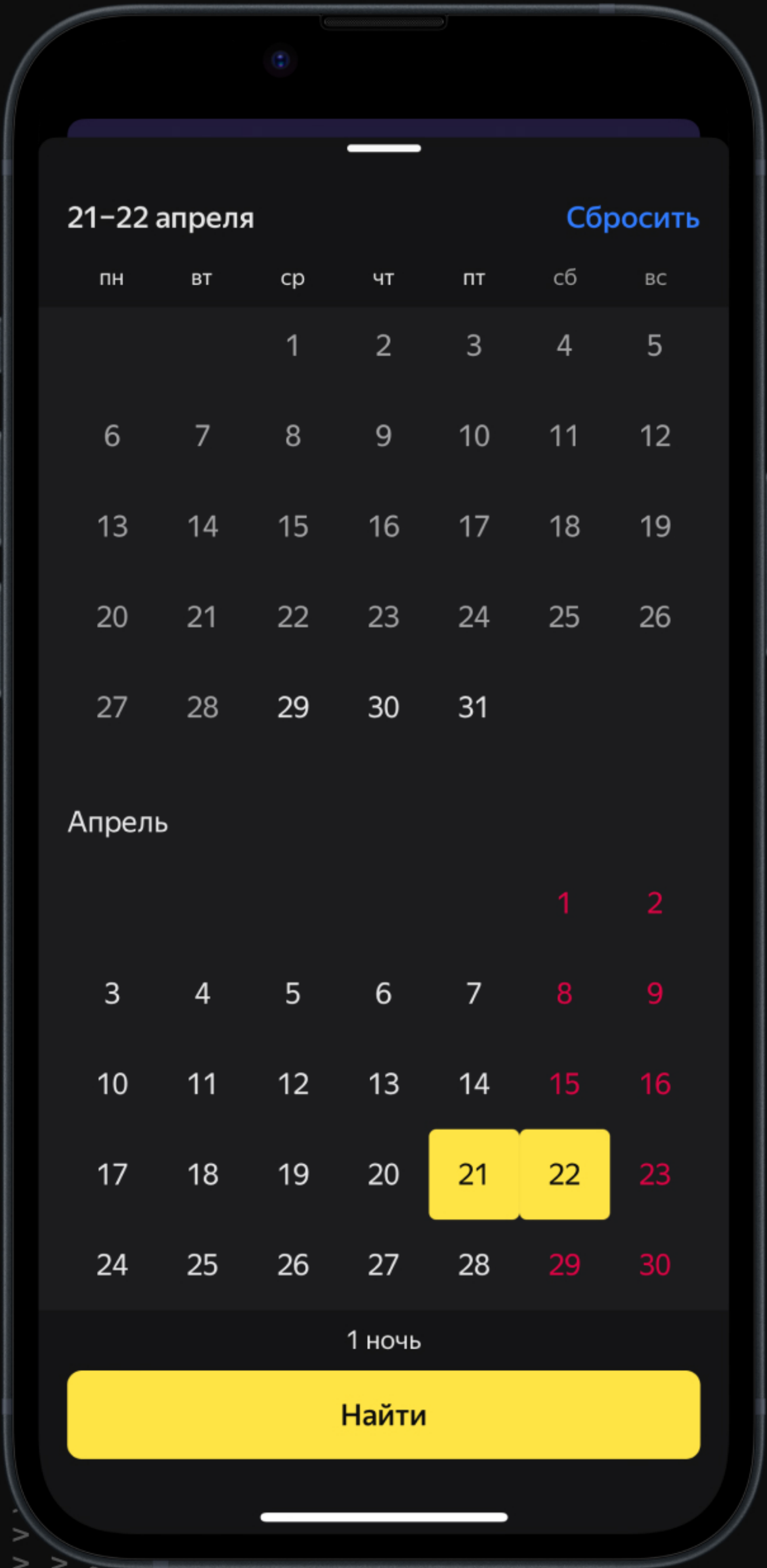
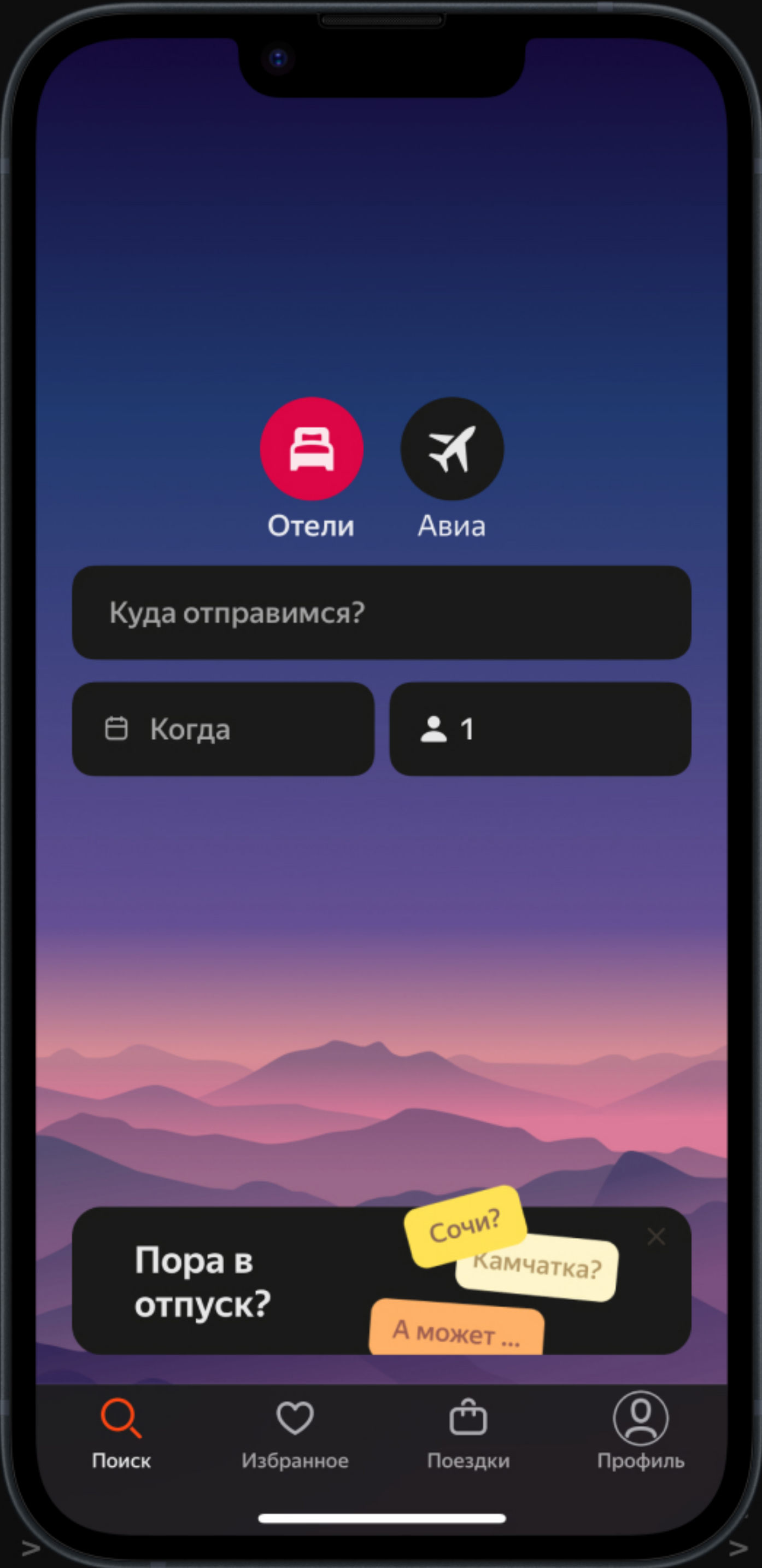
=



МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ЯНДЕКС ПУТЕШЕСТВИЙ



Путешествия




Екатеринбург

21 – 22 апр · 1

Популярные ▾

Цена ▾

Беспла



Маринс Парк Отель

Екатеринбург 3*

4,6

2934 отзыва · Гостиница

📍

Екатеринбург · ↗ 2,1 км до центра

Бесплатная отмена

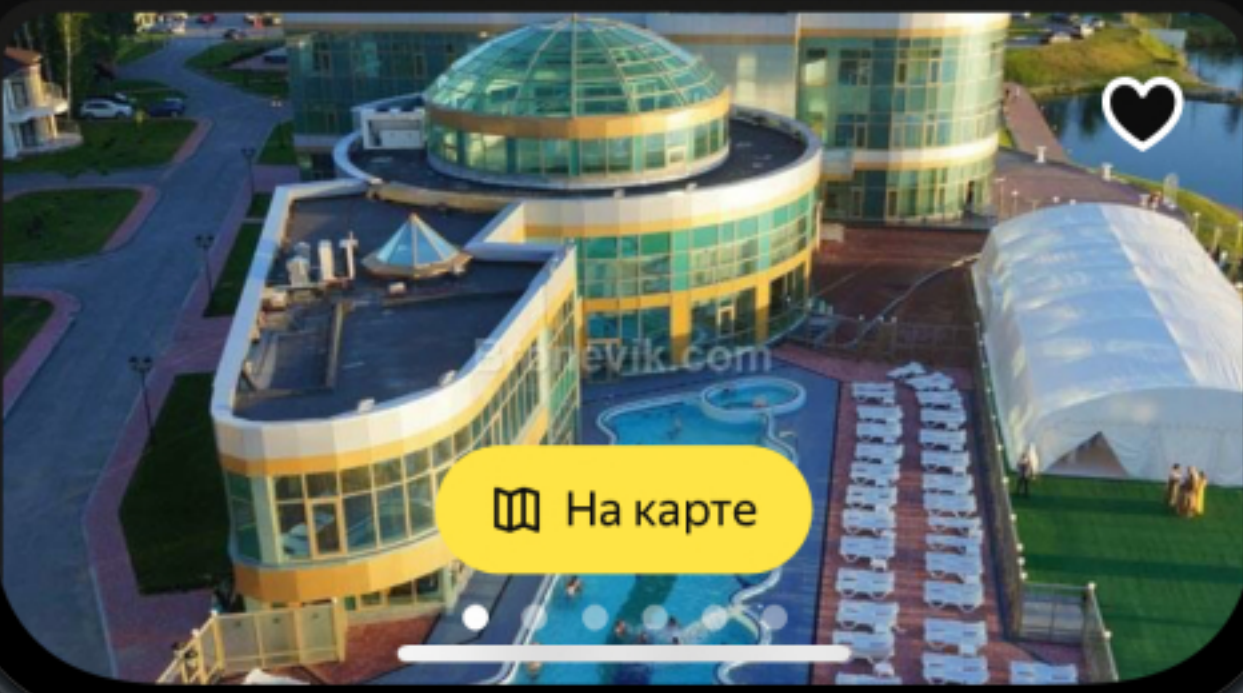
2 900 ₽

1 ночь

🔥 290

баллов в Плюс

2 900 ₽ / ночь



На карте

Фильтры

Готово

Бесплатная отмена

С завтраком

Цена за все ночи

от 0 ₽

до 10 000+ ₽

Питание

>

Звёздность

>

Рядом

>

Удобства

>


Рейтинг

>

До центра

>

Применить



Маринс Парк Отель
Екатеринбург 3*

4,6 2935 отзывов

Читать >

Wi-FiБассейнПарковкаКо

2,1 км до центра
Уральская 570 м

Все удобства в отеле >

Свердловская область, Екатеринбург,
улица Челюскинцев, 106

Выбрать номер от 2 900 ₽

Отзывы об отеле

Заккрыть

Маринс Парк Отель
Екатеринбург 3*

4,6 92% гостей понравилось
расположение
2935 отзывов

Расположение 92%

Чистота номеров 90%

Завтрак 87%

Качество WiFi 86%

Обслуживание 86%

Цена/качество 82%

Сначала популярные ▾ завтраки хорошая г

Александра Самохина
06.12.2022

★★★★★

В отеле номера на заявленные 3
звезды. В номерах есть всё
необходимое. Персонал доброжелате...

Читать полностью ▾

Ирина Никитина

Маринс Парк Отель Ека...
21 – 22 апр Гостей: 1

Стандарт с одной кроватью

Площадь: 15 м²
1 односпальная кровать

Удобства

Бесплатный Wi-Fi
Холодильник

2 900 ₽
Без питания
Бесплатная отмена ?

3 500 ₽
Завтрак вк
Бесплатная

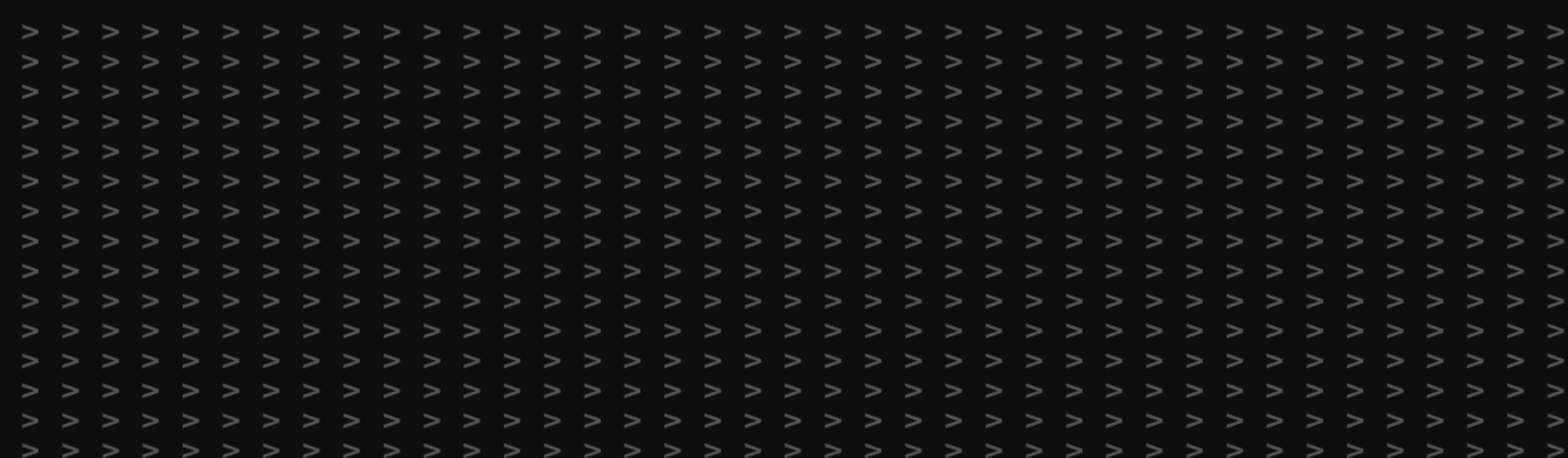
Выбрать

ФАКТЫ ПРО ПРИЛОЖЕНИЕ



ФАКТЫ ПРО ПРИЛОЖЕНИЕ

- Работа над проектом велась полтора года



ФАКТЫ ПРО ПРИЛОЖЕНИЕ

- Работа над проектом велась полтора года
- Архитектуру нужно было закладывать с умом



ФАКТЫ ПРО ПРИЛОЖЕНИЕ

- Работа над проектом велась полтора года
- Архитектуру нужно было закладывать с умом
- Будет много разношерстных пользователей



ФАКТЫ ПРО ПРИЛОЖЕНИЕ

- Работа над проектом велась полтора года
- Архитектуру нужно было закладывать с умом
- Будет много разношерстных пользователей
- Без внешних зависимостей — никак



DOUBLETAPP.

