

1. **Доделать показ тултипов о диапазоне цен на общей карте.** Вид тултипа - как в примере: <http://ticketsale.biletiastana.kz/event/789>
2. **Доделать вывод в подробные попапы трибун.** Для этого дизайнеру необходимо расположить шарики подряд в каждом ряду, так как это сделано в структуре с бека:

```
{
  "SectorName":"E1bottom",
  "SectorRu":"E1(низ)",
  "minPrice":0,
  "maxPrice":0,
  "seatsLeft":294,
  "sector_rows":[
    {
      "RowN":1,
      "tickets":[
        {
          "Barcode":" 339677759446",
          "TicketID":19982,
          "IDSeat":20104,
          "SeatN":1,
          "RowN":1,
          "Price":0,
          "IDStatus":3,
          "StatusName":"Свободный",
          "Tribune": "",
          "SectorName":"E1bottom",
          "SectorRu":"E1(низ)"
        },
        {
          "Barcode":" 227161620744",
          "TicketID":19983,
          "IDSeat":20105,
          "SeatN":2,
          "RowN":1,
          "Price":0,
          "IDStatus":3,
          "StatusName":"Свободный",
          "Tribune": "",
          "SectorName":"E1bottom",
          "SectorRu":"E1(низ)"
        },
        ...
      ]
    },
    {
```

```

"RowN":2,
"tickets":[
  {
    "Barcode":" 245644899551",
    "TicketID":19991,
    "IDSeat":20113,
    "SeatN":1,
    "RowN":2,
    "Price":0,
    "IDStatus":3,
    "StatusName":"Свободный",
    "Tribune": "",
    "SectorName":"E1bottom",
    "SectorRu":"E1(низ)"
  },
  {
    "Barcode":" 636259602383",
    "TicketID":19992,
    "IDSeat":20114,
    "SeatN":2,
    "RowN":2,
    "Price":0,
    "IDStatus":3,
    "StatusName":"Свободный",
    "Tribune": "",
    "SectorName":"E1bottom",
    "SectorRu":"E1(низ)"
  },
  ...
]
},
...
]
}

```

т.е. мы присваиваем каждому ряду в SVG файле номер ряда Лучше всего - дата атрибутами: data-line="3" - ряд, data-seat="3" - место.

3. При наведении на место показываем тултип с информацией о цене, статусе и пр. из json. Вид тултипа - как в примере: <http://ticketsale.biletiastana.kz/event/789>

4. При клике на свободное место фон становится черным и появляется галочка выбора. Дизайнеру - отрисовать белую галочку.

5. После выбора информация о билете появляется в новом диалоговом окне внизу страницы. Значения выбранных билетов забиваются в массив. У диалогового окна должен быть z-index: 9999 чтобы перекрыть fancybox.

6. В окне с выбранными билетами должен быть список купленных билетов в виде тегов с возможностью удаления (посылается соответствующий запрос и значение удаляется из массива). Кроме того, должна быть цена и кнопка "Купить" при нажатии на которую посылается запрос на сервер для проверки актуальности статуса выбранных билетов.

Цена должна пересчитываться в зависимости от количества билетов.

7. Если после нажатия кнопки покупки статус success - выводим попап что покупка совершена, если пока выбирали билет его статус изменился - выводим попап с извинениями и предложением выбрать заново.

Технологии:

Необходимо знать JQuery и SVG. Просто JavaScript тоже знать не мешало бы. Также подключена резина на device.js поэтому все стили задаются не в px а в rem - это задел на будущую адаптивность. 1rem = 10px

Стили, скрипты и svg лежат в папке public. При верстке используется less. Собирается все гульпом (gulpfile.js в корне). Библиотеки подключаются там же. Все зависимости подтягиваются автоматически, нужно только выполнить в корне npm i

Сама страница со схемой лежит в views/eventmap.ejs

Когда дизайнер дорисует детальные трибуны, их нужно положить в public/images/sectors заменив те которые там сейчас. Имена у новых файлов должны быть такими же как у тех что там сейчас, т.к. схема подтягивается аяксом с использованием имени файла в url