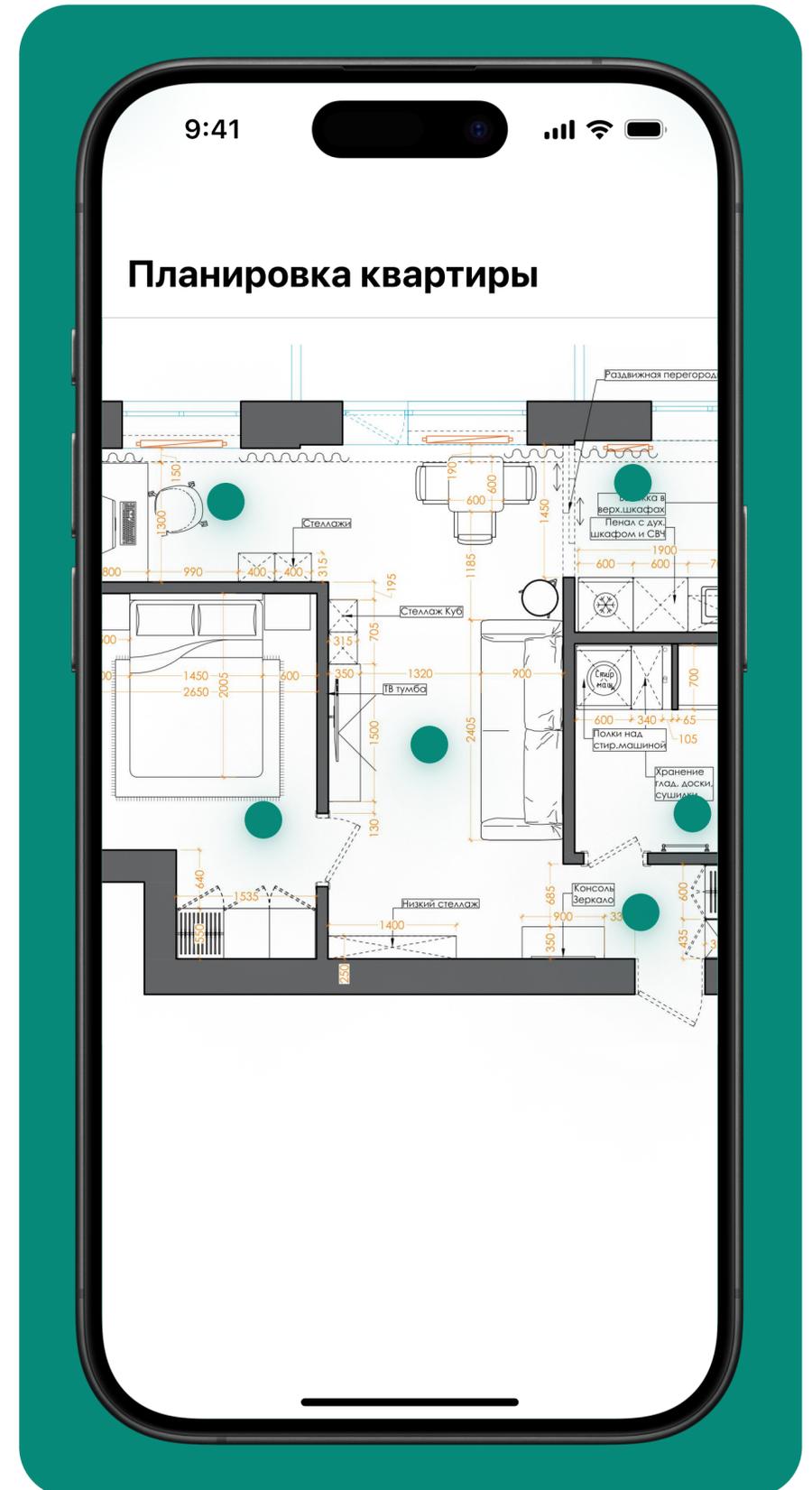




СОЗДАНИЕ INDOOR КАРТЫ ЗДАНИЯ на Flutter

Павел Гершевич Senior Flutter developer





02 / 66

10+ лет в мобильной разработке

6+ лет с Flutter

20+ проектов на Flutter

Собственный курс в самом крупном университете Сибири





03 / 66

О чем поговорим

01



Что такое indoor
map?

Кому и зачем
оно нужно?

02



С чем мы
столкнулись при
разработке?

03



Какие еще
проблемы могут
быть?



04 / 66

01

4 TO TAKE
indoor
map?





05 / 66



Indoor карта

внутренняя карта здания,
которая содержит
информацию, где находятся
комнаты и точки интереса



06 / 66



Точки интереса

места, которые отмечаются маркерами и могут содержать полезную для пользователя информацию



07 / 66



В каких
приложениях
нужно?



08 / 66

В каких приложениях нужно?