ДА ПОЧЕМУ



ОН



А НЕ Я

SWIFTUI



SWIFTUI

ВЫШЕЛ

DOUBLETAPP.

SWIFTUI

ВЫШЕЛ

4

DOUBLETAPP.

SWIFTUI

ВЫШЕЛ

4 ГОДА

SWIFTUI

ВЫШЕЛ

4 ГОДА

НАЗАД

Apple признала проблему с полосками на экранах iPhone 14 Pro и Pro Max. Решение скоро будет

Его уже готовят

Компания Apple признала-таки проблему с жёлто-зелёными полосками на экранах iPhone 14 Pro и Pro Max. Они видны при включении или разблокировке смартфона. Но, к счастью, это не аппаратный сбой, так что ремонтировать устройства не нужно.



AppleInsider.ru

5 проблем iOS 16, которые до сих пор возникают на iPhone

Темы > iOS • Кирилл Пироженко • 22.01.2023, обновлено 13.03.2023

Проблемы с Xcode 14

Вопрос задан 3 месяца назад Изменён 3 месяца назад Просмотров 59 раз



0



После обновления до MacOS Ventura обновился и мой Xcode до 14 версии. Проблема в том, что с ним невозможно работать. Каждое 3-е действие приводит к крашу IDE, а аналогов так и не придумали. Кто и как справляется с этой проблемой? Вот с какими проблемами встречаюсь я: п

«**SwiftUI** — не плохо
и не хорошо. Просто есть свои
особенности, которые стоит
учитывать, если вы хотите тащить
его в **большой проект**»

-Master Oogway



ПЛЮСЫ SWIFTUI



Скорость

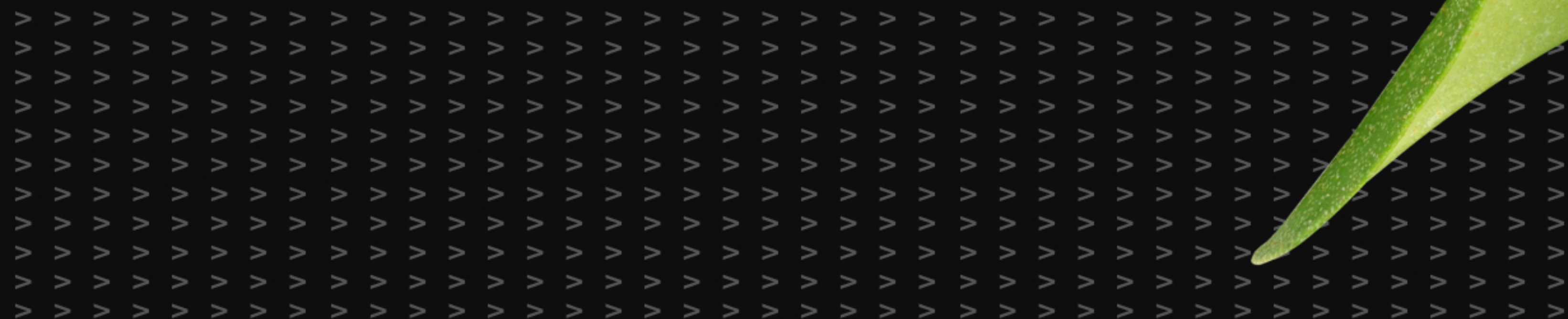


Симбиоз
с Comine

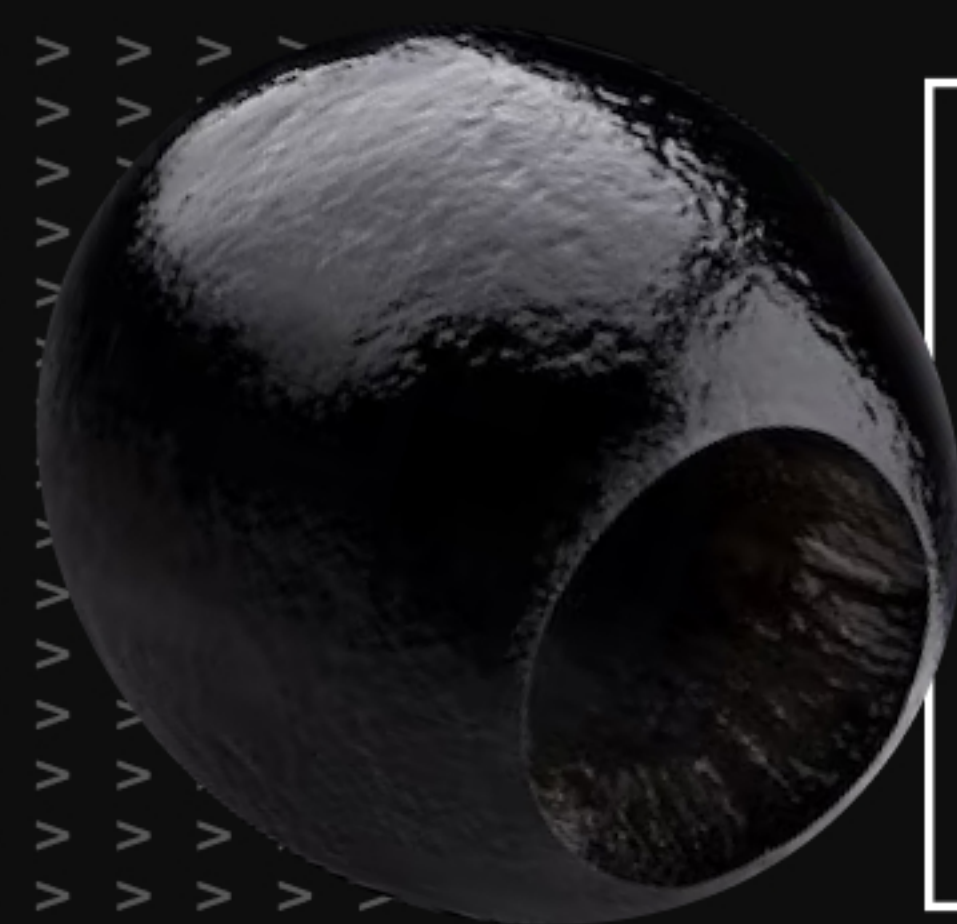


Хайповость
и рекрутабельность

ПОЛОВИМ МАСЛИН ВМЕСТЕ?



