



# СТАЖИРОВКА

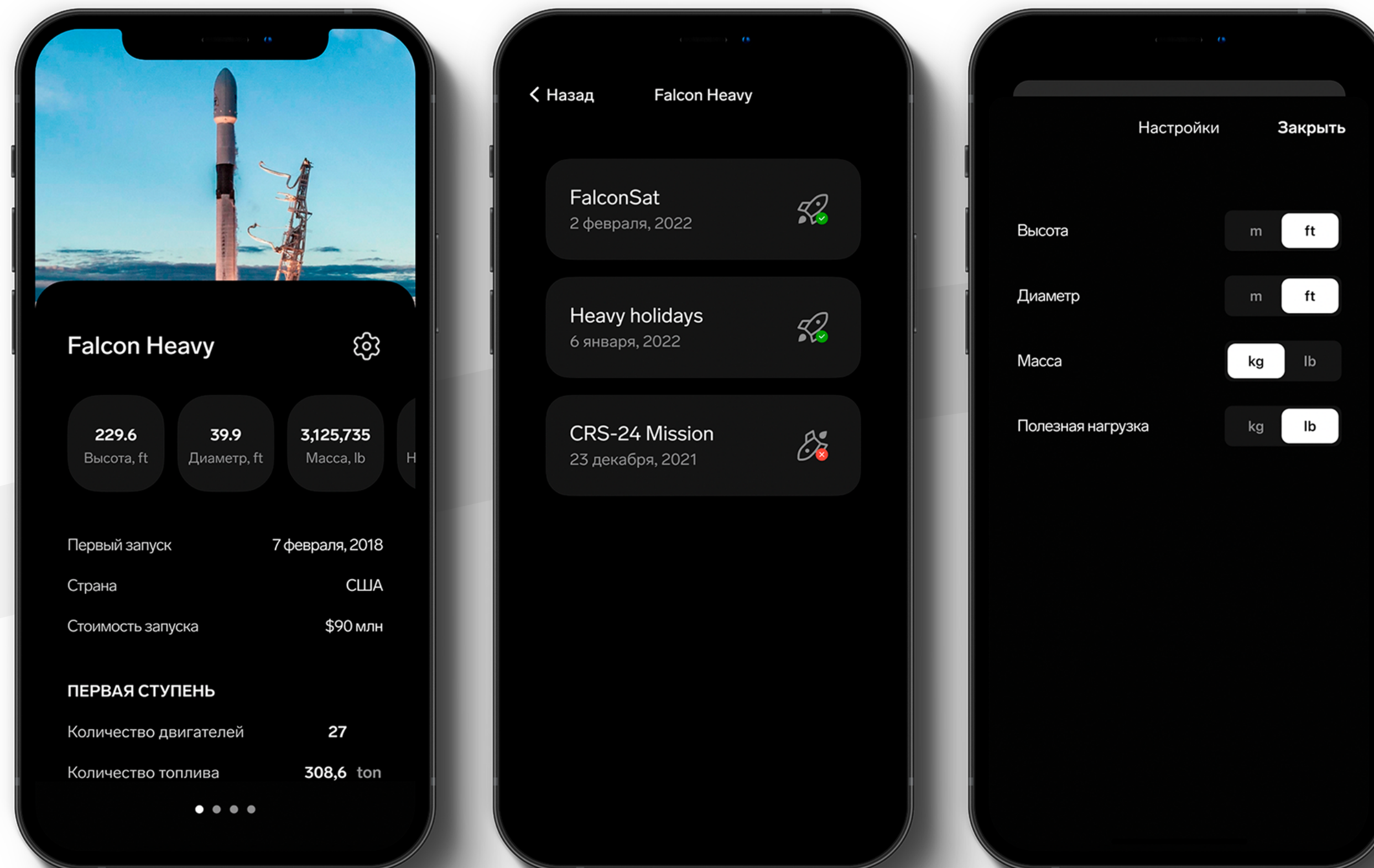
iOS-разработчика

Контур

2022

# Тестовое задание

Чтобы попасть на стажировку в Контур, попробуем сделать мобильное приложение. Приложение должно показывать информацию о космических ракетах SpaceX и список их запусков.