Квесты в реальности

Навигационные приложения

Приложения для путешественников

Приложения для архитекторов, дизайнеров интерьеров и их клиентов

Образовательные учреждения

Бронирование номеров, рабочих мест



09 / 66



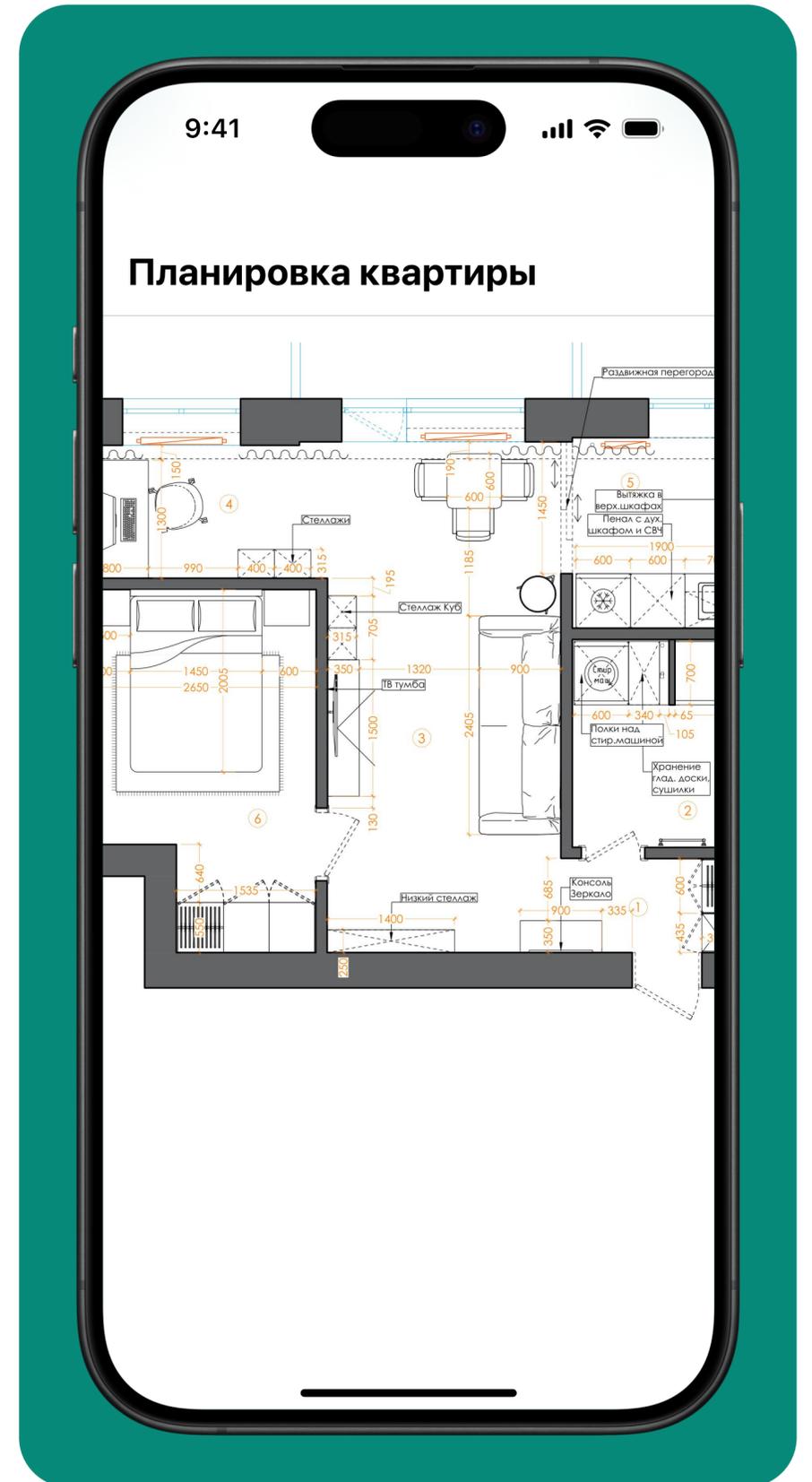
Что умеет indoor map?



10 / 66

Какой функционал бывает

Просто показ карты



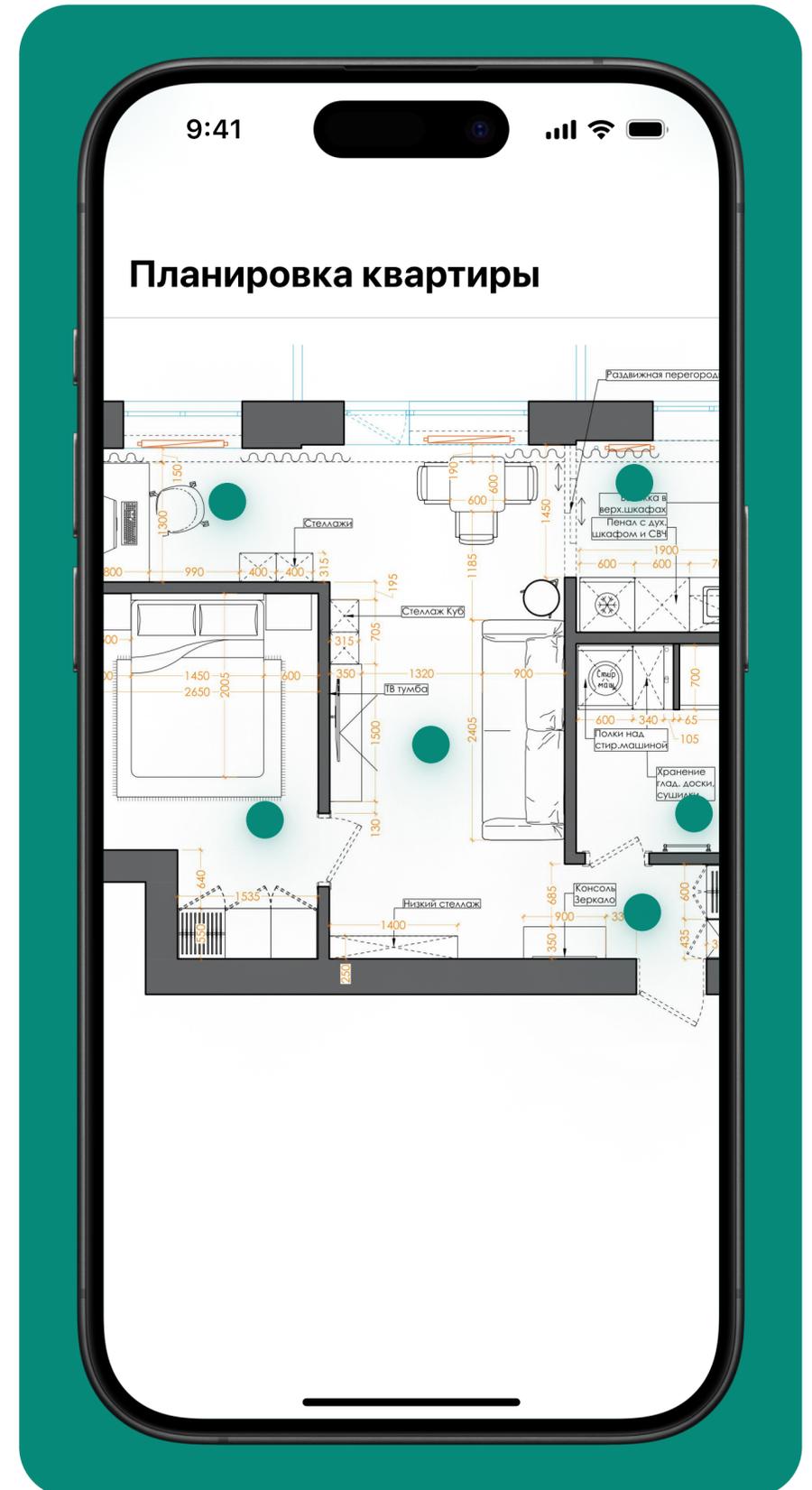


11 / 66

Какой функционал бывает

Просто показ карты

Точки интереса (магазины, кабинеты и т. д.)