АРХИТЕКТУРА



No 1

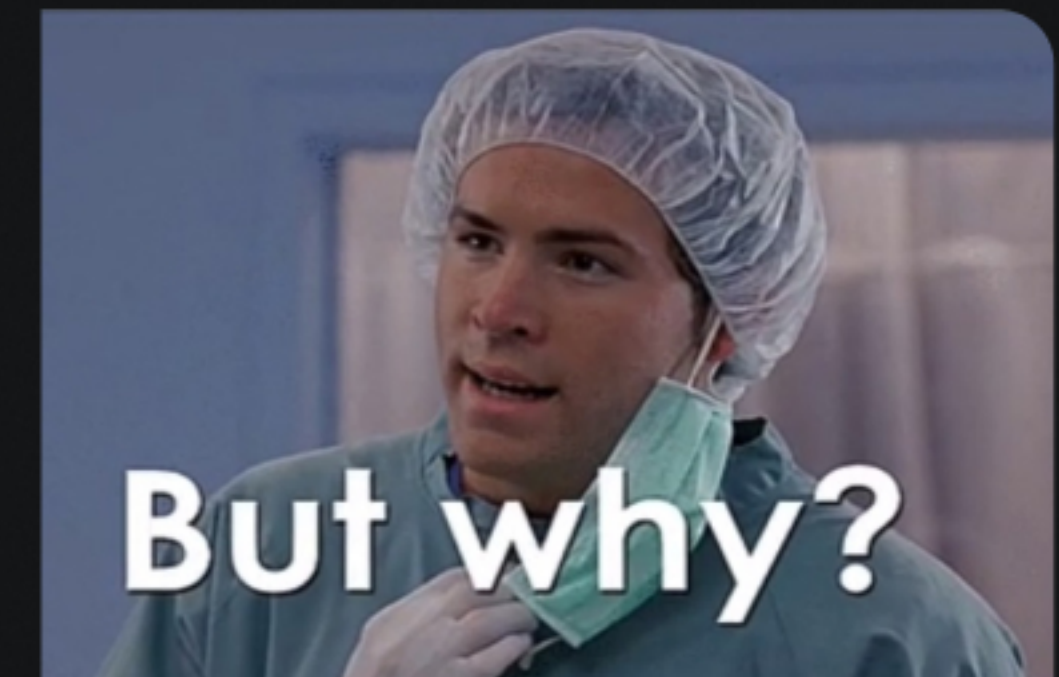
SWIFTUI + MVVM

SwiftUI

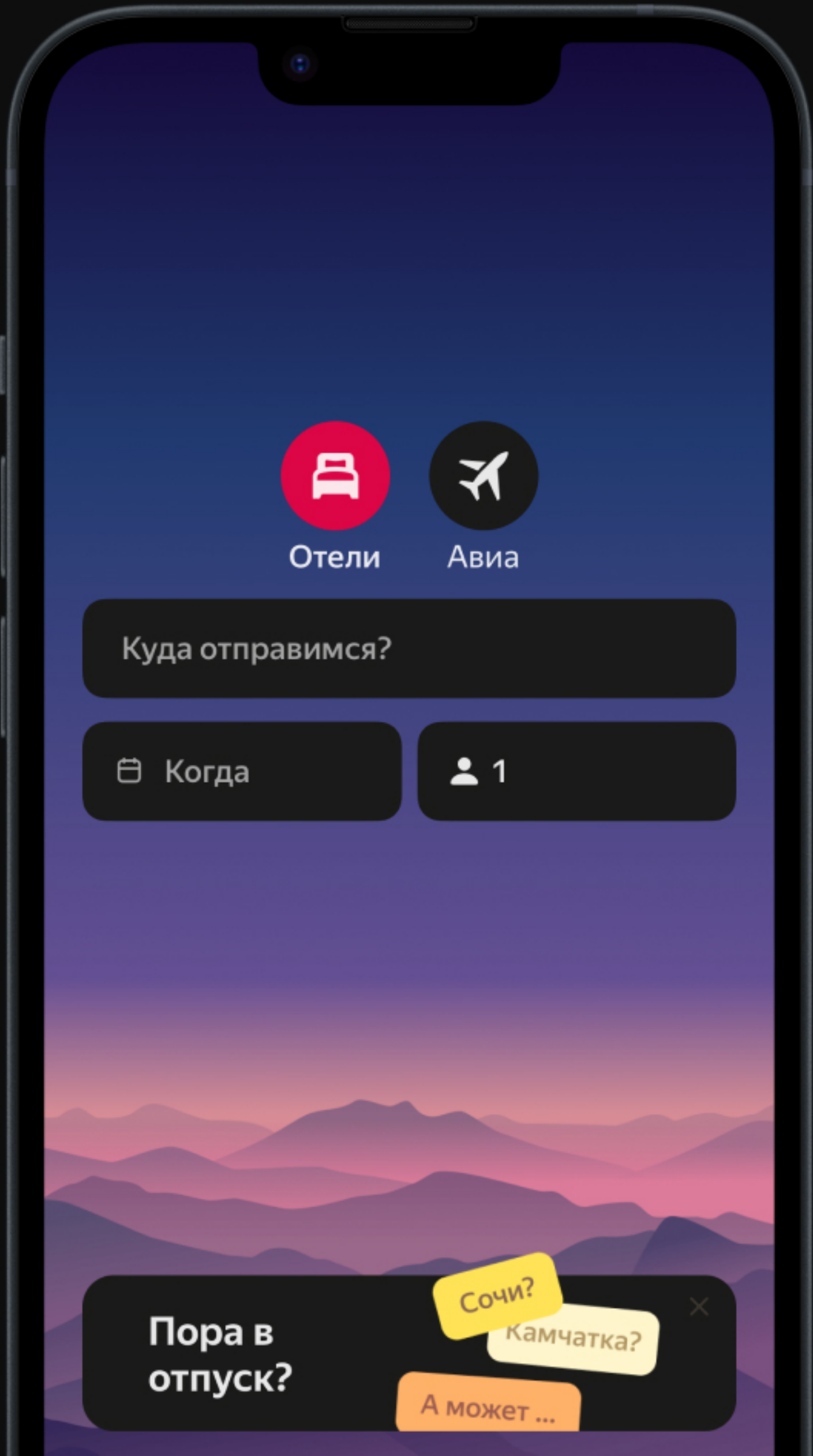


SwiftUI

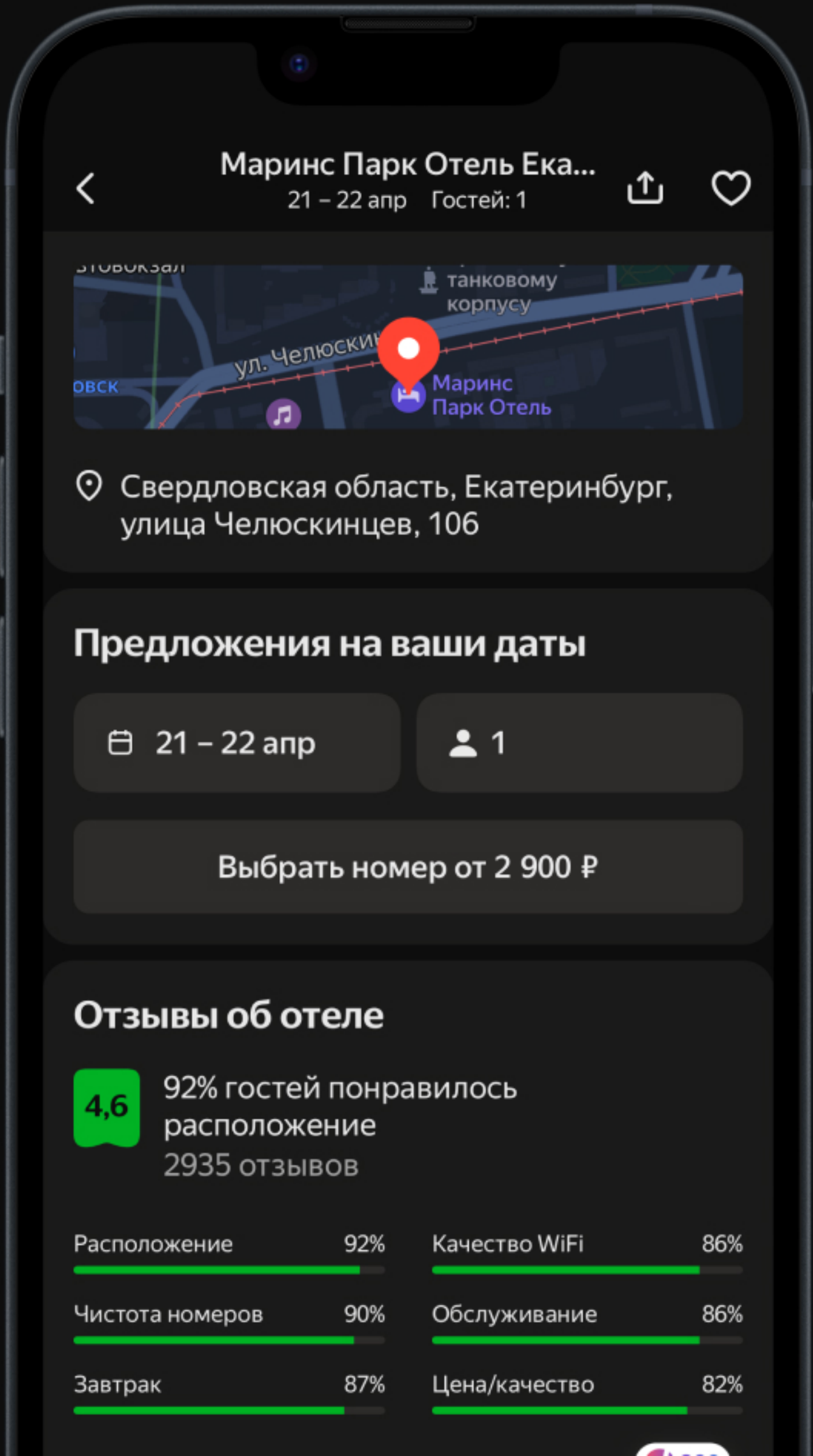
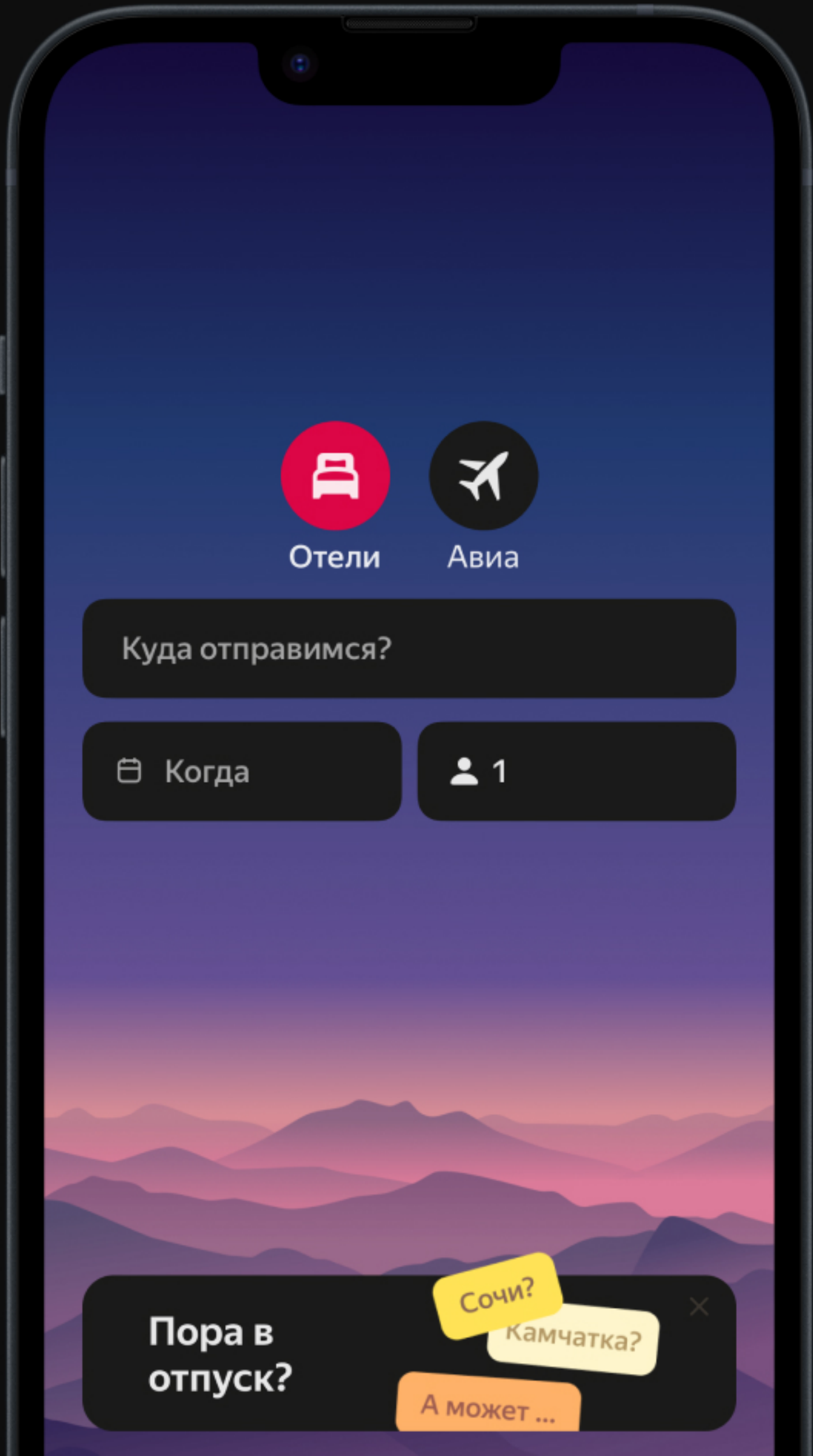
with Controllers or ViewModels