# Три варианта реализации

Есть 3 уровня сложности для задания, каждый следующий уровень дополняет предыдущий.  
Уверенный уровень – набор фич, при котором задание будет считаться выполненным. Выполнение фич для уровней Продвинутой или Максимальной даст дополнительные плюсы.

	Уверенный	Продвинутый	Максимальный
Экран 1 Космические ракеты SpaceX	Сделайте все фичи на слайде Экрана 1  Не делать горизонтальный блок информации с параметрами высоты, диаметра, массы и полезной нагрузки	Сделайте все фичи на слайде Экрана 1  Дополнительно реализовать горизонтальный блок информации с параметрами высоты, диаметра, массы и полезной нагрузки	Сделайте все фичи на слайде Экрана 1  Дополнительно реализовать иконку настроек и навигацию на Экран 3 Настройки  Дополнительно возможность переключения единиц измерения параметров высоты, диаметра, массы и полезной нагрузки.
Экран 2 Список запусков космической ракеты	Сделайте все фичи на слайде Экрана 2	Сделайте все фичи на слайде Экрана 2	Сделайте все фичи на слайде Экрана 2
Экран 3 Настройки	Можно не делать иконку настроек и всё что связано с настройками	Можно не делать иконку настроек и всё что связано с настройками	Сделайте все фичи на слайде Экран 3



# Экран 1

## Космические ракеты SpaceX

Это стартовый экран приложения.

Так как космических ракет несколько, в качестве навигации между ними предлагаем использовать элемент Page Control.

В верхней части экрана располагается случайное изображение космической ракеты и ее наименование.

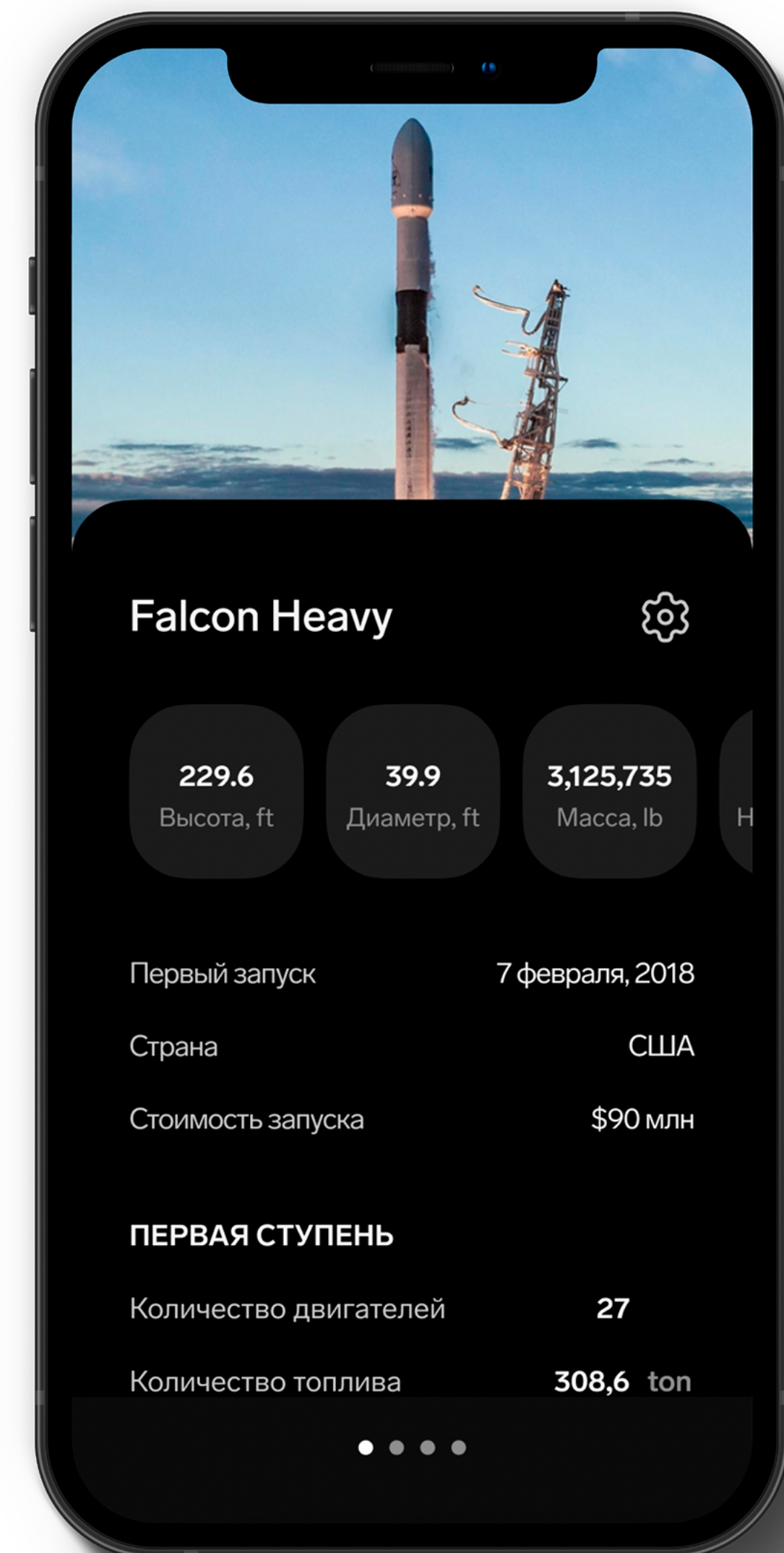
Далее идет блок горизонтальной информации с указанием названия параметра, значения и единицы измерения:

