**Краткое описание**

Данное приложение представляет собой сервис по просмотру представлений некоторого стационарного цирка и по приобретению онлайн билетов (бронированию мест) на представления в нем.

Приложение декомпозировано на следующие составные сервисы:

* «Афиша Цирка» («Circus Timetable Service»)
* «Бронирование Сеансов» (“Session Booking Service”), включая «Обработчик Бронирований» («Booking Processor»)
* «Оплата» («Payment Service»)
* «E-mail Нотификации» («Mailing Service»)

Пользователь (SPA приложение) взаимодействуют с системой по REST / HTTPS.

Процесс бронирования внутри представляет собой асинхронный процесс, состоящий из нескольких шагов и оркестрируемый «Обработчиком Бронирований» (являющимся составным компонентом сервиса «Бронирование Сеансов»). На время выполнения бронирования пользователю отображается заставка “In progress…” (также предполагается возможность получения пользователем / отображения текущего статуса бронирования на любом его этапе либо после его завершения).

При взаимодействии сервисов между собой предполагается использовать межсервисную авторизацию.

Сервисы взаимодействуют между собой преимущественно через шину данных.

**Сущности / понятия предметной области**

1. Представление (Show)
2. Сеанс представления (Show Session)
3. Место в зрительном зале (Seat)
4. Бронирование сеанса (Session Booking)
5. Статус бронирования сеанса (Session Booking Status)
6. Платёж (Payment)
7. E-mail нотификация (E-mail Notification)

**Пользовательские сценарии**

* **Пользовательский сценарий «Просмотр представлений на текущий день»**

**Есть** сайт цирка.  
**И** есть пользователь, зашедший на этот сайт.  
**И** пользователь открывает страницу со списком представлений на текущий день.  
**Тогда** пользователю отображается список представлений на текущий день.  
**И** в карточке каждого представления пользователь видит:

1. Название представления.
2. Краткое описание представления.
3. Изображение.

* **Пользовательский сценарий «Поиск представлений»**

**Когда** пользователь в форме поиска представлений указывает поисковую фразу.  
**Тогда** пользователю отображается список представлений, содержащих поисковую фразу.  
**И** в карточке каждого представления пользователь видит:

1. Название представления.
2. Краткое описание представления.
3. Изображение.

* **Пользовательский сценарий «Просмотр подробной информации о представлении»**

**Когда** пользователь открывает страницу с детальной информацией о представлении.  
**Тогда** пользователю отображается страница с детальной информацией о представлении.  
**И** в карточке представления пользователь видит:

1. Название представления.
2. Подробное описание представления, включая: жанры, дата первой постановки, сведения о цирке-постановщике, премии, список актёров, расписание гастролей, продолжительность представления и т.д.
3. Фотографии с предыдущих представлений.

* **Пользовательский сценарий «Просмотр расписание сеансов представления»**

**Есть** страница с результатами поиска представлений.  
**Либо** есть страница со списком представлений на конкретный день. **Когда** пользователь запрашивает расписание конкретного представления на конкретный день.  
**Тогда** пользователю отображается список сеансов представления на конкретный день.

* **Пользовательский сценарий «Просмотр доступных мест на сеанс»**

**Есть** страница с расписанием сеансов представления на конкретный день.  
**Когда** пользователь запрашивает сведения о доступных местах на выбранный сеанс представления.  
**Тогда** пользователю отображается схема зала со списком доступных мест.  
**И** на данной схеме пользователь видит:

1. Какие места свободны, а какие – заняты.
2. Стоимость мест в зависимости от расположения.
3. Кнопку «Бронировать».