ГДЕ ЭТО НАС ПОГУБИЛО 🦴



ГДЕ ЭТО НАС ПОГУБИЛО 🦴





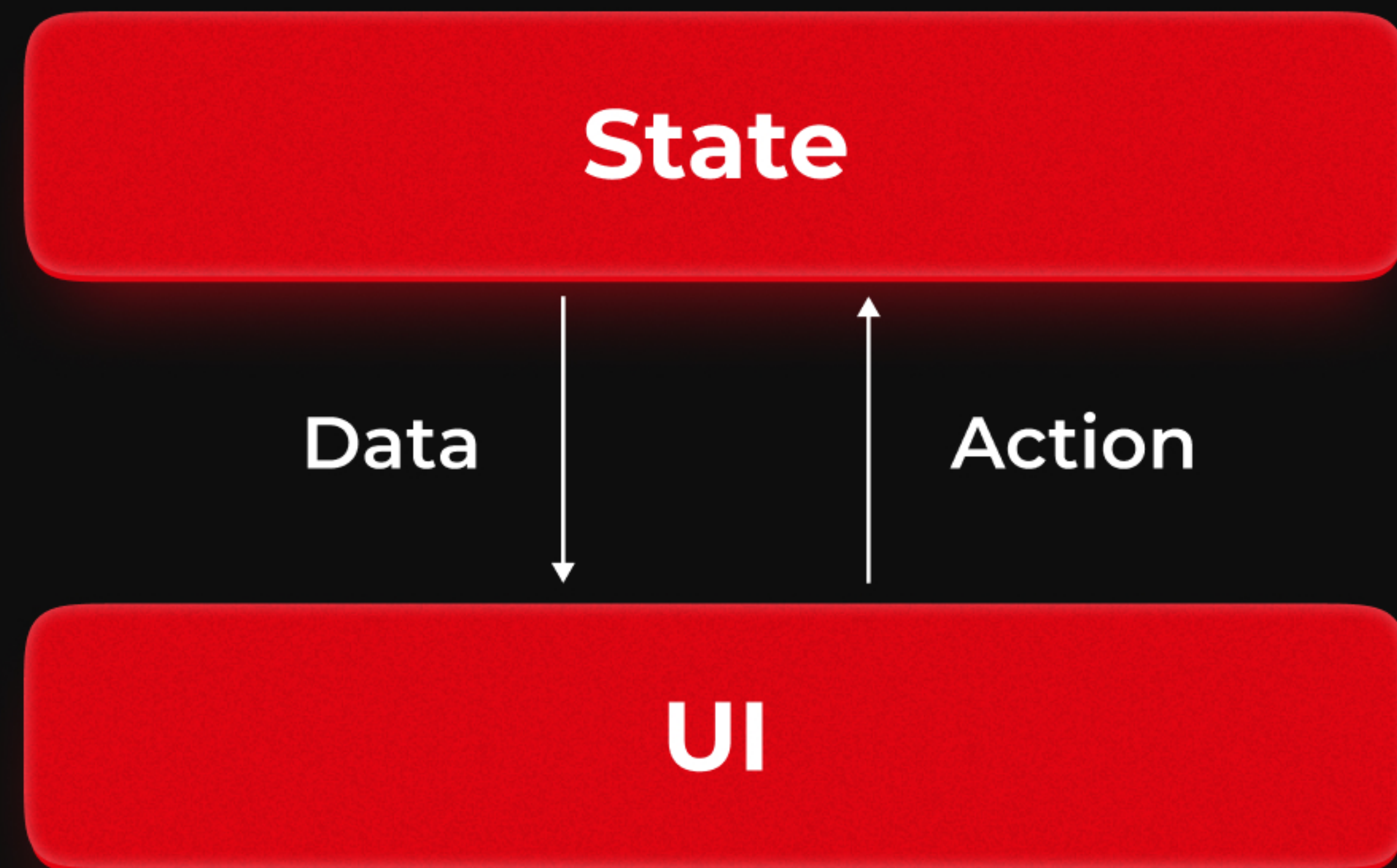
Redux



Redux

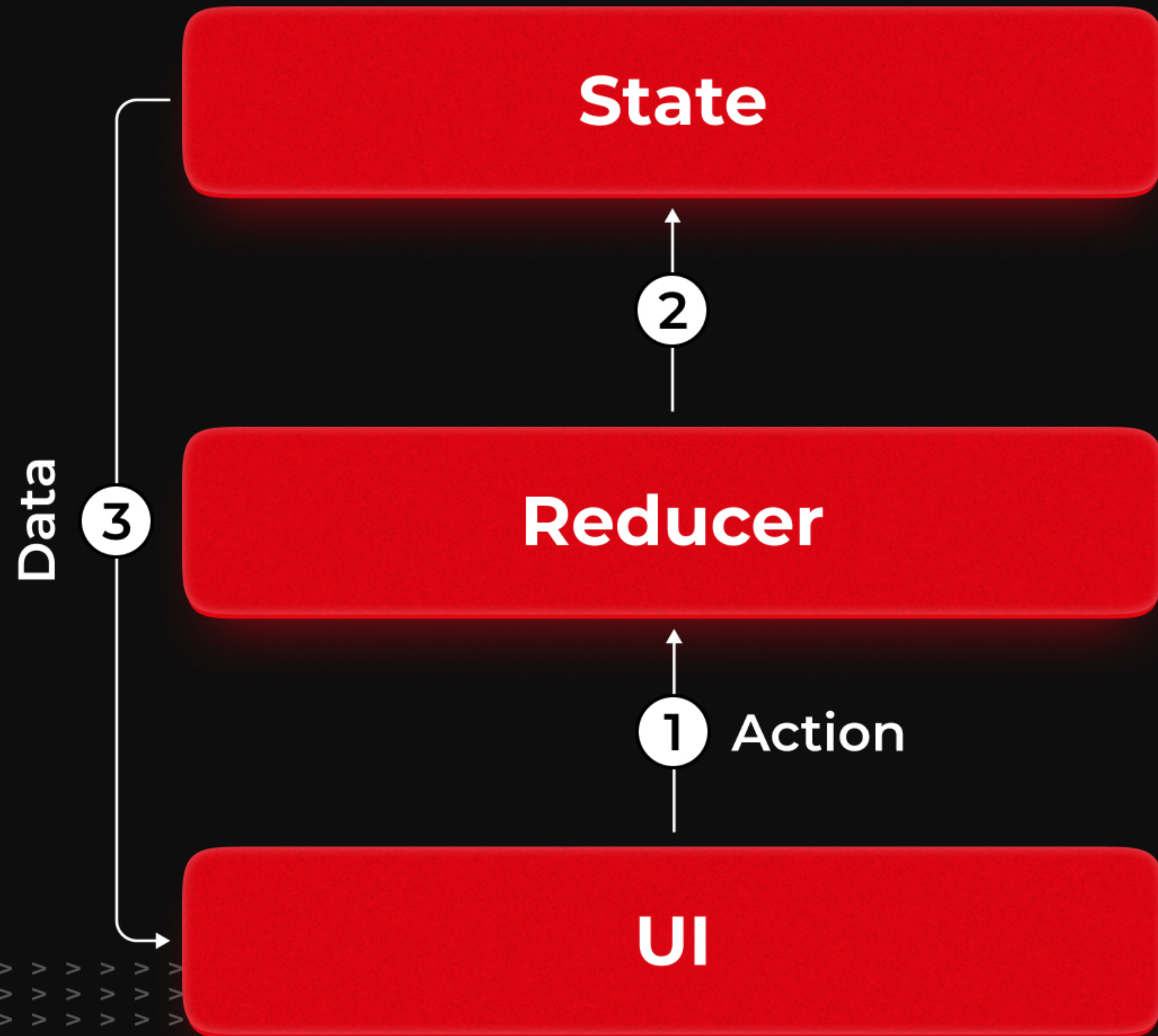


UNIDIRECTIONAL DATA FLOW



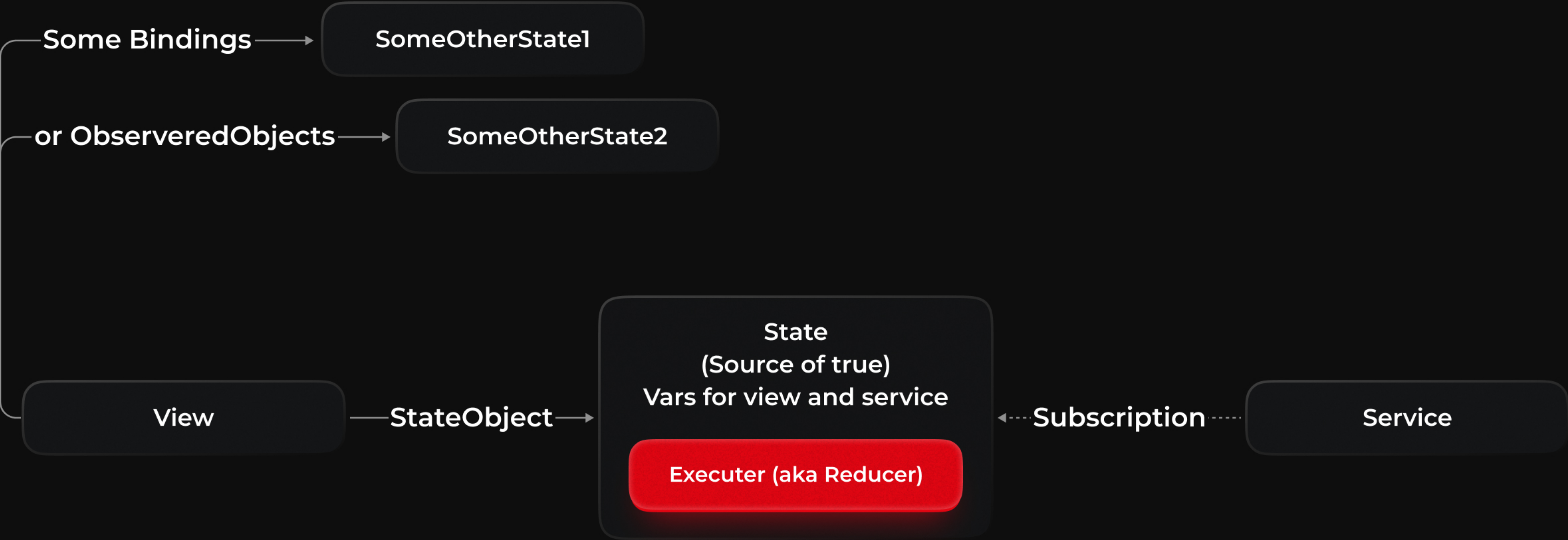
**КАК
В ОБЩЕМ
ВЫГЛЯДИТ
UDF**





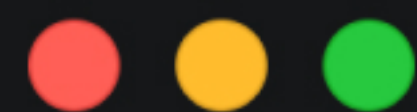
**КАК
В ОБЩЕМ
ВЫГЛЯДИТ
UDF**







```
final class HotelListState: ObservableObject {  
    @Published private(set) var requestStatus: RequestStatus = .initied  
  
    private var hotels: [Hotel] = []  
    private var pagingParams: PagingParams = .init(limit: 50, offset: 0)  
    private var sortingOrder: SortingOrder = .byPrice  
}
```

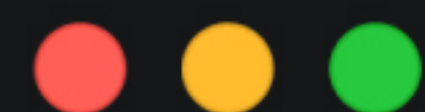



```
extension HotelListState: Executer {  
    enum Action {  
        case updateHotels(PagingParams)  
        case hotelsLoaded([Hotel])  
        case updateSortingOrder(SortingOrder)  
    }  
  
    func execute(action: Action) async {  
        switch action {  
        case let .updateHotels(params):  
            requestStatus = .pending(params)  
        case let .hotelsLoaded(newHotels):  
            hotels = newHotels  
        case let .updateSortingOrder(newOrder):  
            sortingOrder = newOrder  
        }  
    }  
}
```




```
struct HotelListView: View {  
    @StateObject private var state: HotelListState  
  
    @Binding private var filtersState: FiltersState  
    private var heading = "Yandex Travel hotel list!"  
  
    var body: some View {  
        Text(heading)  
        ForEach(state.hotels) { hotel in  
            Text(hotel.name)  
        }  
    }  
}
```

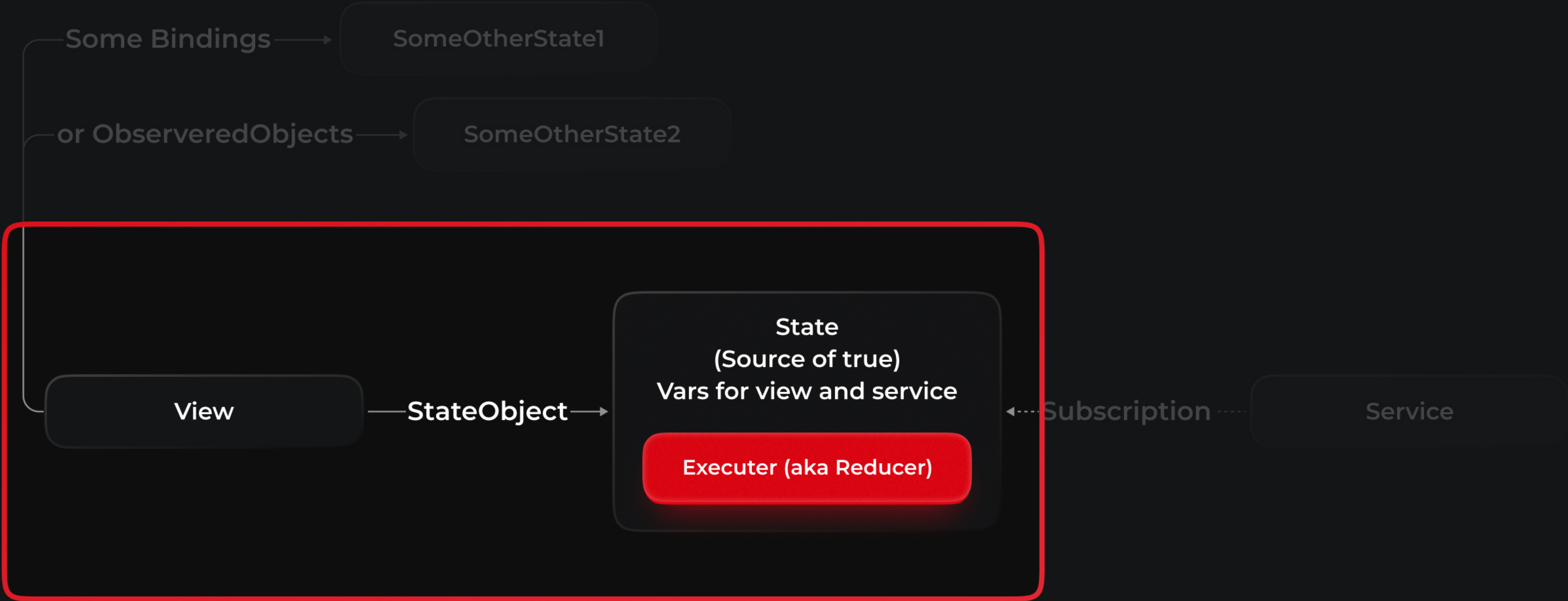




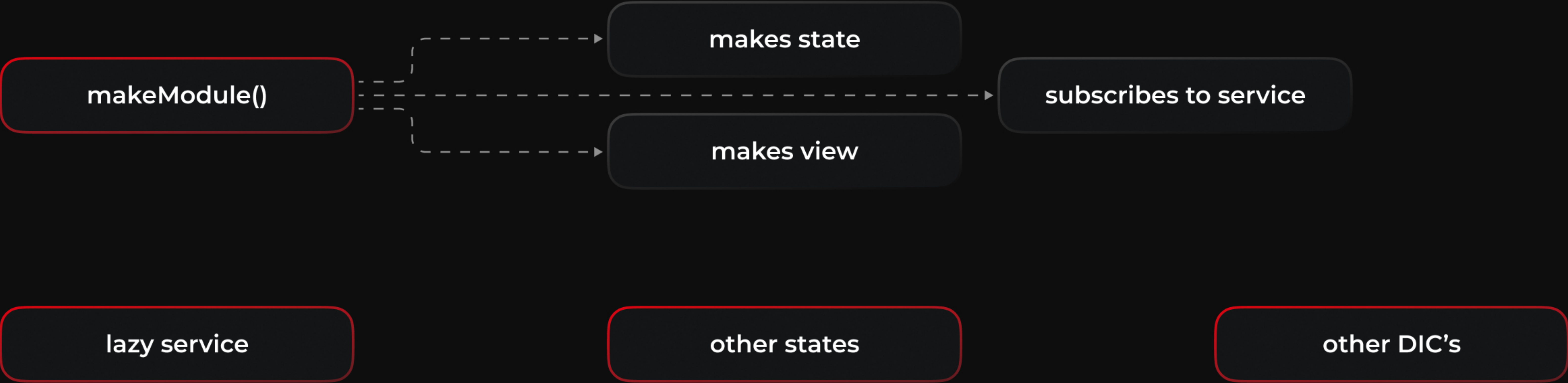
```
final class HotelListService {  
    private let networkService: NetworkService  
    private var fetchHotelsListTask: Task<Void, Error>?  
  
    func subscribe(to state: HotelListState) {  
        state.$requestStatus  
            .sink { [weak self] requestStatus in  
                switch requestStatus {  
                case let .pending(params):  
                    self?.fetchHotelsList(  
                        request: makeRequest(pagingParams: params),  
                        executor: state  
                    )  
                case .inited, .loaded:  
                    somethingElse()  
                }  
            }  
    }  
}
```




```
private func fetchHotelsList(request: Request, executer: HotelListState) {  
    fetchHotelsListTask = Task {  
        let result = await networkService.perform(request)  
        switch result {  
        case let .success(response):  
            await executer.execute(action: .hotelsLoaded(response))  
        case let .failure(error):  
            await handleError(error)  
        }  
    }  
}
```

Module



Some DIC



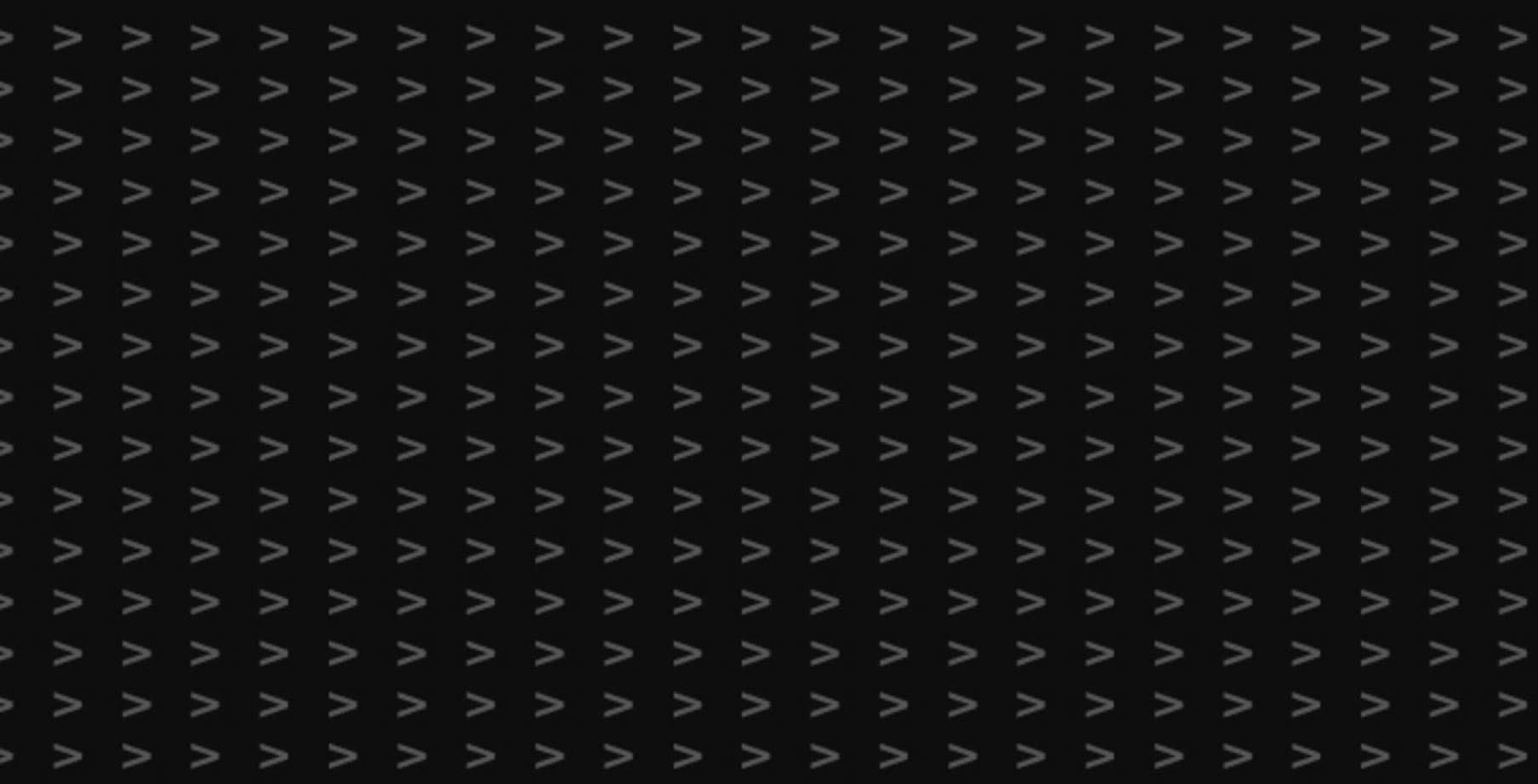
```
lazy var hotelListService = HotelListService()  
  
func makeHotelListState() {  
    HotelListState().then { state  
        hotelListService.subscribe(to: state)  
    }  
}
```


СПРАВНИМ С UIKIT

MVC V2.0
VIPER MVC
CLEANSWIFT RIBS
MVP YARCH
MVVM MVI
SILVER

VS

UDF



НАВИГАЦИЯ

№2





```
NavigationView {  
    NavigationLink(destination: DetailsView()) {  
        Text("Details")  
    }  
}
```




```
struct ContentView: View {
    @State private var isShowingDetailsView = false

    var body: some View {
        NavigationView {
            VStack {
                NavigationLink(destination: DetailsView(), isActive: $isShowingDetailsView) {
                    EmptyView()
                }
                Button("Details") {
                    self.isShowingDetailsView = true
                }
            }
        }
    }
}
```




```
struct ContentView: View {
    @State private var selection: String? = nil

    var body: some View {
        NavigationView {
            VStack {
                NavigationLink(destination: Text("View 1"), tag: "View1", selection: $selection) {
                    EmptyView()
                }
                NavigationLink(destination: Text("View 2"), tag: "View2", selection: $selection) {
                    EmptyView()
                }
                Button("Show View 1") {
                    self.selection = "View1"
                }
                Button("Show View 2") {
                    self.selection = "View2"
                }
            }
        }
    }
}
```


FlowStacks

Coordinator pattern in SwiftUI

FlowStacks allow you to manage complex SwiftUI navigation and presentation flows with a simple array. This makes it easy to hoist navigation state into a higher-level coordinator, allowing you to write isolated views that have zero knowledge of their context within the navigation flow of an app.