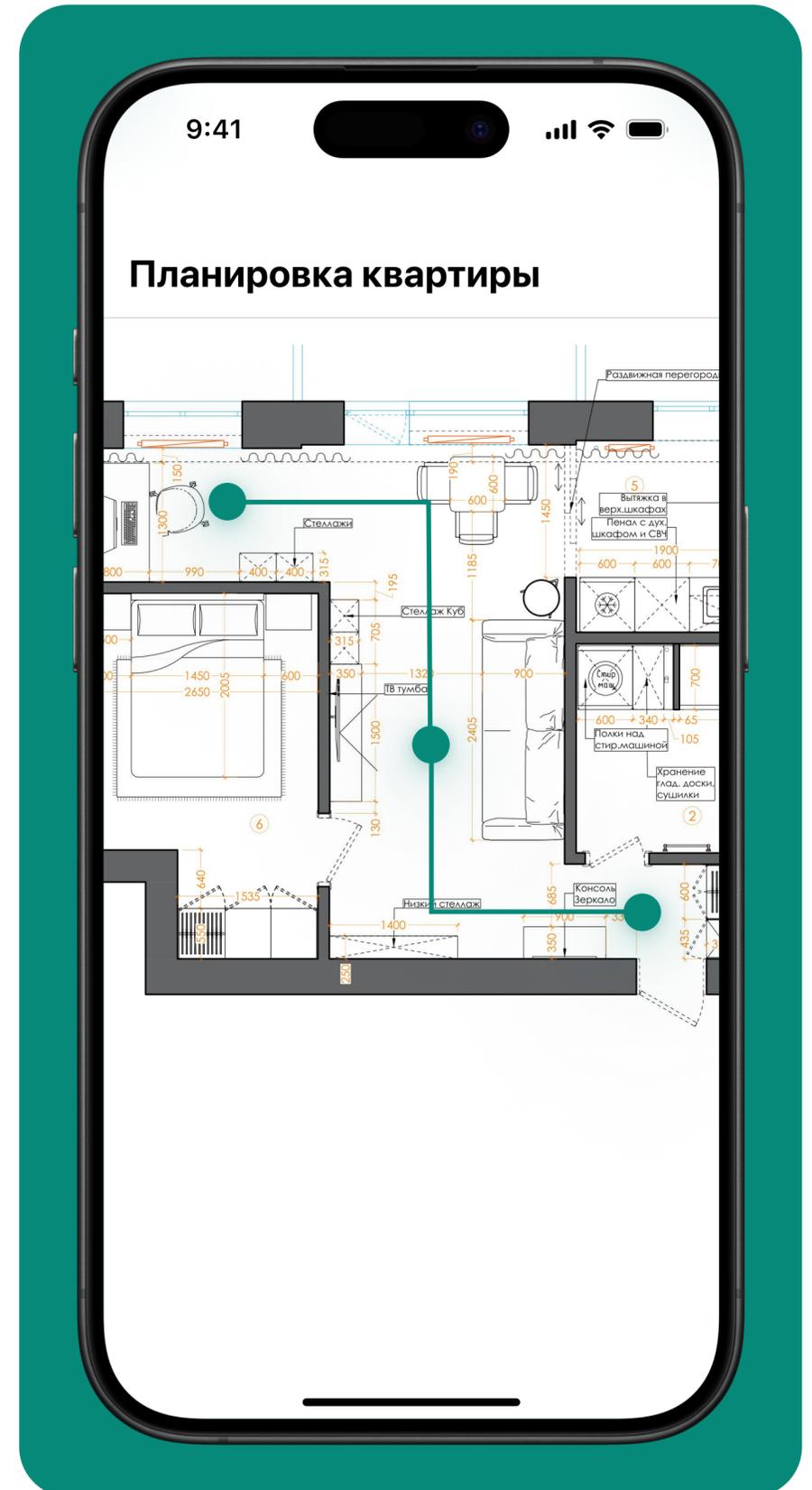
12 / 66

Какой функционал бывает

Просто показ карты

Точки интереса (магазины, кабинеты и т. д.)

