

# Тестовое задание

## 1. Общая информация

- Прототип дизайна находится по гипер-ссылке в [Figma](#)
- Названия экранов в техническом задании соответствуют названию экранов в Figma
- Данные для экранов, полученные из запросов на сервер и данные, продемонстрированные в Figma могут отличаться
- Отрисовка элементов из макета в Figma должна быть идентична
- Все шрифты, цвета и иконки можно брать из Figma
- Для проверки задания нужно прислать видео работающего приложения, всех его экранов, а также отправить ссылку на GIT
- На выполнение ТЗ дается 10 дней с момента получения этого файла
- В гите не указывайте название компании, чтобы другие участники не смогли скопировать вашу работу
- Желаем вам успехов!

## 2. Описание экранов

### 2.1. Экран «Отель»

При запуске приложения начальным экраном является экран «Отель». Отель всегда один и тот же, так как это тестовое задание. Получить данные об отеле можно при запросе на сервер по этому [API](#).



Список элементов:

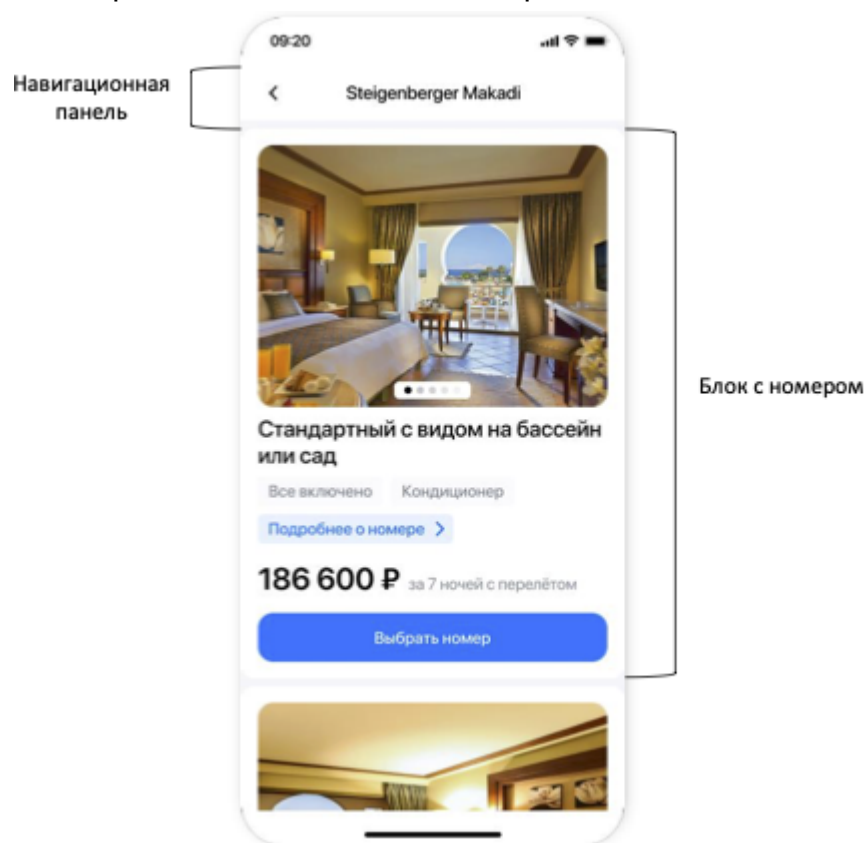
- **Карусель из фотографий** – это карусель (слайдер), который при свайпе вправо/влево перелистывает фотографии.

- **Основные данные про отель** – это блок с основными данными про отель. Адрес – это кнопка, но при нажатии на нее не происходит ничего (она некликабельна).
- **Подробные данные об отеле** – это блок с подробными данными об отеле. Кнопка «Удобства», «Что включено», «Что не включено» не имеют никаких действий, то есть они являются некликабельными, их нужно просто сверстать по макету.
- **Кнопка выбора номера** – это кнопка, при нажатии на которую осуществляется переход на экран «Номер».

## 2.2. Экран «Номер»

Этот экран открывается при нажатии на кнопку «к выбору номера» из экрана «Отель». Данные для этого экрана находятся по следующему [API](#).

Сам экран состоит из списка номеров.