* **Пользовательский сценарий «Бронирование мест на сеанс»**

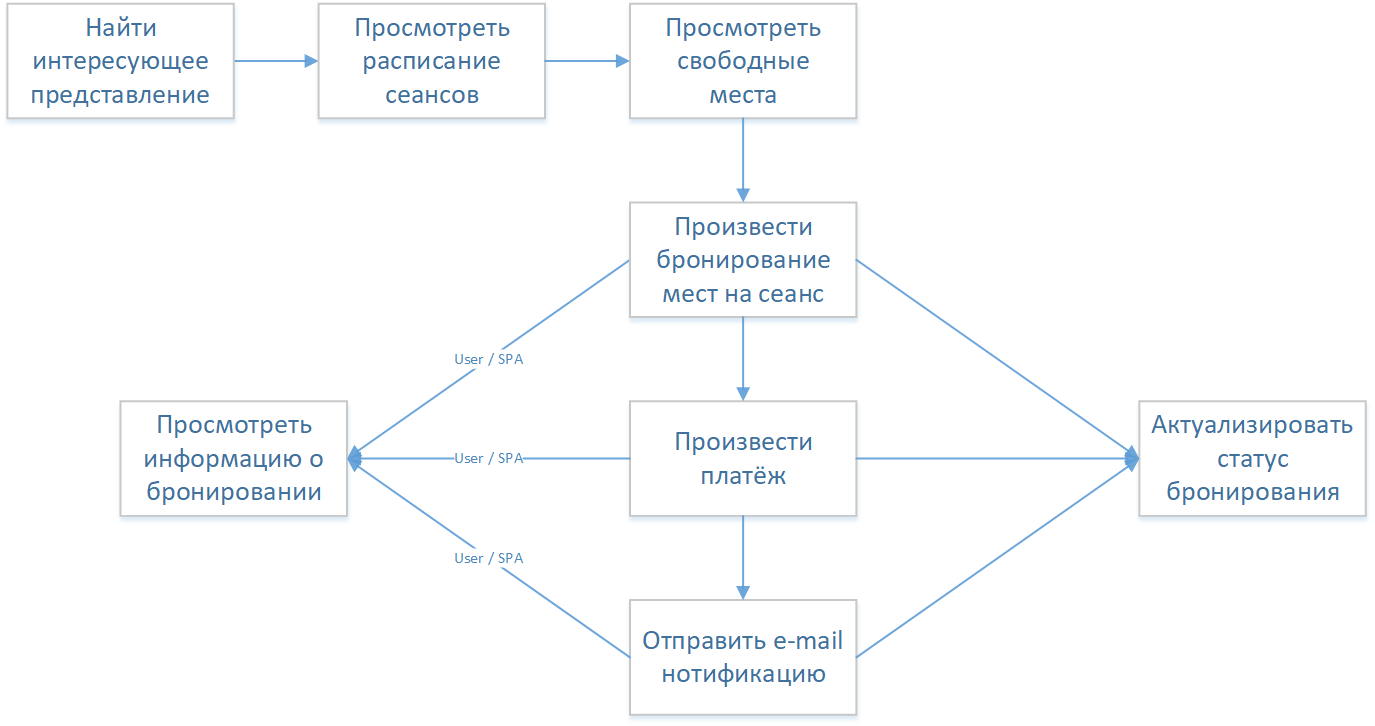
**Есть** страница с информацией о доступных местах на выбранный сеанс представления.  
**Когда** пользователь выбирает свободные места на схеме.  
**И** нажимает кнопку «Бронировать».  
**Тогда** пользователю отображается форма оплаты.  
**И** на форме оплаты пользователь видит следующие элементы:

1. Краткая сводка о бронируемом представлении и сеансе.
2. Поля для ввода контактной информации: Имя, Фамилия, Адрес электронной почты.
3. Поля для указания данных банковской карты.
4. Сумма к оплате.
5. Кнопка «Оплатить».

* **Пользовательский сценарий «Оплата и подтверждение бронирования сеанса»**

**Есть** форма отплаты бронирования выбранных мест на сеанс представления.  
**Когда** пользователь вводит контактную информацию и данные банковской карты.  
**И** нажимает кнопку «Оплатить»  
**Тогда** с банковской карты пользователя списывается указанная сумма.  
**И** подтверждается бронирование выбранных мест.  
**И** на указанный адрес электронной почты приходит нотификация о подтверждении бронирования, содержащая следующую информацию: краткие сведения о представлении и сеансе, выбранные места, данные плательщика, оплаченная стоимость.