Навигация по зданию

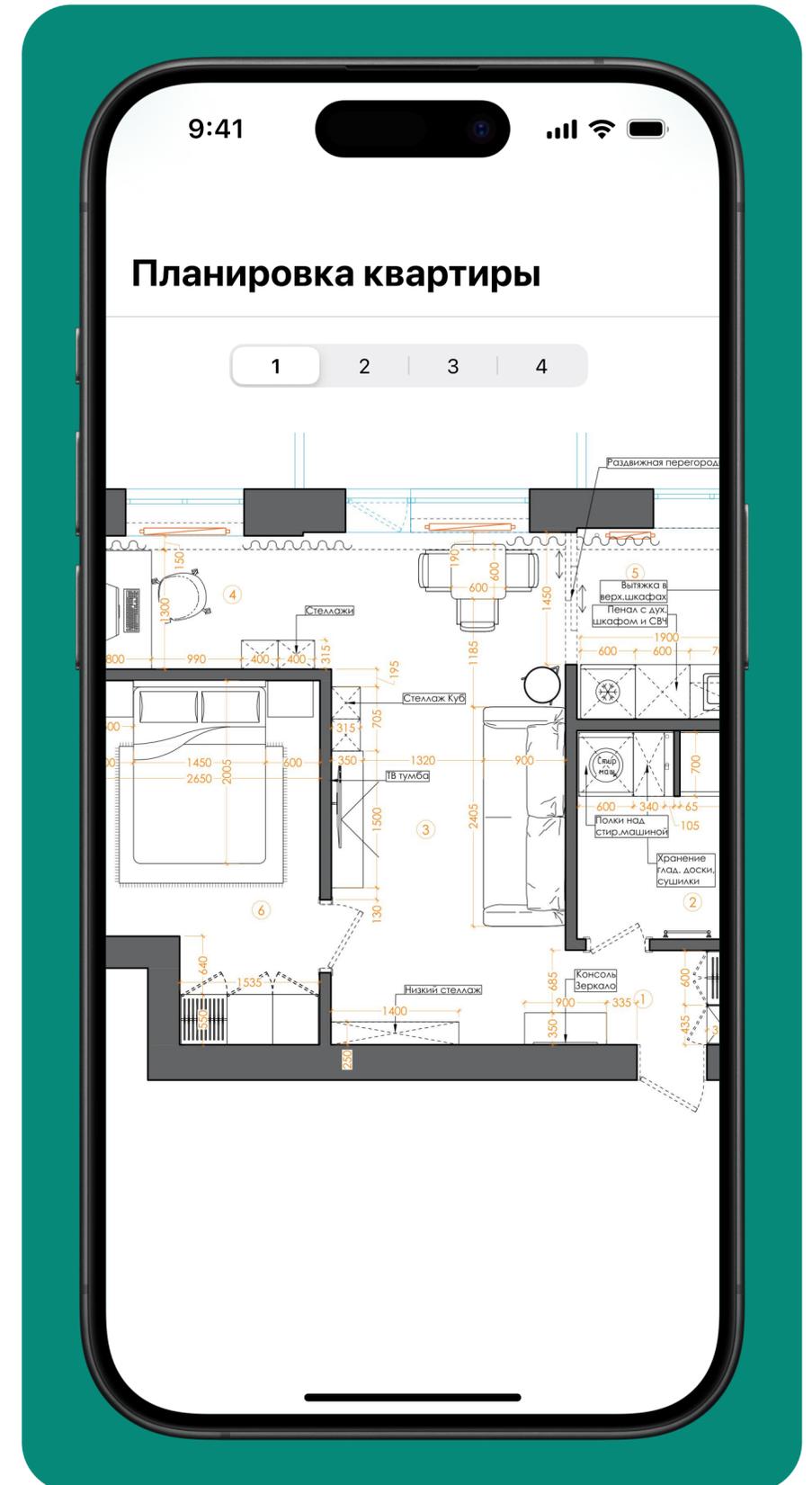


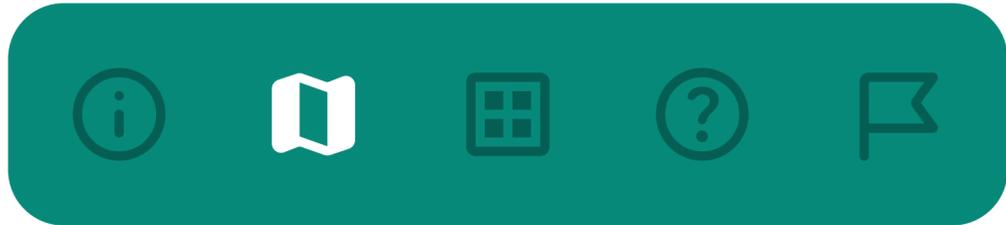


13 / 66

Какой функционал бывает

- Просто показ карты
- Точки интереса (магазины, кабинеты и т. д.)
- Навигация по зданию
- Переключение между этажами





14 / 66



Способы создания indoor map



15 / 66

Как можно сделать indoor map?

Использовать подложку из реальной карты и при определенном уровне приближения показывать из GeoJSON

Отображать только внутреннюю карту здания



16 / 66

Использование GeoJSON

Плюсы

Привязка к
координатам

Можно сделать вместо точки
ПОЛИГОН

Минусы

Сложнее реализация



17 / 66

Когда ИСПОЛЬЗОВАТЬ GeoJSON?

- Несколько зданий рядом

- Большие площади

- Нужно отображение полигонами



18 / 66

Использование схемы здания

Плюсы

Детализация -
изначально

Простая реализация

Минусы

Точки интереса - именно точками

Проблемы с
геопозиционированием



19 / 66

Когда
ИСПОЛЬЗОВАТЬ
ТОЛЬКО indoor map?