You might like this library if:

- ✔ You want to be able support deeplinks into deeply nested navigation routes in your app.
- ✔ You want to be able to easily reuse views within different navigation contexts.
- ✔ You want to easily go back to the root screen or a specific screen in the navigation stack.
- ✔ You want to use the coordinator pattern to keep navigation logic in a single place.
- ✔ You want to break an app's navigation into multiple reusable coordinators and compose them together.

The library works by translating the array of screens into a hierarchy of nested `NavigationLinks` and presentation calls, so:

- ⊘ It does not rely on UIKit at all.
- ⊘ It does not use `AnyView` to type-erase screens.
- ⊘ It does not try to recreate `NavigationView` from scratch.





```
enum Screen {  
    case home  
    case numberList  
    case numberDetail(Int)  
}
```




```
struct AppCoordinator: View {
    @State var routes: Routes<Screen> = [.root(.home)]

    var body: some View {
        Router($routes) { screen, _ in
            switch screen {
            case .home:
                HomeView(onGoTapped: showNumberList)
            case .numberList:
                NumberListView(onNumberSelected: showNumber, cancel: goBack)
            case .numberDetail(let number):
                NumberDetailView(number: number, cancel: goBackToRoot)
            }
        }
    }

    private func showNumberList() {
        routes.presentSheet(.numberList, embedInNavigationView: true)
    }

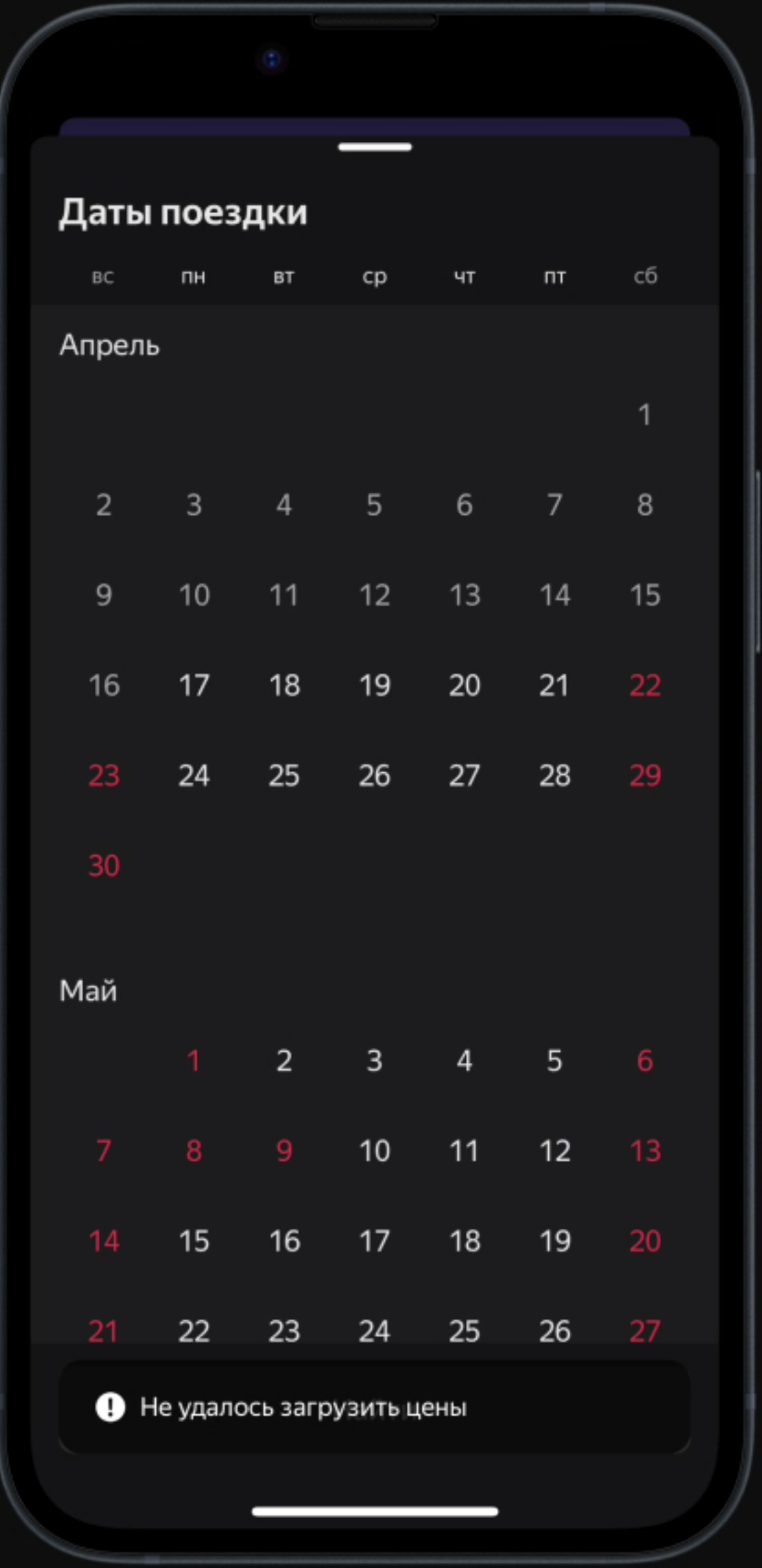
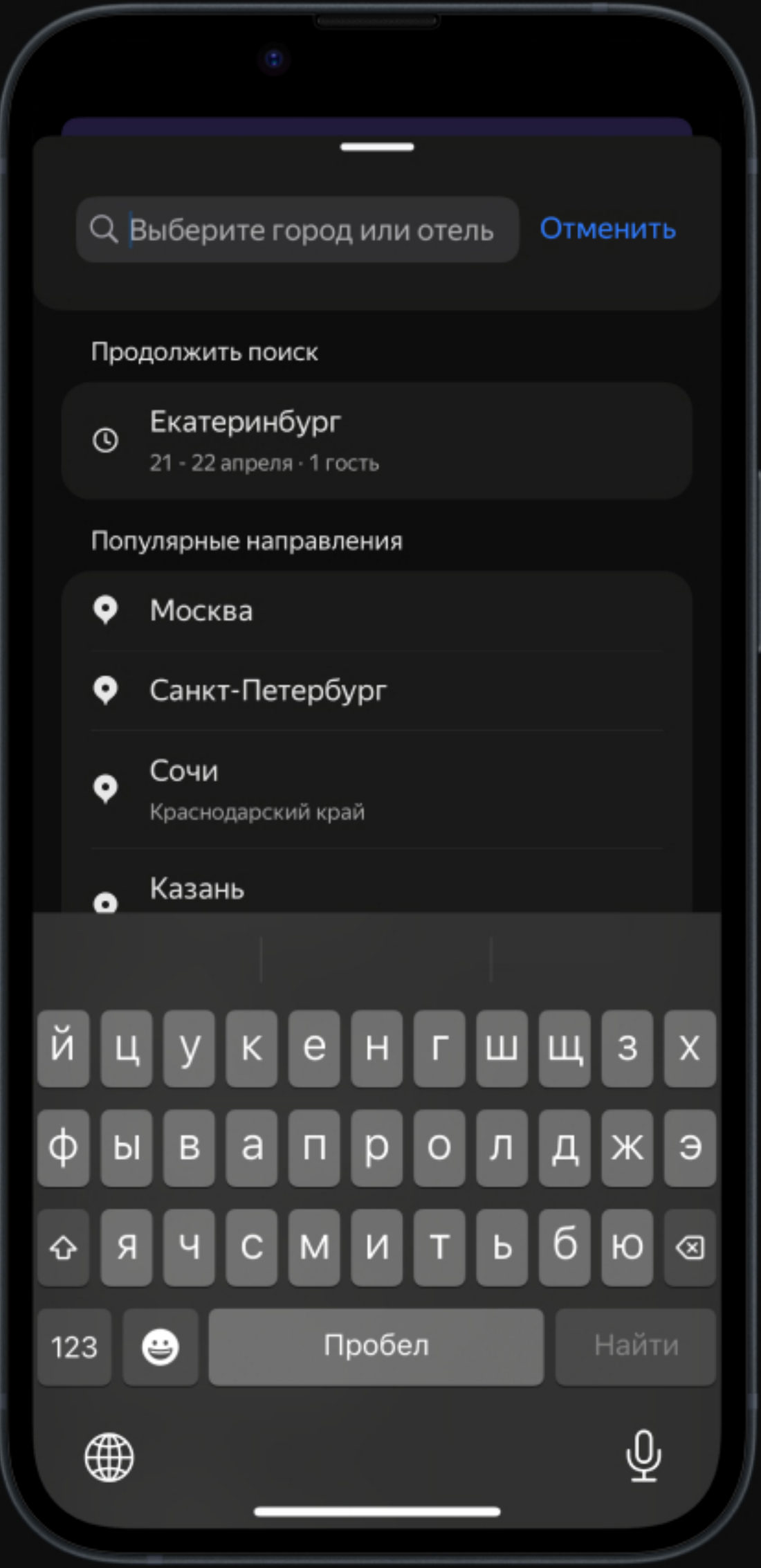
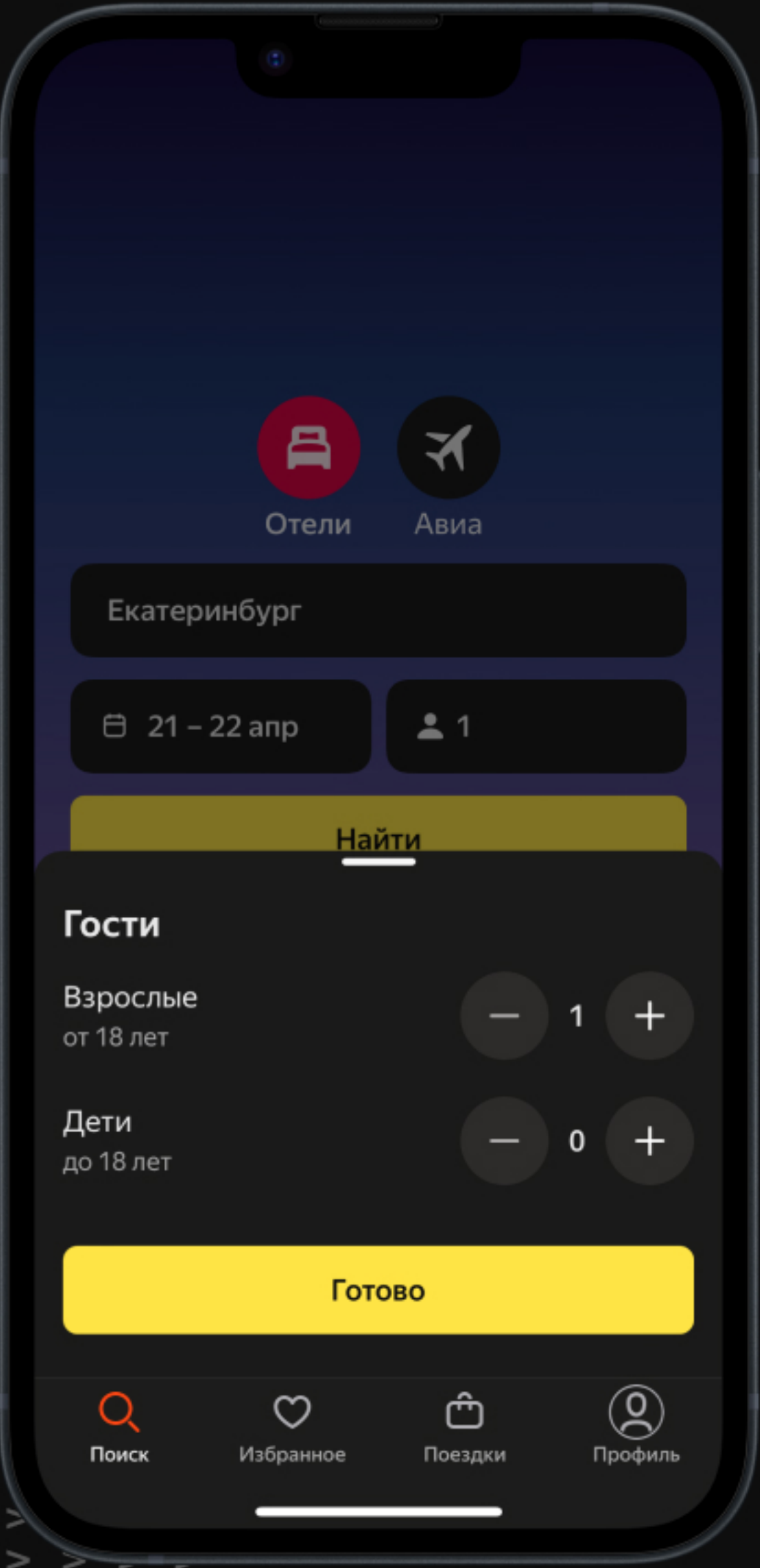
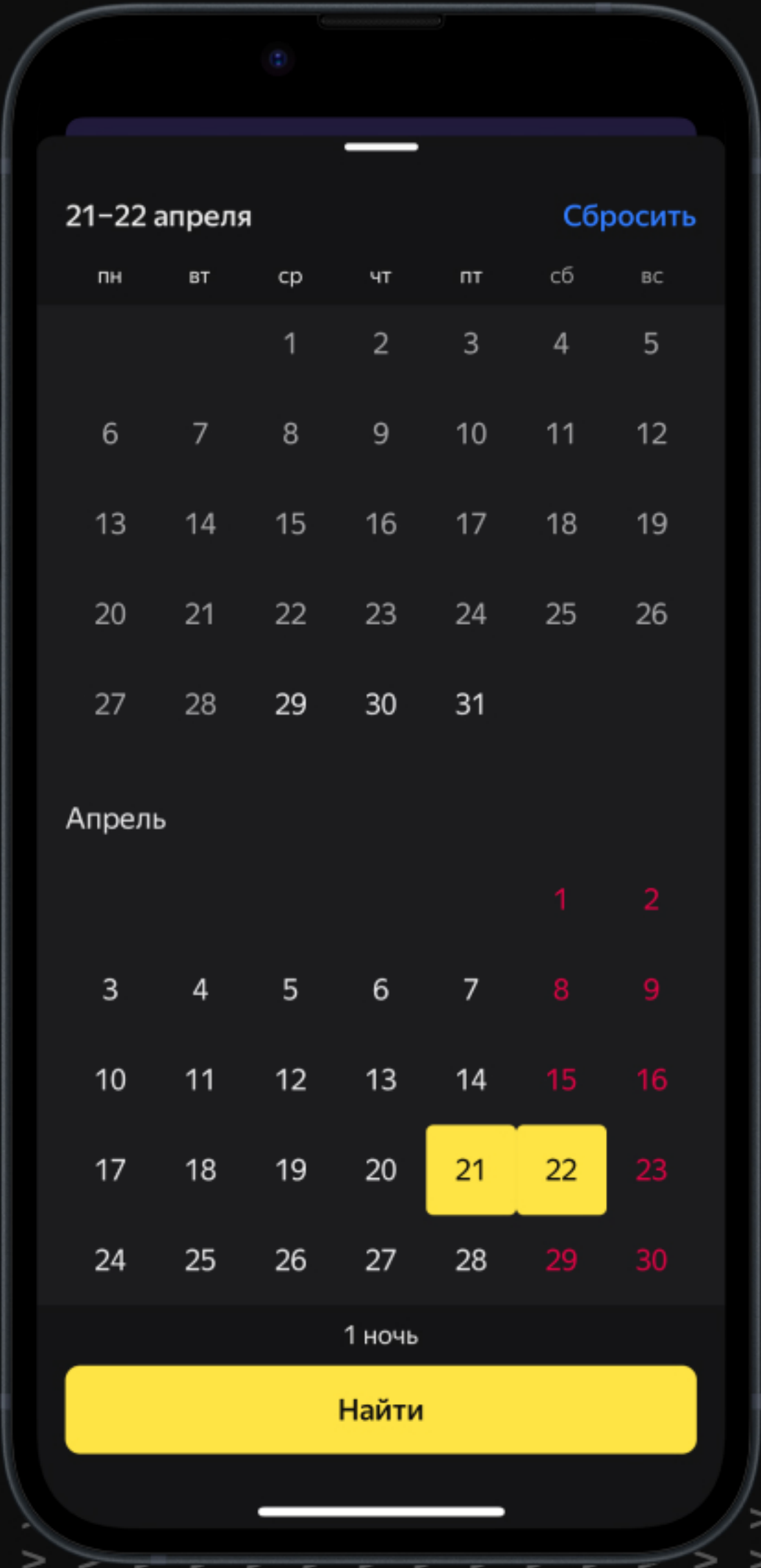
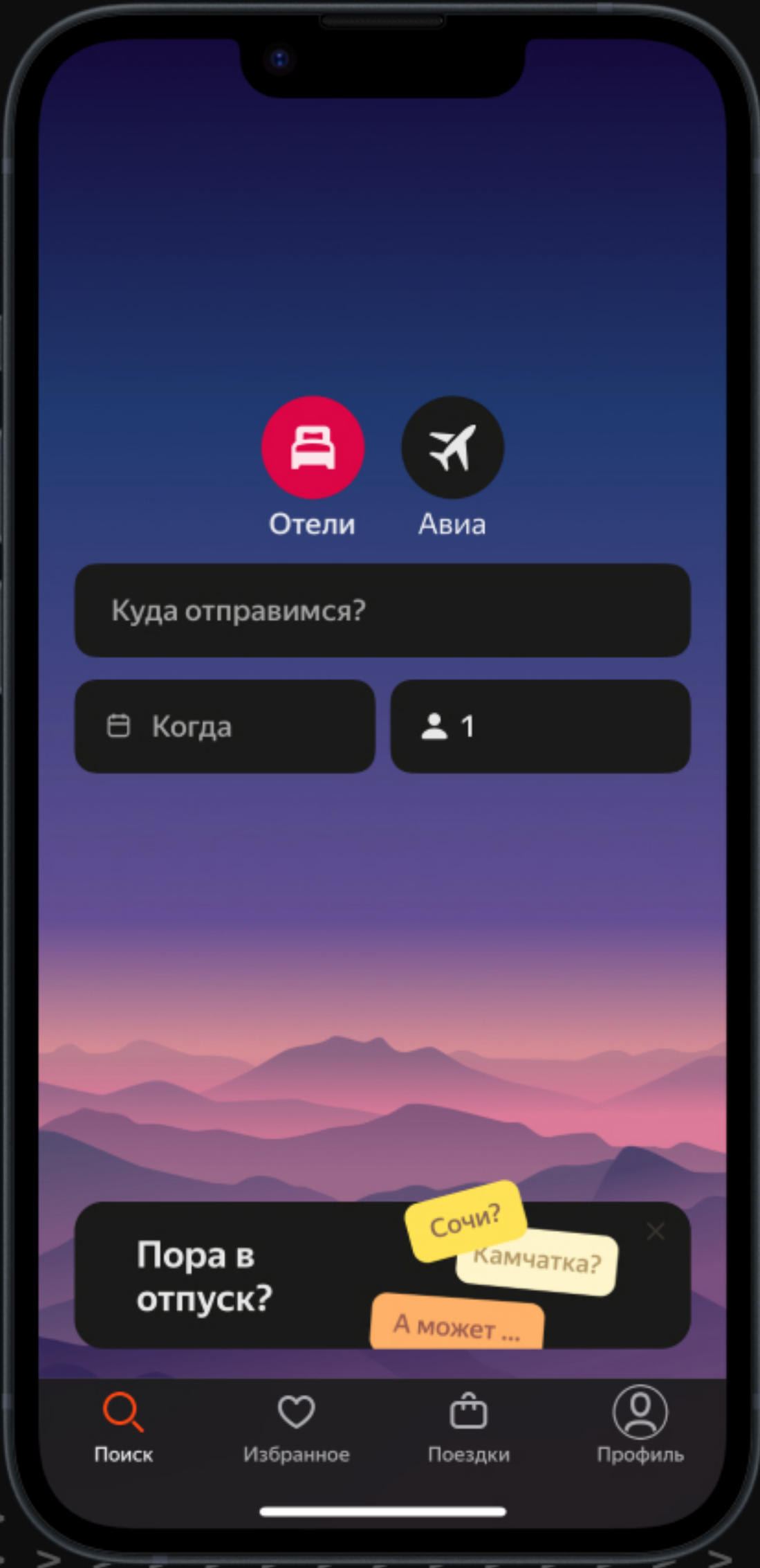
    private func showNumber(_ number: Int) {
        routes.push(.numberDetail(number))
    }

    private func goBack() {
        routes.goBack()
    }

    private func goBackToRoot() {
        routes.goBackToRoot()
    }
}
```