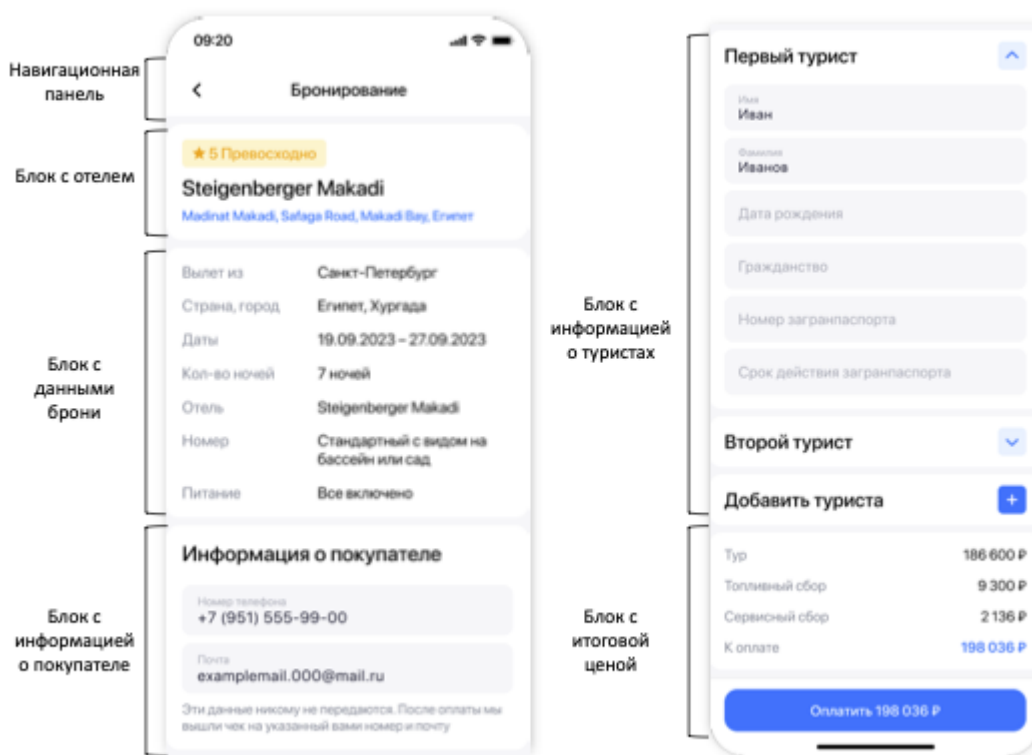
Список элементов:

- **Навигационная панель** – это панель, в которой имеется название экрана и кнопка. Самое название экрана – это название отеля откуда мы перешли. Кнопка назад возвращает на начальный экран «Отель»

- **Блок с номером** – это ячейка у TableView, которая в себе содержит всю необходимую информацию о номере. Кнопка «подробнее о номере» некликабельна (по нажатию не переходит никуда). При нажатии на кнопку «Выбрать номер» любой ячейке всегда осуществляется переход на один и тот же экран «Бронирование» (то есть если номера отеля могут быть разные, но при нажатии на кнопку «Выбрать номер» всегда открываются одни и те же данные на экране «Бронирование»).

## 2.3. Экран «Бронирование»

Этот экран открывается при нажатии на кнопку «Выбрать номер» из экрана «Номер». Данные для этого экрана находятся по следующему [API](#).



Список элементов:

- **Навигационная панель** – это панель, состоящая из названия экрана и кнопки назад, которая возвращает на экран «Номер».
- **Блок с отелем** – содержит рейтинг отеля, название отеля и адрес отеля. При нажатии на адрес ничего не происходит.
- **Блок с данными брони** – содержит город вылета, город прилета, даты брони, кол-во ночей, название отеля, название номера, питание. Все данные нужно получать по API.

- **Блок с информацией о покупателе** – содержит номер телефона и почту, данные обязательно должны быть заполнены. Это текстовые поля, куда пользователь может ввести свои данные. При нажатии на поле номера телефона отработывает автоматическая маска с шаблоном +7 (\*\*\*) \*\*\*\_\*\*\_\*\* и при вводе номера телефона числа подставляются в шаблон (начинать ввод нужно без 7 или 8). Почта проходит валидацию по стандартному шаблону.
- **Блок с информацией о туристах** – содержит в себе изначально вкладку «Первый турист» и кнопку «Добавить туриста» (первый турист обязательно должен ввести свои данные и все данные должны быть заполнены). Если пользователь захочет добавить еще одного туриста, то нажимает на «+» и добавляется еще одна вкладка с названием «Второй турист», еще одного «Третий турист» и так далее. Вкладки скрываются и раскрываются.
- **Блок с итоговой ценой** – содержит цену за тур, топливный сбор, сервисный сбор и итоговую цену к оплате. Итоговая сумма формируется путем сложения всех трех позиций. Все данные берутся из API