Одно здание

Нужны только точки на карте

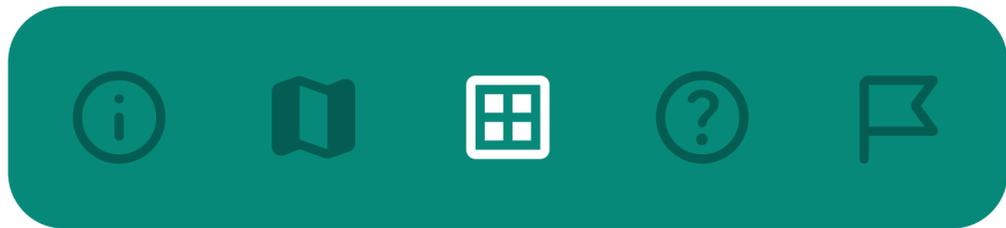


20 / 66

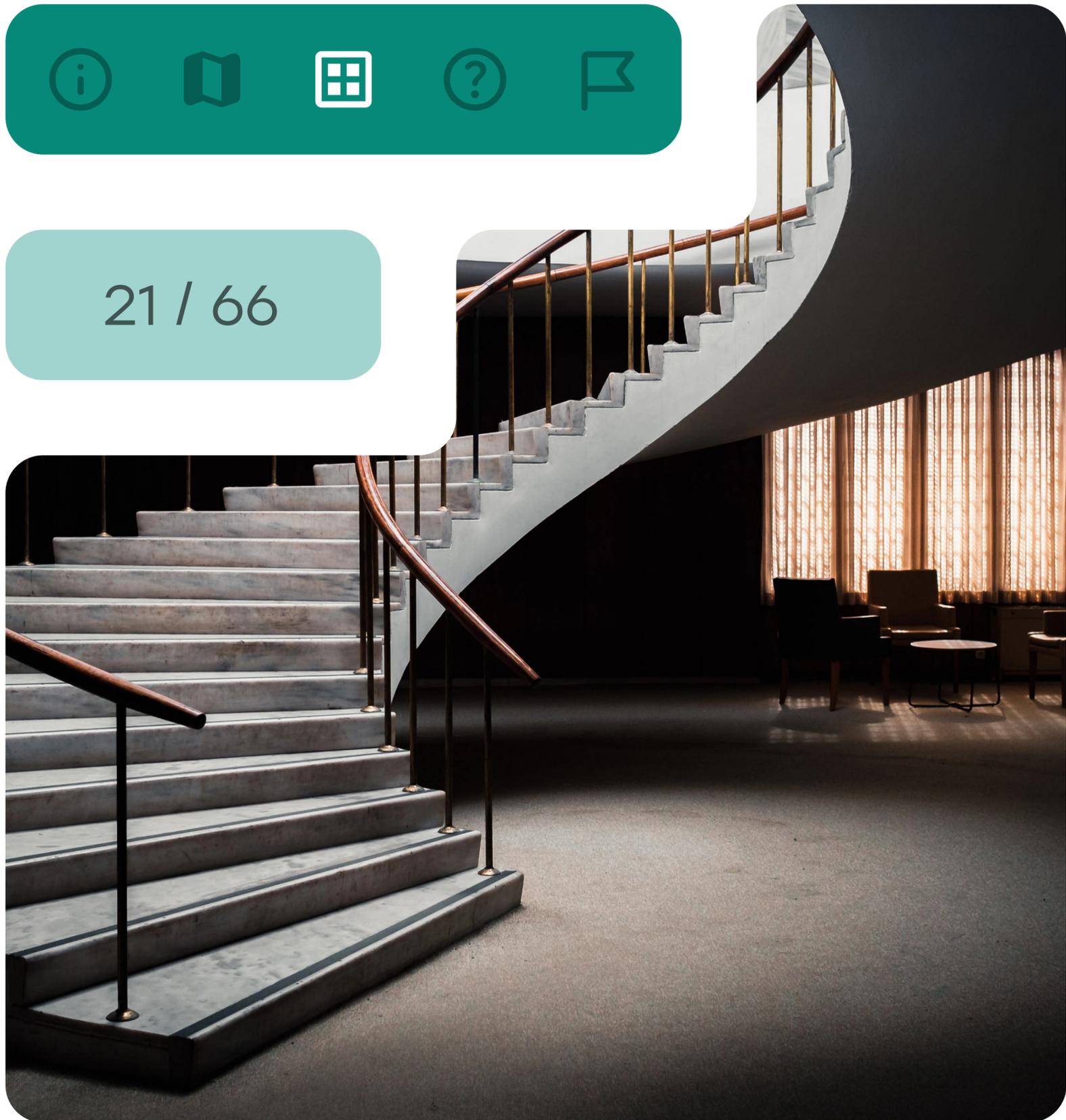
02

Как мы
попробовали
это сделать





21 / 66



Отображение
карты одного
этажа
здания
на экране



22 / 66

Какой функционал заложили?

- Вывод SVG изображения на экран

- pinch to zoom

- double tap to zoom

- Поворот карты с привязкой к сторонам света



23 / 66

Interactive Viewer

```
InteractiveViewer(  
  boundaryMargin:  
    const EdgeInsets.all(double.infinity),  
  minScale: 0.75,  
  maxScale: 3,  
  child: SvgPicture(  
    ...  
  ),  
),
```



24 / 66

Interactive Viewer

```
InteractiveViewer(  
  boundaryMargin:  
    const EdgeInsets.all(double.infinity),  
  minScale: 0.75,  
  maxScale: 3,  
  child: SvgPicture(  
    ...  
  ),  
),
```



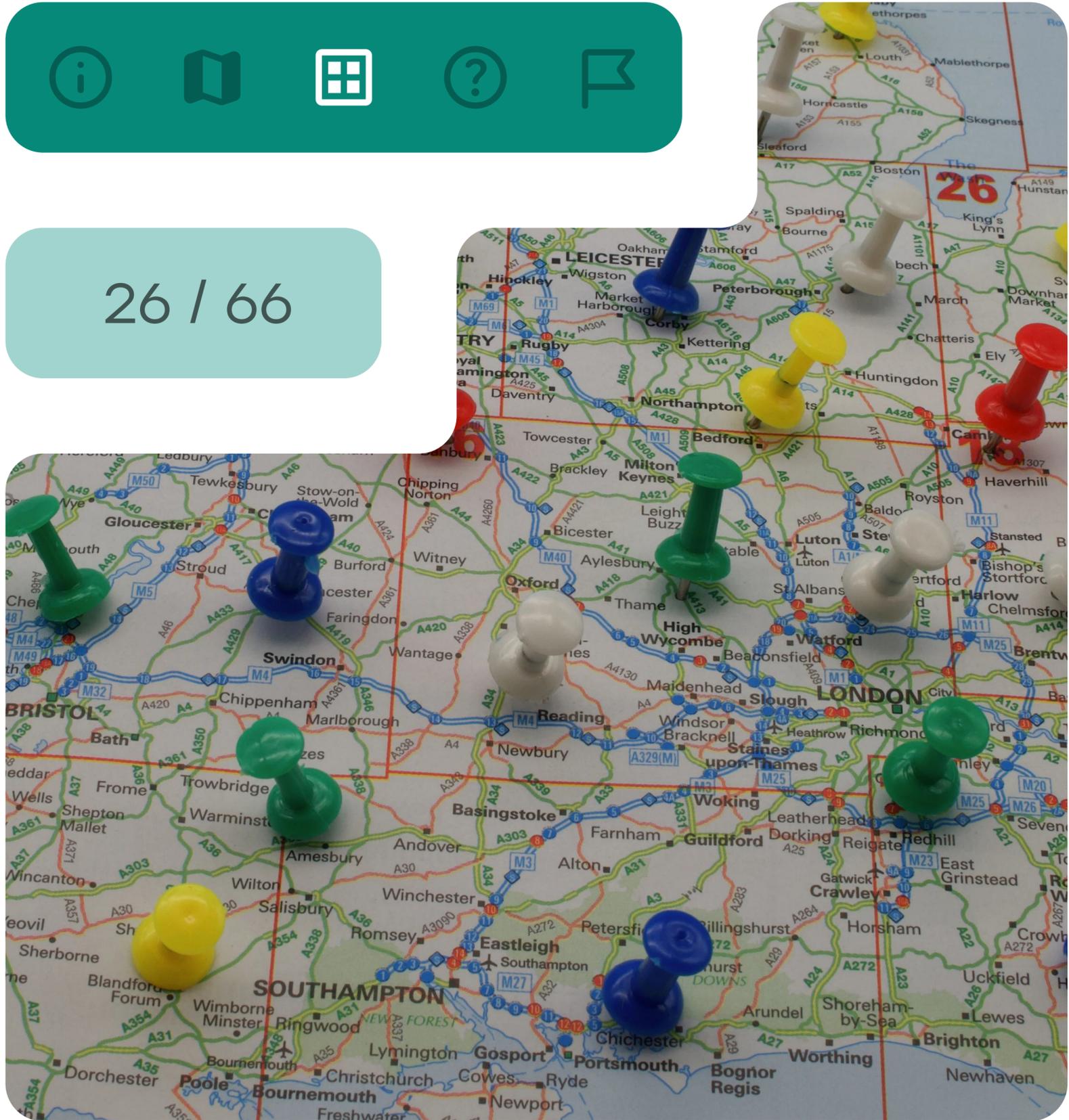
25 / 66

Interactive Viewer

```
InteractiveViewer(  
  boundaryMargin:  
    const EdgeInsets.all(double.infinity),  
  minScale: 0.75,  
  maxScale: 3,  
  child: SvgPicture(  
    ...  
  ),  
),
```



26 / 66



Отображение маркеров и областей на карте



27 / 66

Отображение маркеров

```
InteractiveViewer(  
  boundaryMargin:  
    const EdgeInsets.all(double.infinity),  
  minScale: 0.75,  
  maxScale: 3,  
  child: SvgPicture(  
    ...  
  ),  
),
```



28 / 66

Отображение маркеров

```
InteractiveViewer(  
  boundaryMargin:  
    const EdgeInsets.all(double.infinity),  
  minScale: 0.75,  
  maxScale: 3,  
  child: Stack(  
    children: [  
      SvgPicture(...),  
      ...markers,  
    ],  
  ),  
)
```