- высота ракеты
- диаметр
- масса
- полезная нагрузка для id «leo» (низкой околоземной орбиты)

Затем необходимо отобразить вертикально следующую информацию:

- дата первого запуска
- страна
- стоимость запуска
- первая ступень - количество двигателей
- первая ступень - количество топлива в тоннах
- первая ступень - время сгорания в секундах
- вторая ступень - количество двигателей
- вторая ступень - количество топлива в тоннах
- вторая ступень - время сгорания в секундах

В нижней части экрана показывается кнопка «Посмотреть запуски», при нажатии на данную кнопку происходит переход на экран 2.



# Экран 2

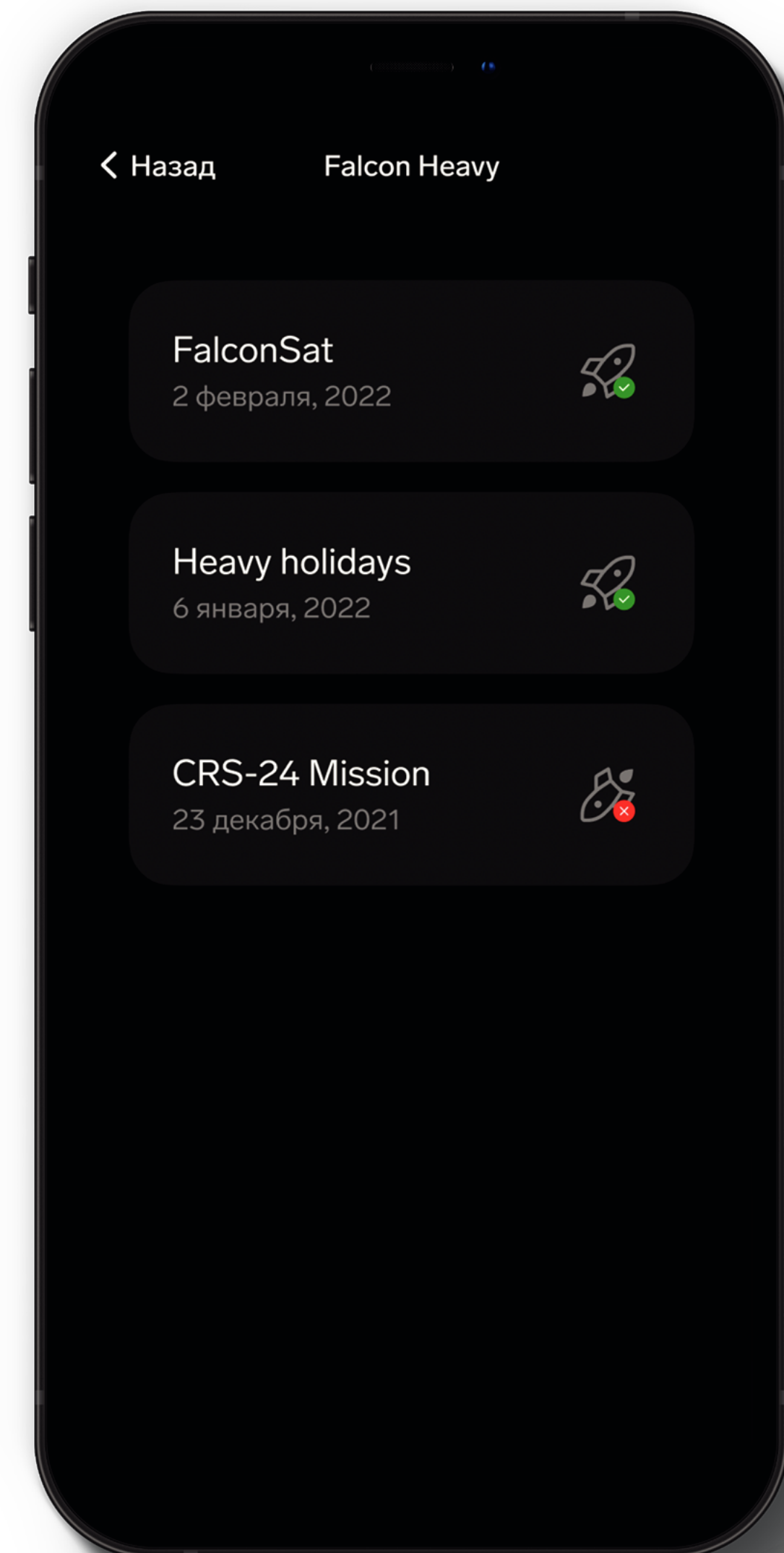
## Список запусков космической ракеты

Таблица, в которой отображается название космической ракеты и список ее состоявшихся запусков.

По каждому запуску ракеты показываем:

- название запуска
- дата
- иконка успешного/неуспешного запуска

Информацию о запусках конкретной ракеты можно фильтровать либо на стороне приложения, либо на сервере, воспользовавшись фильтрацией в запросе ([документация к запросам](#))



# Экран 3

## Настройки

Предусмотреть напротив названия ракеты на экране 1 кнопку, при нажатии на которую презентуем модально Настройки.

Данный экран предполагает выбор единиц измерения для следующих параметров ракет:

- высота ракеты
- диаметр
- масса
- полезная нагрузка

После закрытия экрана Настроек выбранные единицы измерения должны примениться к соответствующим параметрам на главном экране со списком ракет.

\*Дополнительно - реализовать сохранение пользовательских настроек и при последующих запусках приложения их применять