ЧТО у НАС?







```
struct MordaCoordinator: View {
  @ObservedObject var navigationState: MordaNavigationState
  @ObservedObject var snackbarState: SnackbarState

  var body: some View {
    rootView
      .showing($navigationState.routes, embedInNavigationView: true) { screen in
        routesBuilder(screen: screen)
      }
      .sheet(item: $navigationState.presentedSheet, ... )
      .bottomSheet(item: $navigationState.travellersCountSheet) { ... }
      .snackbar(style: mordaSnackbarState.style)
  }
}
```





```
extension MordaCoordinator {  
    @ViewBuilder func routesBuilder(screen: MordaNavigationState.Screen) -> some View {  
        switch screen {  
        case .hotelList:  
            mordaFactory.makeHotelListView()  
        case let .hotelInfo(hotel):  
            mordaFactory.makeHotelInfoCoordinator(hotel: hotel)  
        }  
    }  
}
```




```
struct MordaCoordinator: View {
  @ObservedObject var navigationState: MordaNavigationState
  @ObservedObject var snackbarState: SnackbarState

  var body: some View {
    rootView
      .showing($navigationState.routes, embedInNavigationView: true) { screen in
        routesBuilder(screen: screen)
      }
      .sheet(item: $navigationState.presentedSheet, ... )
      .bottomSheet(item: $navigationState.travellersCountSheet) { ... }
      .snackbar(style: $navigationStatr.mordaSnackbarState.style)
  }
}
```





```
public protocol NavigationState: ObservableObject {  
    associatedtype Screen where Screen: Routable // енамчик с экранами  
    var routes: Routes<Screen> { get set } // текущие пути  
}
```





```
public extension NavigationState {  
    @MainActor func push(_ screen: Screen) {  
        routes.push(screen)  
    }  
}
```





```
public extension NavigationState {  
    @MainActor func pop() -> Bool {  
        if !routes.isEmpty {  
            routes.pop()  
        }  
        return !routes.isEmpty  
    }  
}
```





```
final class MordaNavigationState: NavigationState {  
    @Published var routes: Routes<Screen> = []  
    @Published var presentedSheet: ModalSheet?  
    @Published var travellersCountSheet: TravellerCountSheet?  
  
    lazy var mordaSnackBarState = SnackBarState()  
}
```




```
extension MordaNavigationState {  
    enum Screen: Routable {  
        case hotelsSerp  
        case hotelInfo(Proto.Hotel)  
    }  
  
    enum ModalSheet {  
        case hotelsDestination()  
        case hotelsCalendar()  
    }  
  
    enum TravellerCountSheet {  
        case hotels()  
    }  
}
```




```
extension MordaNavigationState: Executer {  
    enum Action {  
        case present(ModalSheet)  
        case dismissSheet  
        case navigateToHotelList  
    }  
  
    @MainActor func execute(action: Action) async {  
        switch action {  
        case .dismissSheet:  
            presentedSheet = nil  
        case let .present(sheet):  
            presentedSheet = sheet  
        case .navigateToHotelSerp:  
            hotelNavigationState?.goBackToRoot()  
            routes.push(.hotelsSerp)  
        }  
    }  
}
```


ПРИМЕР ИСПОЛЬЗОВАНИЯ НАВСТЕЙТА ИЗВНЕ



```
navigationState.push(.hotelInfo)
```

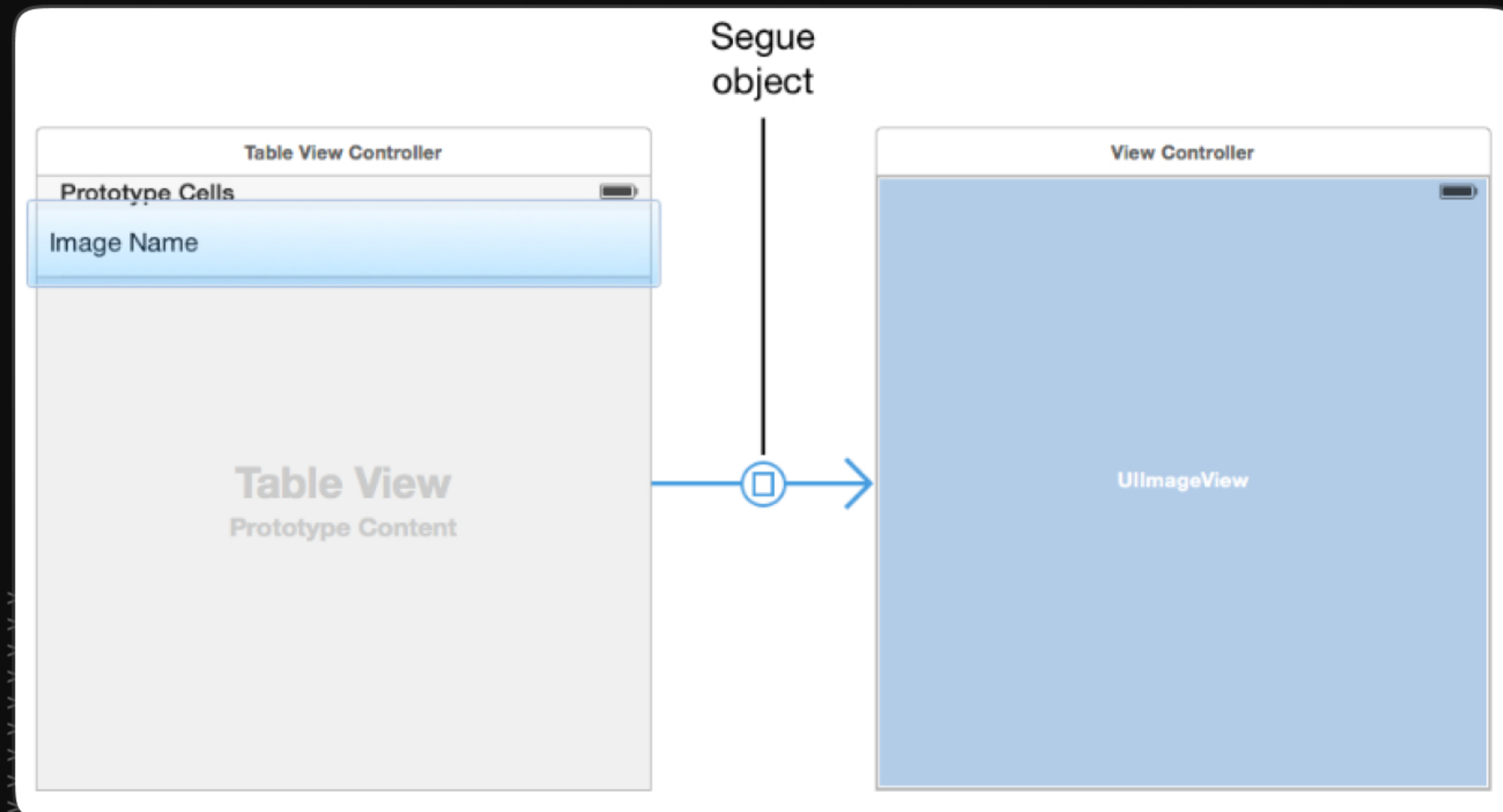


```
navigationState.execute(action: .dismissSheet)
```



СПРАВНИМ С UIKIT

Спасибо, что не



ЗАВИСИМОСТИ

№3

Проект на SwiftUI



а зависимости на UIKit!



О боже, Крис, они же
ещё джуны





```
public protocol UIViewRepresentable : View where Self.Body == Never {  
    associatedtype UIViewType : UIView  
  
    @MainActor func makeUIView(context: Self.Context) -> Self.UIViewType  
  
    @MainActor func updateUIView(_ uiView: Self.UIViewType, context: Self.Context)  
}
```





```
public protocol UIViewControllerRepresentable : View where Self.Body == Never {  
    associatedtype UIViewControllerType : UIViewController  
  
    @MainActor func makeUIViewController(context: Self.Context) -> Self.UIViewControllerType  
  
    @MainActor func updateUIViewController(_ viewController: Self.UIViewControllerType, context:  
Self.Context)
```

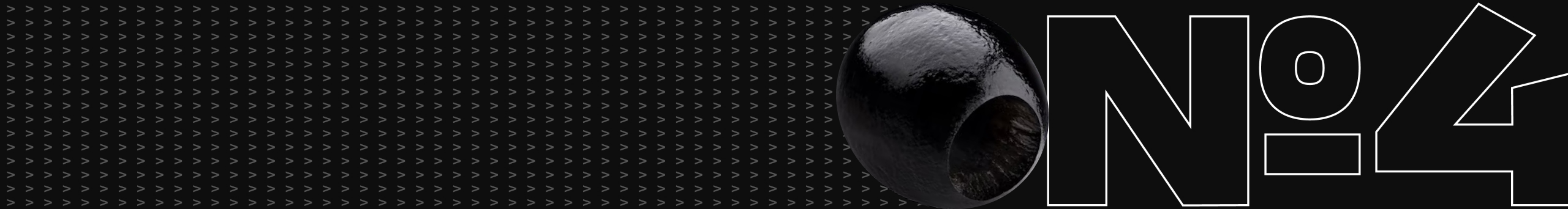



Пока билдится превьюха [no nda]

12 members, 2 online



ОСИ



[iOS] Инпуты цены не отображаются на ios 14

[iOS] Не работают шторки на экране деталей отеля и офферов (iOS 14)

[IOS] Отели: Не подгружается информация о ценах на странице деталей отеля для iOS 14.2

Кнопка перехода к чекауту на iOS 14 становится некликабельной при скролле

[iOS] Дёрганный скролл в pull-to-refresh (iOS 14)

[iOS] На ios 14 видна адресная строка

[iOS] Отельная галерея: уезжают точки внизу слайдера (iOS 14)

[iOS] WebView crash (iOS 14, push?)