29 / 66

Проблема №1

Отображение
маркера в
нужном месте

Мы не знаем размер
карты, который нам
приходит

Соответственно, мы не можем
узнать точное место



```
double maxDeltaDx = mapSize.width / 2;  
double maxDeltaDy = mapSize.height / 2;  
  
double newDx = (maxDeltaDx) * position.dx;  
double newDy = (maxDeltaDy) * position.dy;  
  
if (newDx != 0) {  
    newDx = maxDeltaDx + newDx;  
} else {  
    newDx = maxDeltaDx;  
}
```



```
double maxDeltaDx = mapSize.width / 2;  
double maxDeltaDy = mapSize.height / 2;  
  
double newDx = (maxDeltaDx) * position.dx;  
double newDy = (maxDeltaDy) * position.dy;  
  
if (newDx != 0) {  
    newDx = maxDeltaDx + newDx;  
} else {  
    newDx = maxDeltaDx;  
}
```



```
double maxDeltaDx = mapSize.width / 2;  
double maxDeltaDy = mapSize.height / 2;  
  
double newDx = (maxDeltaDx) * position.dx;  
double newDy = (maxDeltaDy) * position.dy;  
  
if (newDx != 0) {  
    newDx = maxDeltaDx + newDx;  
} else {  
    newDx = maxDeltaDx;  
}
```



```
if (newDy < 0) {  
    newDy = maxDeltaDy + newDy.abs();  
} else if (newDy > 0) {  
    newDy = maxDeltaDy - newDy;  
} else {  
    newDy = maxDeltaDy;  
}
```

```
return Offset(newDx, newDy);
```



```
if (newDy < 0) {  
    newDy = maxDeltaDy + newDy.abs();  
} else if (newDy > 0) {  
    newDy = maxDeltaDy - newDy;  
} else {  
    newDy = maxDeltaDy;  
}
```

```
return Offset(newDx, newDy);
```



35 / 66

Отображение маркера

```
Positioned(  
  left: newOffset.dx,  
  top: newOffset.dy,  
  child: SizedBox(  
    width: mapObjectData.size.width,  
    height: mapObjectData.size.height,  
    child: mapObjectWidget,  
  ),  
);
```



36 / 66

Отображение маркера

```
Positioned(  
  left: newOffset.dx,  
  top: newOffset.dy,  
  child: SizedBox(  
    width: mapObjectData.size.width,  
    height: mapObjectData.size.height,  
    child: mapObjectWidget,  
  ),  
);
```



37 / 66

Нужен якорь

Можно сделать в виде точки или любой другой картинки

Проблема №2

Обычный маркер находится ниже, чем точка



38 / 66

Добавляем
якорь

```
MapObject(  
  x: 120,  
  y: 48,  
  width: 30,  
  height: 30,  
  anchor: Alignment.bottomCenter,  
),
```



39 / 66

Добавляем
якорь

```
MapObject(  
  x: 120,  
  y: 48,  
  width: 30,  
  height: 30,  
  anchor: Alignment.bottomCenter,  
),
```



40 / 66

Добавляем
якорь

```
MapObject(  
  x: 120,  
  y: 48,  
  width: 30,  
  height: 30,  
  anchor: Alignment.bottomCenter,  
),
```



41 / 66

Добавляем
якорь

```
MapObject(  
  x: 120,  
  y: 48,  
  width: 30,  
  height: 30,  
  anchor: Alignment.bottomCenter,  
),
```



42 / 66

Добавляем якорь

```
double left = x;  
double top = y;  
switch(anchor) {  
    case Alignment.center:  
        left = x + width / 2;  
        top = y + height / 2;  
        break;  
    case Alignment.bottomCenter:  
        left = x + width / 2;  
        break;  
    ...  
}
```



43 / 66

Добавляем якорь

```
double left = x;  
double top = y;  
switch(anchor) {  
  case Alignment.center:  
    left = x + width / 2;  
    top = y + height / 2;  
    break;  
  case Alignment.bottomCenter:  
    left = x + width / 2;  
    break;  
  ...  
}
```



44 / 66

Добавляем якорь

```
double left = x;  
double top = y;  
switch(anchor) {  
    case Alignment.center:  
        left = x + width / 2;  
        top = y + height / 2;  
        break;  
    case Alignment.bottomCenter:  
        left = x + width / 2;  
        break;  
    ...  
}
```



45 / 66

Добавляем якорь

```
Positioned(  
  left: left,  
  top: top,  
  child: SizedBox(  
    width: width,  
    height: height,  
    child: ...,  
  ),  
);
```



46 / 66

Добавляем
якорь

```
Positioned(  
  left: left,  
  top: top,  
  child: SizedBox(  
    width: width,  
    height: height,  
    child: ...,  
  ),  
);
```



47 / 66

Добавляем якорь

```
Positioned(  
  left: left,  
  top: top,  
  child: SizedBox(  
    width: width,  
    height: height,  
    child: ...,  ),  
);
```