# Вспомогательная информация

## API

Для получения информации используйте:

Космические ракеты - <https://api.spacexdata.com/v4/rockets> | Запуски - <https://api.spacexdata.com/v4/launches>

## Дизайн

Дизайн [по ссылке](#) является рекомендованным, любые ваши улучшения приветствуются.

Дополнительно можете воспользоваться [интерактивным режимом дизайна](#), основной экран прокручивается, кнопки настроек и «Посмотреть запуски» являются активными.

## MacOS и Xcode

Для выполнения задания необходима IDE Xcode. Ее можно установить из AppStore на операционной системе macOS.

На просторах интернета есть информация о том, как стать пользователем данной операционной системы, если вы не являетесь обладателем техники Apple.

## Технологии

Исходный код должен быть написан на Swift, для разработки интерфейса можно использовать UIKit или SwiftUI как с помощью Interface Builder, так и в коде.

Для работы с сетевыми запросами можете использовать нативный API, либо сторонние библиотеки, например Moya или Alamofire.

## Результат

В качестве результата тестового задания ожидается проект приложения для Xcode, запускаемый под операционную систему iOS 13 и старше.

Ключевое требование — работает, не падает.

В задании проверяются необходимы знания по трем основным направлениям для iOS-разработчика:

- Знания языка разработки и окружения (среда разработки, система контроля версий и прочее)
- Знание теоретических основ разработки (паттерны, архитектурные подходы)
- Знание фреймворков и популярных библиотек

Информации по программированию огромное количество, чтобы не потеряться в этом потоке, можете воспользоваться нашими [рекомендациями](#). Удачи!