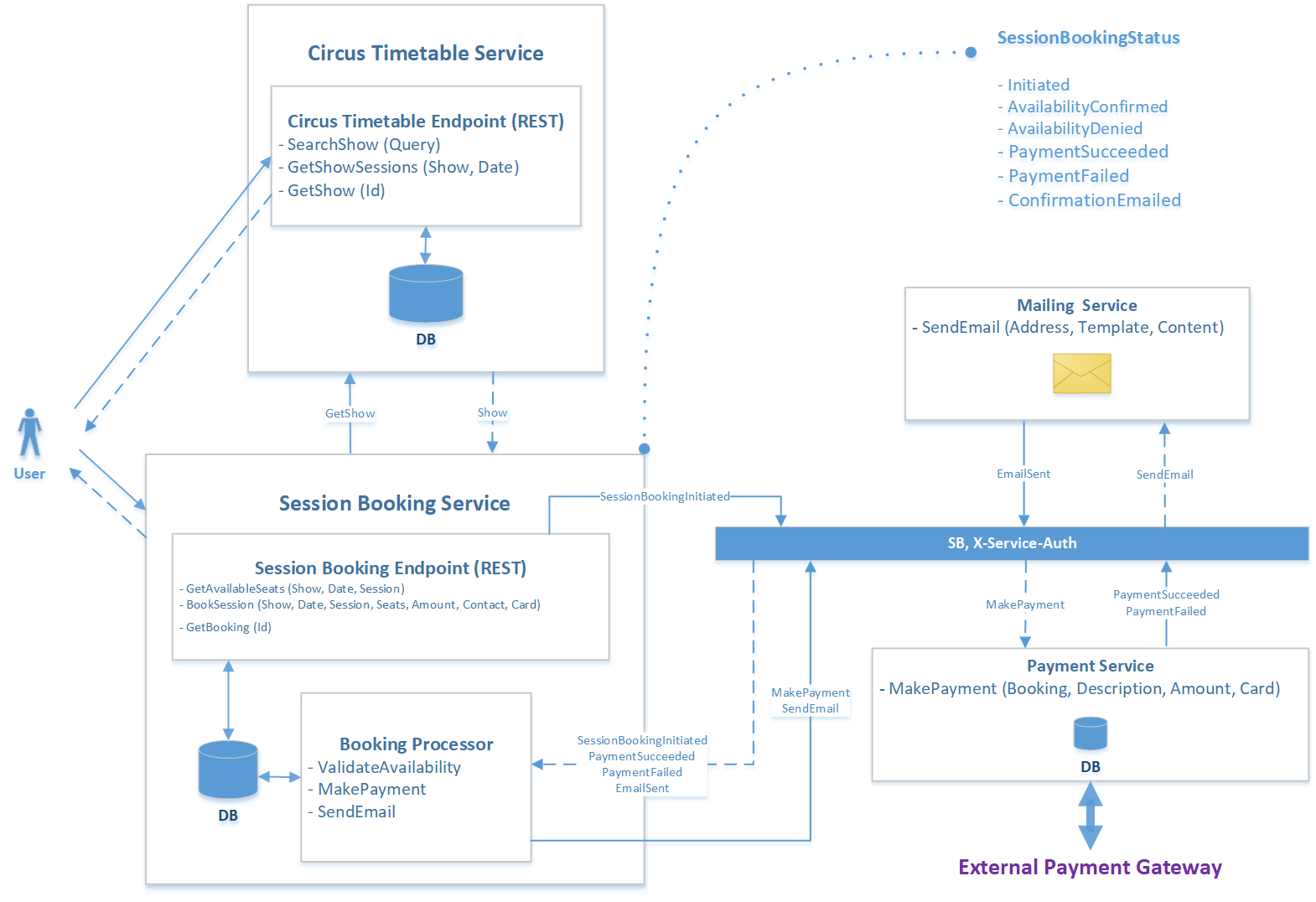
**Бизнес действия**



**Сервисы и их зоны ответственности**

1. **Сервис «Афиша Цирка» (“Circus Timetable”).**
   * Возвращает список представлений на запрошенную дату
   * Выполняет поиск представлений по поисковой фразе
   * Возвращает список сеансов представления на запрошенную дату
   * Возвращает подробное описание запрошенного представления
2. **Сервис «Бронирование Сеансов» (“Session Booking”)**
   * Возвращает список доступных мест на конкретный сеанс представления в указанную дату с указанием с указанием стоимости
   * Инициирует бронирование указанных мест на представление по соответствующей стоимости
   * Возвращает описание / текущий статус указанного бронирования
   * Компонент “Обработчик Бронирований” (“Booking Processor”)
     + проверят доступность выбранных мест при бронировании
     + оркестрирует взаимодействии с другими сервисами при обработке соответствующих этапов процесса бронирования
     + актуализирует текущий статус бронирования при обработке соответствующих этапов процесса бронирования
     + осуществляет оплату бронирования через взаимодействие с сервисом “Оплата”
     + осуществляет отправку e-mail нотификации о бронировании через взаимодействие с сервисом “Отправка E-mail нотификаций”
3. **Сервис «Оплата» (“Payment Service”)**
   * Выполняет платёж на указанную сумму через внешний платёжный шлюз (используя указанное назначение платежа и данные банковской карты пользователя)
4. **Сервис «Отправка E-mail нотификаций» (“Mailing Service”)**
   * Выполнят отправку e-mail нотификаций на указанный адрес электронной почты, используя указанные шаблон и содержимое письма

**Схема взаимодействия сервисов**

****

**Контракты сервисов**

**Сервис «Афиша Цирка» (“Circus Timetable”)**

Запросы