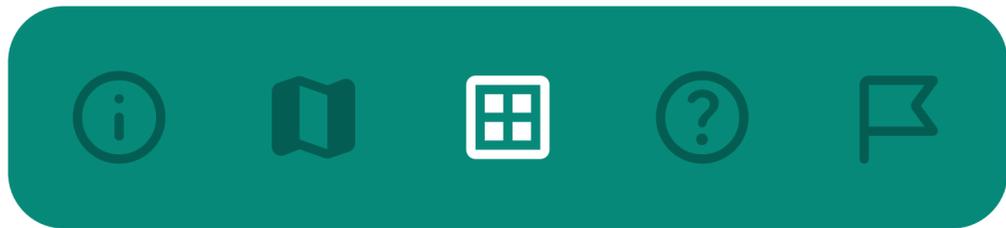
48 / 66

Проблема №3

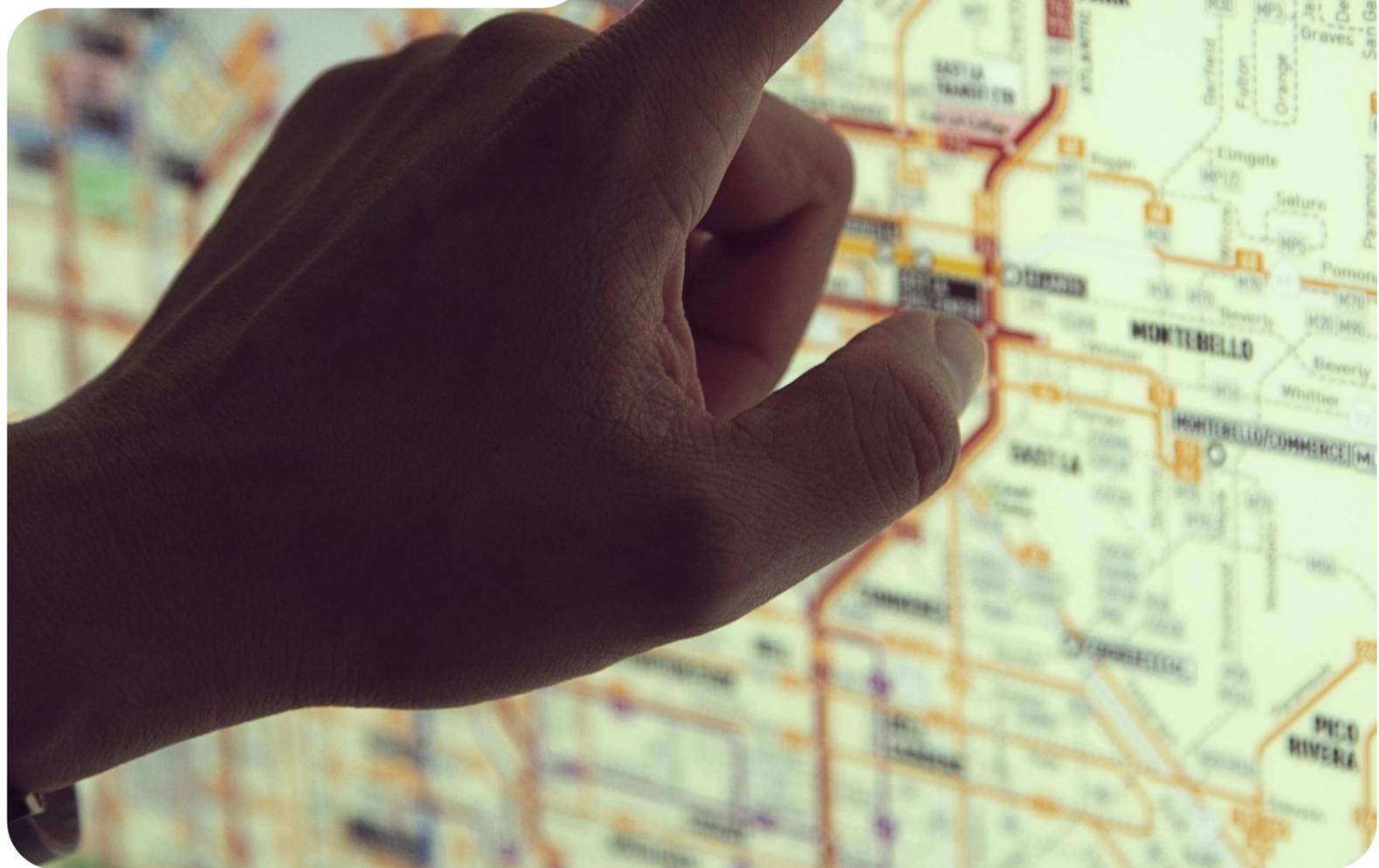
Несколько
маркеров в
одном месте
перекроют
друг друга