## Сценарии использования

### *Пользователь хочет заполнить номер телефона*

1. Пользователь, находясь на экране «бронирование», хочет заполнить номер телефона в блоке «информация о покупателе». Кликает на поле
2. Автоматически поле заполняется маской +7 (\*\*\*) \*\*\*\_\*\*\_\*\*
3. Пользователь начинает вводить номер телефона, начиная с цифры 9
4. Цифра 9 встает на место первой звездочки
5. Пользователь продолжает ввод цифр и по мере заполнения меняются соответствующие звездочки на цифры

### *Пользователь хочет заполнить почту*

1. Пользователь, находясь на экране «бронирование», хочет заполнить почту в блоке «информация о покупателе». Кликает на поле
2. Вводит валидную почту, например, john123@mail.ru

### *Альтернативный сценарий*

1. Пользователь вводит некорректную почту, например, джон21@mail.ru или john/12@mail.ru, или john12@mailru
2. Пользователь, выйдя из режима ввода данных в поле, хочет заполнить остальную информацию
3. Высвечивается ошибка о том, что почта введена некорректно (просто закрашивается поле в цвет по шестнадцатеричному коду (HEX) #EB5757 с непрозрачностью 15%)

### *Пользователь хочет ввести данные о туристе*

1. Пользователь, находясь на экране «бронирование», хочет заполнить данные о туристе. Раскрывает вкладку, если она скрыта и начинает вводить данные о туристе
2. Пользователь вводит все необходимые данные
3. Пользователь добавляет нового туриста или заканчивает заполнение данных

### *Альтернативный сценарий*

1. Пользователь заполняет частично данные или не заполняет данные вовсе
2. Пользователь нажимает на кнопку «оплатить»
3. Высвечивается ошибка о том, что данные необходимо заполнить. Так же, как и с полем почты, закрашивать поле в тот же цвет по HEX коду.

### *Пользователь хочет оплатить бронь*

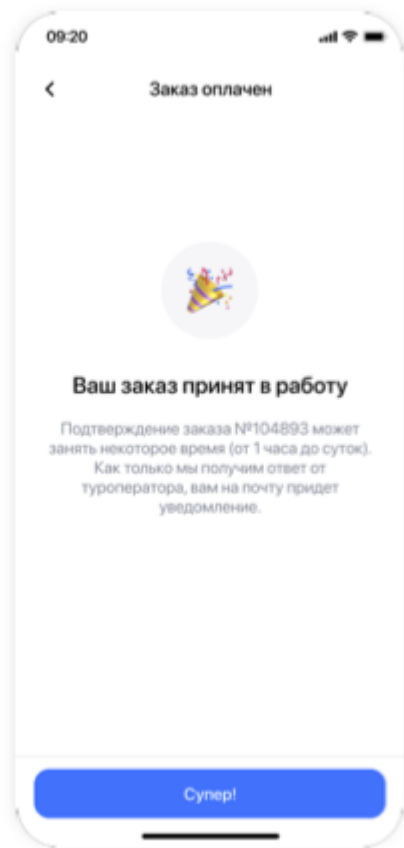
1. Пользователь, находясь на экране «бронирование», хочет оплатить бронь и нажимает на кнопку «оплатить»
2. Открывается экран «Оплачено»

### *Альтернативный сценарий*

1. Оплата не проходит, так как пользователь не заполнил соответствующие поля
2. Все незаполненные поля закрашивает в цвет ошибки

## 2.4. Экран «Оплачено»

Этот экран открывается при успешном сценарии на экране «Бронирование» при нажатии на кнопку «Оплатить». API нет, просто отрисовать по Figma.



При нажатии на кнопку «Супер!» осуществляется переход на начальный экран – «Отель». Либо пользователь может вернуться назад на экран «Бронирование», нажав на кнопку назад в навигационной панели.

Номер заказа каждый раз должен быть разным (можно поставить рандом).

### 3. Рекомендуемый стек технологий

iOS:

- Swift
- MVVM + Coordinator
- SwiftUI
- Combine