* Поиск представлений: GET /api/v1/shows/search/{query}
* Список представлений на указанную дату: GET /api/v1/shows/daily/{date}
* Информация о представлении: GET /api/v1/shows/{id}
* Расписание сеансов представления: GET /api/v1/shows/timetable/{id}/{date}

Команды: -

События: -

Зависимости: -

**Сервис “Бронирование Сеансов” (“Session Booking”)**

Запросы

* Список и стоимость доступных мест на сеанс:  
  GET /api/v1/seats/{showId}/{date}/{session}
* Информация о бронировании:  
  GET /api/v1/booking/{id}

Команды:

* Забронировать места на сеанс:  
  POST /api/v1/booking/{showId}/{date}/{session}  
  Body:  
   {  
   “seats” : [  
   { “row” : 1, “number” : 10 },  
   { “row” : 1, “number” : 11 }  
   ],  
   “price” : { “amount” : 100, “currency”: “RUB” },  
   “contact” : {  
   “firstName”: “…”,  
   “lastName”: “…”,  
   “email” : “…”  
   },

“card” : { … }  
 }

* BookingProcessor.MakePayment
* BookingProcessor.SendEmail

События:

* BookingService.SessionBookingInitiated

Зависимости:

* При формировании команды на оплату и отправки e-mail нотификации о подтверждении бронирования запрашивает подробную информацию о представлении у Сервиса «Афиша Цирка» (СircusTimetableService.GetShow(id)).
* BookingService.SessionBookingInitiated
* Событие PaymentService.PaymentSucceeded
* Событие PaymentService.PaymentFailed
* Событие MailingService.EmailSent

**Сервис “Оплата” (“Payment Service”)**

Запросы: -

Команды: -

События:

* PaymentService.PymentSucceeded
* PaymentService.PaymentFailed

Зависимости:

* Команда BookingProcessor.MakePayment

**Сервис «Отправка E-mail нотификаций» (“Mailing Service”)**

Запросы: -

Команды: -

События:

* MailingService.EmailSent

Зависимости:

* Команда BookingProcessor.SendEmail