Добавляем
функционал по
объединению точек

Нужно 2 алгоритма -
для объединения и для получения
уровня зума



49 / 66



Определение местоположения пользователя



50 / 66

Вариант №1

**Пользователь
сканирует
что-либо**

Сканирование
QR-кода

Сканирование
NFC-метки

Ввод какого-либо кода



51 / 66

Сканирование физического объекта

Каждой точке
выдается ID

По ID генерируем QR-код /
зашиваем в NFC метку

Ставим QR-код / NFC метку
около точки интереса



52 / 66

Сканирование физического объекта

Пользователь
сканирует

Приложение получает
ID местоположения

Запрашиваем
точку с сервера

Отображаем
пользователя на карте





53 / 66

Вариант №2

Использование традиционной навигации

Только для
больших зданий

Низкая точность
позиционирования

Нужно знать координаты
здания и масштаб карты



54 / 66

Сканирование физического объекта

Запрашиваем
разрешение

Получаем данные
от сенсоров

Вычисляем положение
в здании

Отображаем
пользователя на карте



55 / 66

Вариант №3

Использование
Wi-Fi и
Bluetooth

Сложность
алгоритма

Достаточная точность, если
известно местоположение
источников



56 / 66

Использование
beacons

Вариант №3

Использование
Wi-Fi и
Bluetooth



57 / 66

03

С чем еще
можно
столкнуться





58 / 66



Выбор нескольких зданий и этажей



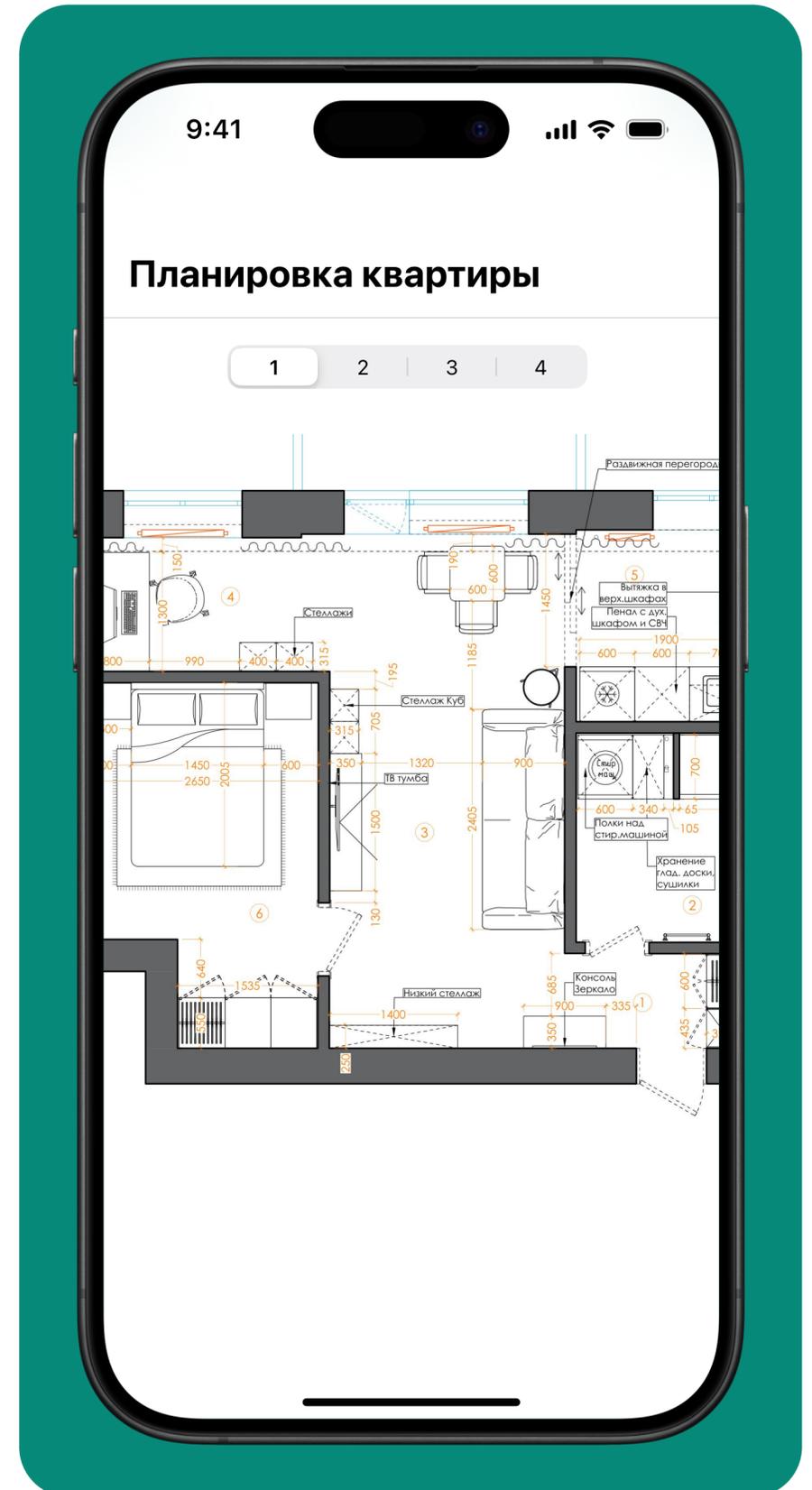
59 / 66

Выбор здания и этажа

Переключатель этажей

Переключатель зданий

Автоматическое определение





60 / 66



Прокладывание маршрутов



61 / 66

Прокладывание маршрутов

Серверная часть

Добавление
маршрутных точек

Добавление расстояния
между точками

Формирование
графа

Алгоритм поиска
пути в графе





62 / 66

Прокладывание маршрутов Клиентская часть

Пользователь
выбирает точки

API формирует путь и
отдает нам маршрут

Выводим путь
при помощи линий

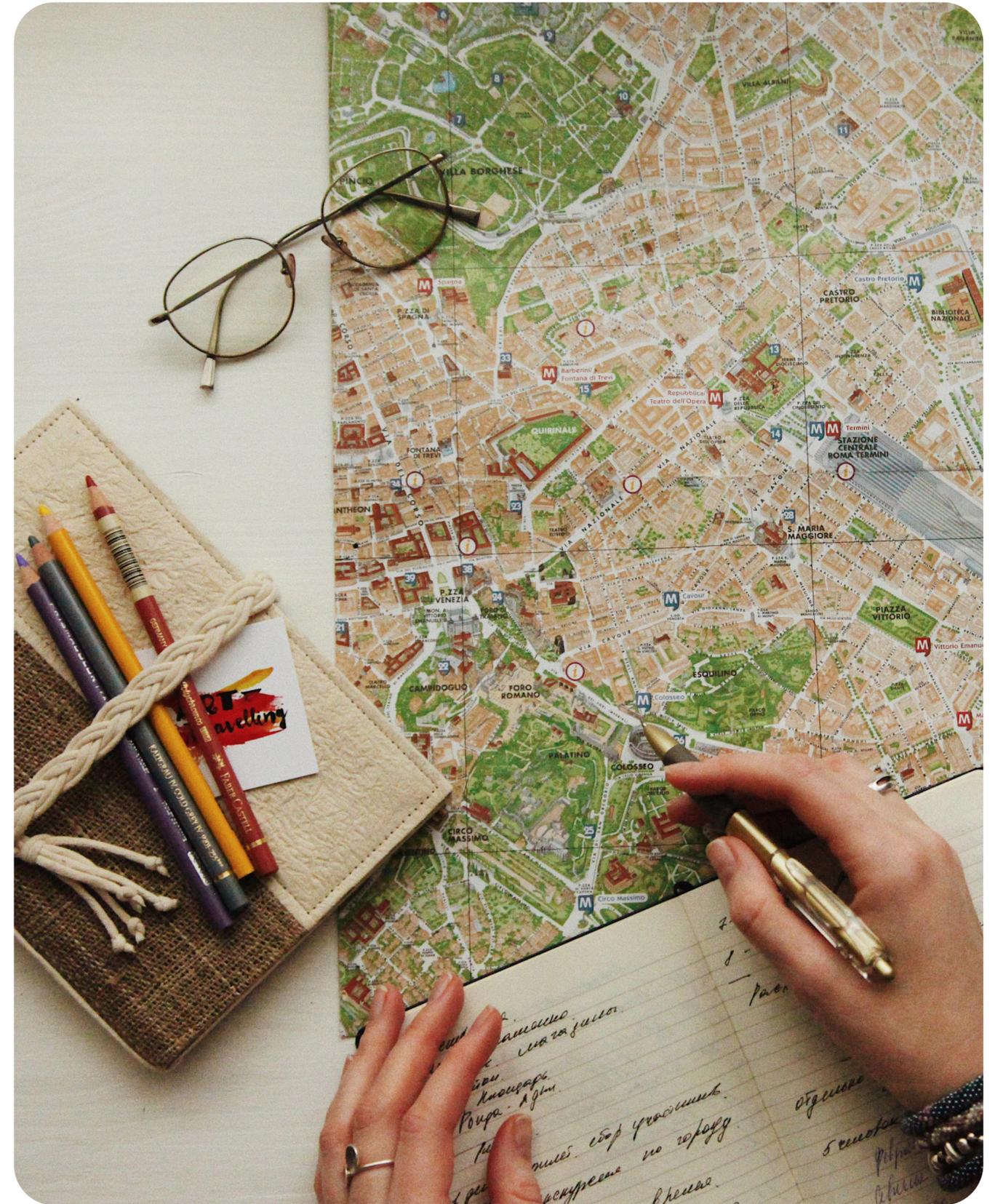




63 / 66

00

ИТОГИ





Узнаем координаты углов изображения и размер карты

Для маркеров делаем якорь (можно задать один раз через Provider или тему)

Выбираем способ позиционирования исходя из потребностей



65 / 66

ИТОГИ

Хранение данных лучше доверять Backend

Прокладывание маршрутов с большим количеством точек может быть затратно



66 / 66

Спасибо за внимание!

Telegram
@ftl_notes

GitHub
github.com/FogNature