[research] [iOS] Стабильно не работает авиавыдача на iOS 14.8.1

[iOS] Починить автоинпут на iOS 14.6



[iOS][13] Саджест уехал на середину страницы

[iOS][13.4] не работает саджест

[iOS] Таббар закрывает нижнюю часть scrollview (не даёт доскроллить до конца)

[iOS] Авиа: Выдача - Не происходит перепоиск при добавлении даты Обратно

[iOS] Фриз на открытии календаря

[iOS] Баг на iOS 13: календарь сжался

[iOS] iOS 13: сниппет выходит за правую границу экрана

[iOS][13] Загадочное исчезновение навбара

[iOS] Баг на iOS 13: рандомные города на саджесте

[iOS][13] Поехала вёрстка экрана поездки без залога

[iOS] Краш на выдаче Авиа на iOS 13

[iOS] iOS 13. Мелькает морда на выдаче

[iOS] Баг на iOS 13: поехала сортировка

[iOS] Beta al dente: Проблемы с запуском на iOS 13 (девайс-специфичные)

[iOS] Мои поездки. При переходе из профиля кидается ошибка "что-то пошло не так"



[iOS][13] Саджест уехал на середину
страницы

[iOS][13.4] не работает саджест

[iOS] Таббар закрывает нижнюю часть scrollview (не даёт доскроллить до конца)

**[iOS] Авиа: Выдача - Не происходит
перепойск при добавлении даты Обратно**

[iOS] Фриз на открытии календаря

[iOS] Баг на iOS 13: календар

Численное исчезновение навбара

[iOS] Баг на iOS 13: серый квадрат

[iOS] Баг на iOS 13: рандомные города на саджете

без залоги́на

Честе Авија на iOS 13

[iOS] Баг на iOS 13: поехала сортировка

[iOS] iOS 13. Мелькает морда на выдаче

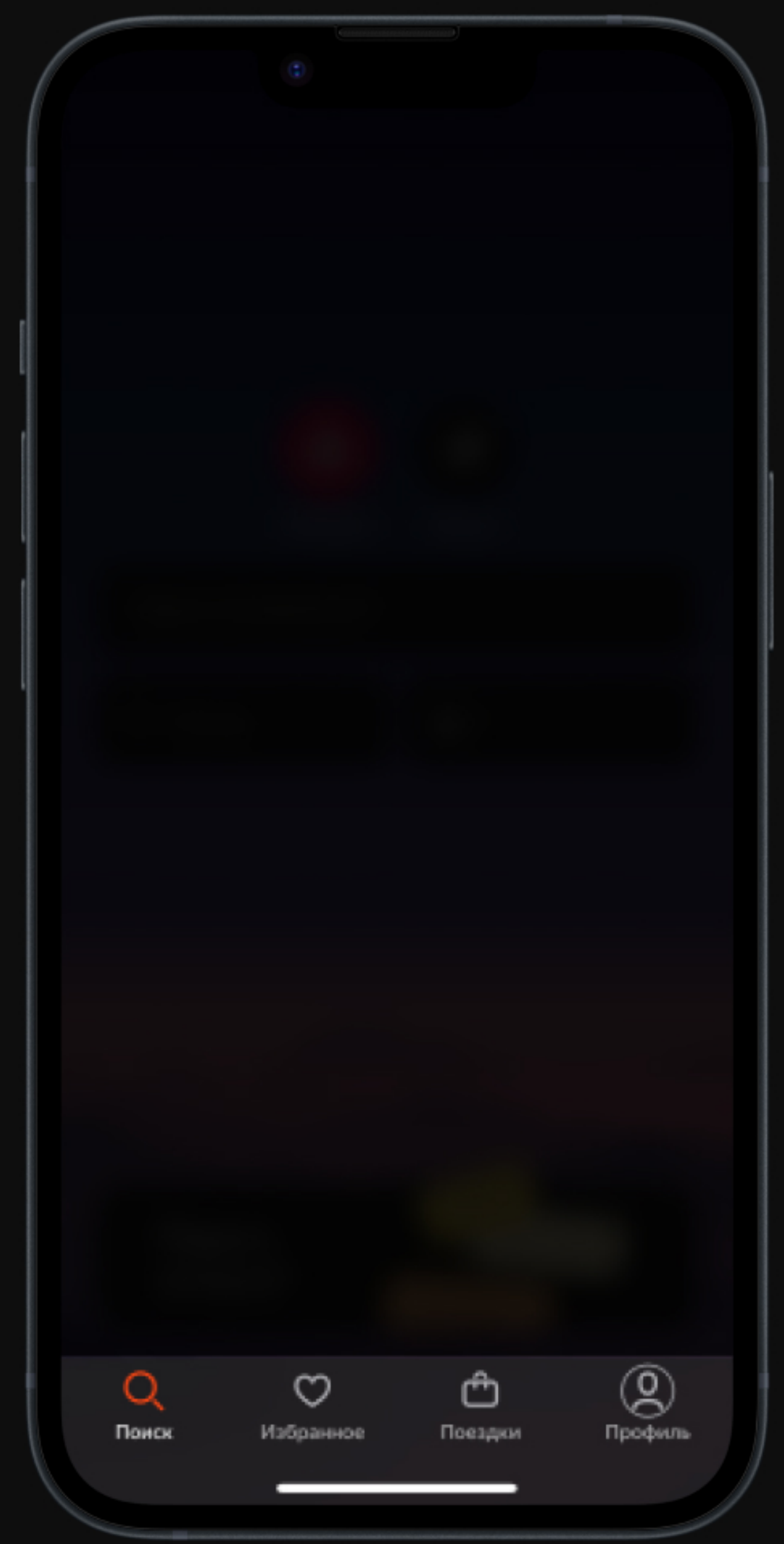
[iOS] Beta al dente: Проблемы с запуском на iOS 13 (девайс-специфичные)

[iOS] Мои поездки. При переходе из профиля кидается ошибка "что-то пошло не так"

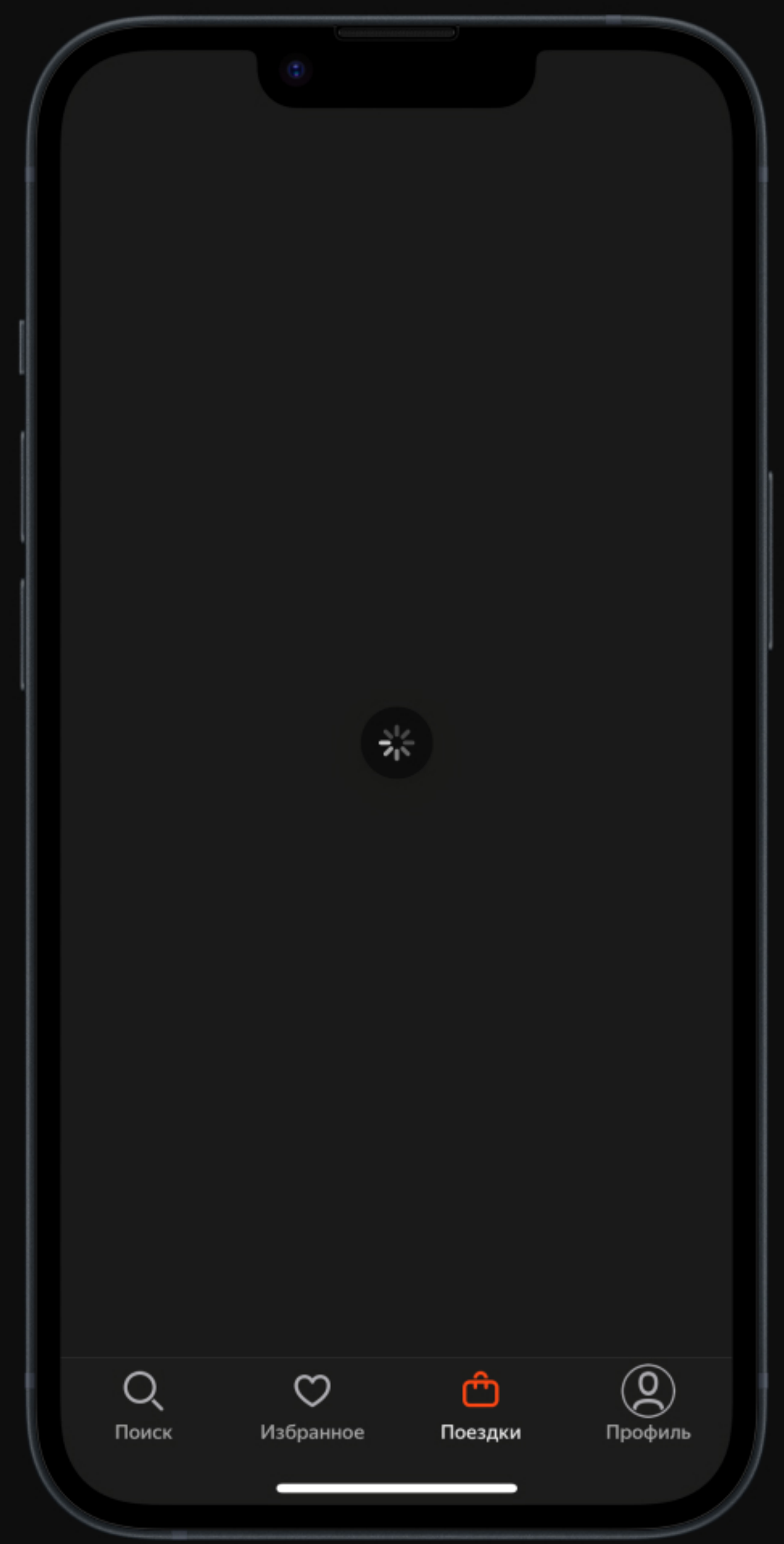
[iOS] Поднятие таргета до iOS 14

СЛОЖНЫЕ ОБЪЕКТЫ

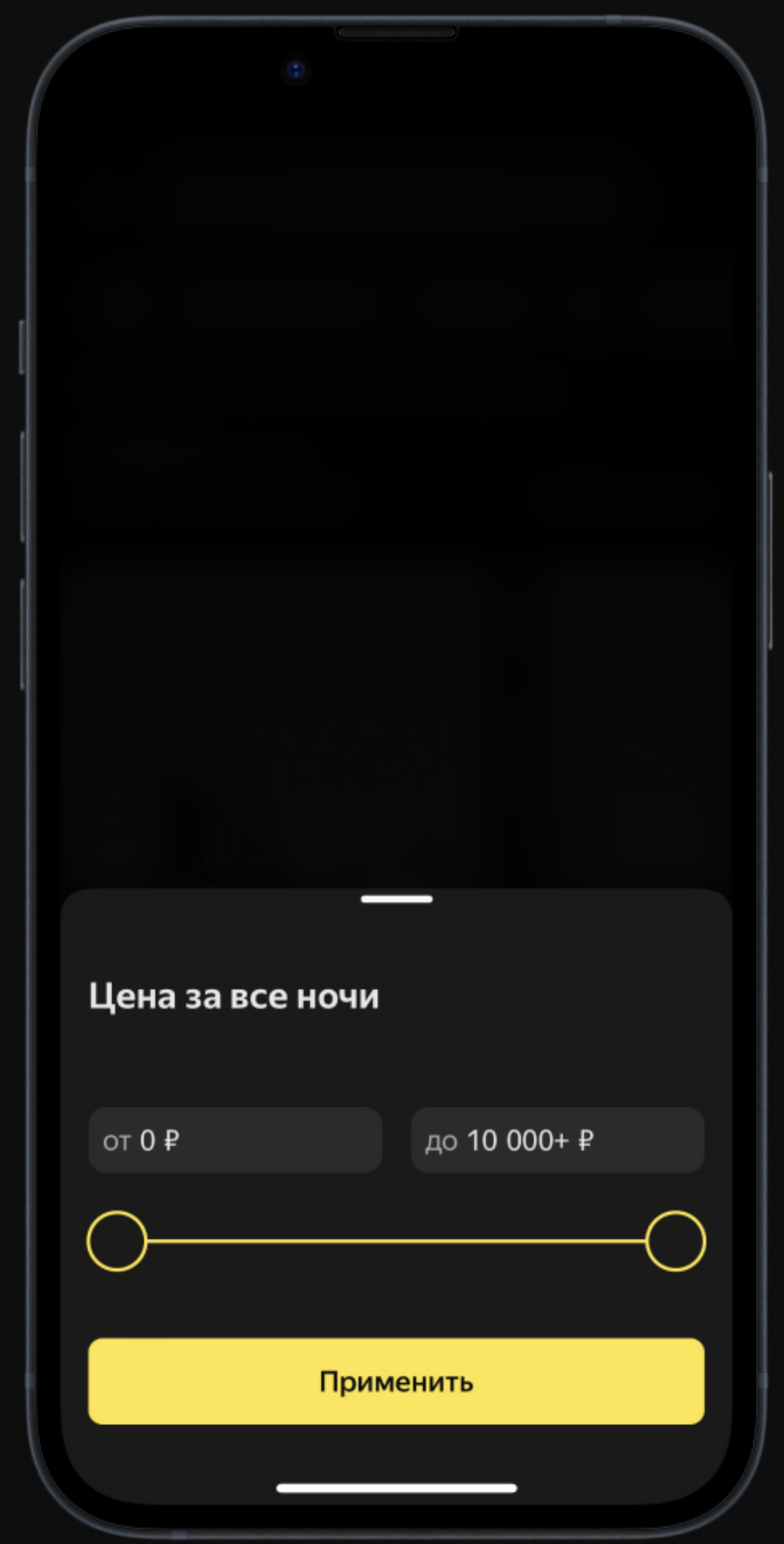
№5



Tabbar

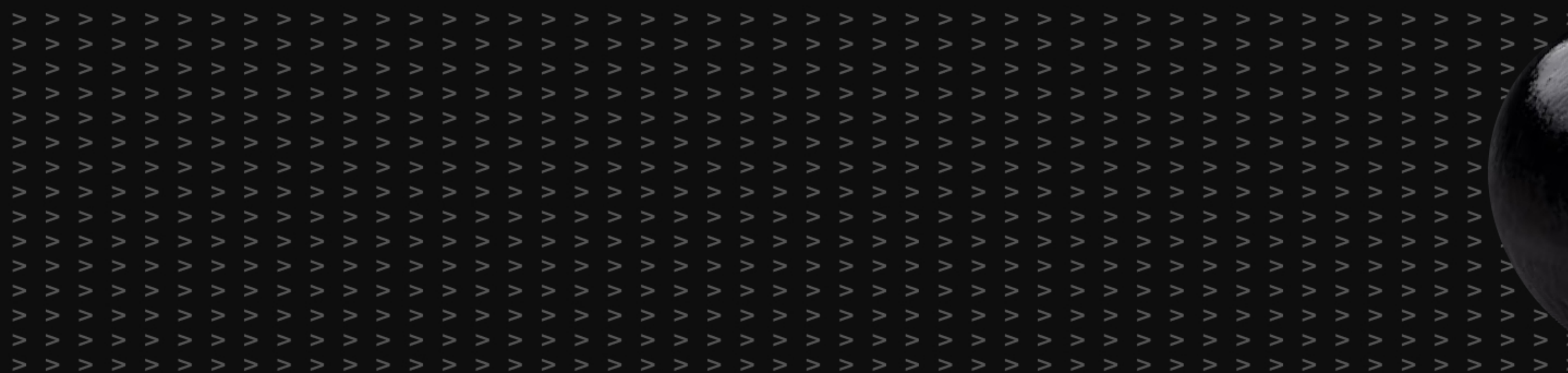


WebView



PartialSh*t

ОНБОРДИНГ



№6



**ЭТОМУ
ДЖУНУ
ВСЕГО 22
ГОДА**

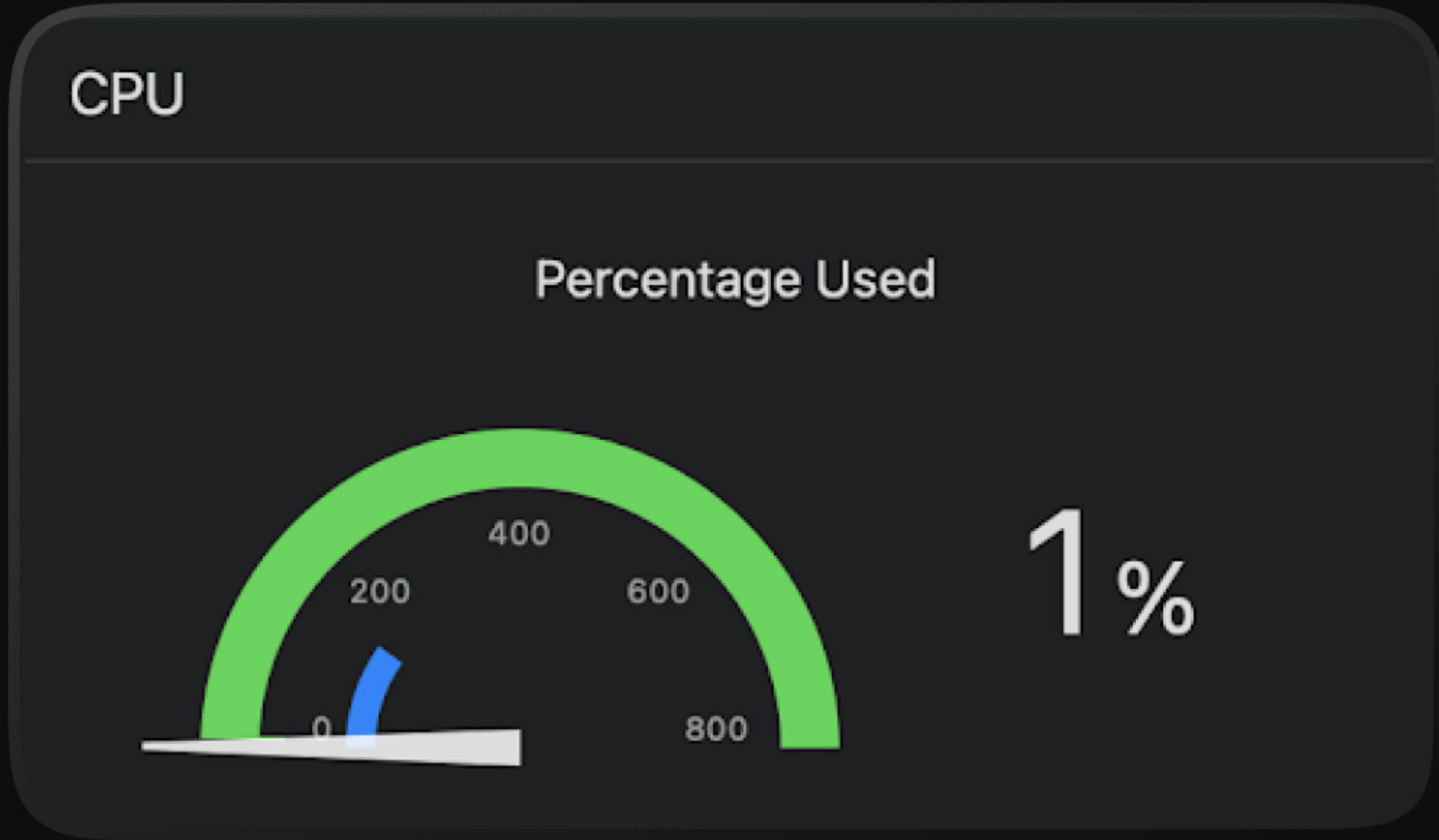
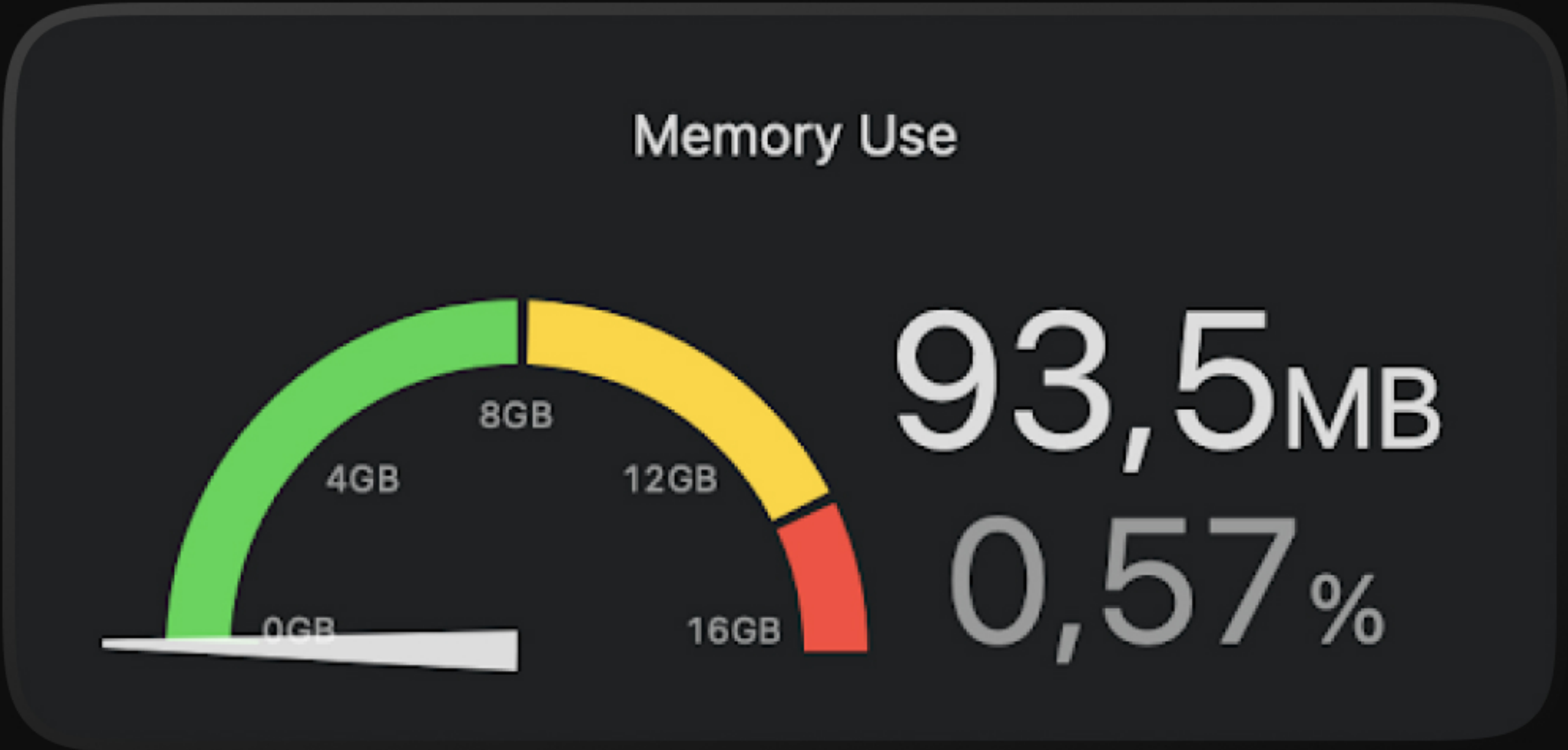
ЧТО В ИТОГЕ?



ЧТО В ИТОГЕ?



● Архитектура удалась!



ЧТО В ИТОГЕ?



- Архитектура удалась!
- Навигация нас устроила

ЧТО В ИТОГЕ?



- Архитектура удалась!
- Навигация нас устроила
- Починка зависимостей в процессе

ЧТО В ИТОГЕ?



- Архитектура удалась!
- Навигация нас устроила
- Починка зависимостей в процессе
- Таргет поднят до iOS 14

ЧТО В ИТОГЕ?



- Архитектура удалась!
- Навигация нас устроила
- Починка зависимостей в процессе
- Таргет поднят до iOS 14
- Офигенная команда, готовая к вызовам

ЧТО В ИТОГЕ?



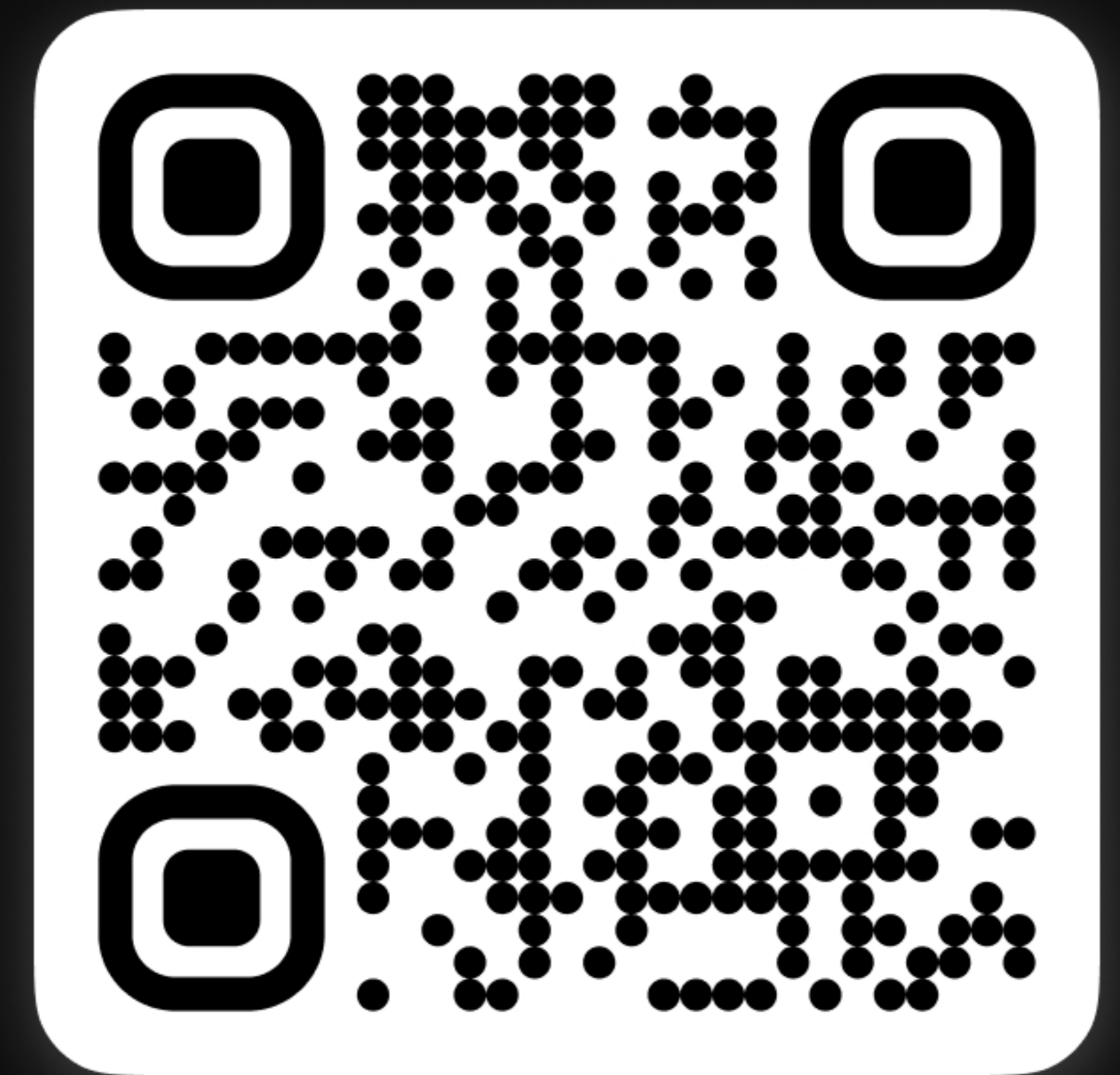
- Архитектура удалась!
- Навигация нас устроила
- Починка зависимостей в процессе
- Таргет поднят до iOS 14
- Офигенная команда, готовая к вызовам
- Ну и написали доку 🎉



ЖАЛЕЕМ ЛИ МЫ?



ЕСЛИ ВЫ СОБИРАЕТЕСЬ ДЕЛАТЬ ПРИЛОЖЕНИЕ НА
НОВЫХ ТЕХНОЛОГИЯХ: IOS, **ANDROID**, **FLUTTER** —
ПРИХОДИТЕ МЫ УМЕЕМ



Выбирайте те **ТЕХНОЛОГИИ**,
которые вам по **ДУШЕ**, а которые
не по душе — **не выбирайте**

-Антон Назаров



СТАТЬЯ ПРО НАВИГАЦИЮ В **АНДРОИД** В КЕЙСЕ ЯНДЕКС ПУТЕШЕСТВИЙ



СПАСИБО ЗА ВНИМАНИЕ!

Кейс ↘



Полина Скалкина
iOS разработчик
📍 @bypolinafox