**UC02. Сформировать расписание мастера**

**Контекст использования:** Менеджер создает расписание работы мастера в виде слотов для выбора клиентами.

**Область действия:** Барбершоп, оказание услуг клиентам.

**Уровень:** Цель пользователя.

**Основное действующее лицо:** Менеджер.

**Предусловия:**

- Менеджер получил график работы мастеров.

- Менеджер зашел на сайт.

**Основной сценарий:**

***Шаг 1. Менеджер открывает страницу ввода слотов мастеров на обслуживание***

**Приложение:** Веб-интерфейс (Клиент)

**Роли:** Менеджер

**Описание:** *Менеджер заходит в веб-интерфейс, который предоставляет форму для ввода новых слотов для мастеров. На странице отображается календарь с доступными слотами для мастеров.*

**Интеграция:** *Взаимодействие с сервером для загрузки текущих данных о мастерах и их доступных слотах.*

**Интеграционный метод:**

**HTTP GET** запрос на сервер для получения информации о текущем расписании мастеров.

**Пример:** GET <https://www.barbershop_example.com/api/masters/slots>

**Параметры:** date: дата для фильтрации доступных слотов (опционально)

**Ответ:** Сервер возвращает список мастеров и их текущие доступные слоты.

**Пример:**

Код: 200 OK

Тело ответа:

{

"masters": [

{"id": 1, "name": "Иван Иванов", "slots": [{"start": "2025-06-14T09:00", "end": "2025-06-14T10:00"}]},

{"id": 2, "name": "Алексей Сидоров", "slots": [{"start": "2025-06-14T10:00", "end": "2025-06-14T11:00"}]}

]

}

**Шаг 2. *Менеджер выбирает дату и слот (период времени длительностью 1 час), выбирает мастера из сущности «Сотрудники», которым указана роль «Мастер» в сущности «Роль сотрудника»***

**Приложение:** Веб-интерфейс (Клиент)

**Роли:** Менеджер

**Описание:** *Менеджер выбирает дату и время для нового слота (время для слота фиксировано на 1 час), а также назначает мастера, которому будет закреплен этот слот. В интерфейсе отображаются только те сотрудники, у которых есть роль «Мастер».*

**Интеграция:** Запрос информации о сотрудниках, имеющих роль «Мастер».

**Интеграционный метод:**

**HTTP GET** запрос к серверу для получения списка сотрудников с ролью «Мастер» из базы данных.

**Пример:**

GET <https://www.barbershop_example.com/api/employees?role=master>

**Параметры (см после знака ?):** role: роль сотрудника (в данном случае «Мастер»)

**Ответ:** Сервер возвращает список сотрудников, которым назначена роль «Мастер».

**Пример:**

**Код: 200 OK**

**Тело ответа:**

**{**

**"masters": [**

**{"id": 1, "name": "Иван Иванов"},**

**{"id": 2, "name": "Алексей Сидоров"}**

**]**

**}**

***Шаг 3. Система подтверждает, что выбранный слот не забронирован в системе клиентом***

**Приложение:** Веб-интерфейс (Клиент) и сервер

**Роли:** Система, Менеджер

**Описание:** *после того как менеджер выбрал слот и мастера, система проверяет, не занят ли данный слот клиентом. Если слот уже забронирован, система информирует менеджера об этом.*

**Интеграция:** Взаимодействие с системой бронирования для проверки статуса выбранного слота.

**Интеграционный метод:**

**HTTP GET** запрос к серверу с параметрами: дата, время и мастер.

**Пример:** GET <https://www.barbershop_example.com/api/slots/check>

**Параметры:**

date: дата выбранного слота

time: время выбранного слота

master\_id: ID выбранного мастера

**Ответ:** Сервер возвращает статус слота (занят или свободен).

**Пример:**

**Ответ:**

Код: 200 OK

Тело ответа:

{

"slot\_status": "available"

}

***Шаг 4. Менеджер подтверждает закрепление времени за мастером***

***Приложение: Веб-интерфейс (Клиент)***

**Роли:** Менеджер

**Описание:** *если слот не забронирован, менеджер подтверждает закрепление времени за выбранным мастером. После этого система фиксирует это изменение.*

**Интеграция:** *Взаимодействие с базой данных для записи информации о закрепленном слоте.*

**Интеграционный метод:**

**HTTP POST** запрос на сервер для добавления нового слота в базу данных.

**Пример:** POST <https://www.barbershop_example.com/api/slots/book>

Тело запроса:

{

"master\_id": 1,

"slot\_time": "2025-06-14T09:00",

"client\_id": null

}

**Ответ:** Сервер возвращает подтверждение успешного добавления слота.

**Пример:**

Код: 201 Created

Тело ответа:

{

"status": "success",

"message": "Slot successfully booked."

}

***Шаг 5. Система закрепляет слот: дата, время, мастер***

**Приложение:** Сервер

**Роли:** Система

**Описание:** *Система сохраняет выбранный слот за мастером в базе данных, обновляя информацию о доступных слотах.*

**Интеграция:** Запись в базу данных с обновлением информации о свободных слотах.

**Интеграционный метод:**

**SQL запрос (в рамках HTTP POST метода из шага 4)** для обновления состояния слота в базе данных.

**Пример: UPDATE slots SET status = 'booked', master\_id = :master\_id WHERE slot\_time = :slot\_time;**

**Ответ:** Подтверждение успешной записи в базу данных.

**Пример:**

Код: 200 OK

Тело ответа:

{

"status": "success",

"message": "Slot successfully updated."

}

**Альтернативный сценарий:**

***Шаг 1. Система сообщает, что выбранный слот забронирован в системе клиентом***

**Приложение:** Веб-интерфейс (Клиент) и сервер

**Роли:** Система, Менеджер

**Описание:** *если слот выбран, но уже забронирован другим клиентом, система уведомляет менеджера о том, что слот занят.*

**Интеграция:** Проверка статуса слота в базе данных.

**Интеграционный метод:**

**HTTP GET** запрос для проверки состояния слота в базе данных.

**Пример:** GET <https://www.barbershop_example.com/api/slots/check>

**Параметры:**

date: дата выбранного слота

time: время выбранного слота

master\_id: ID выбранного мастера

**Ответ:** Сервер возвращает сообщение о том, что слот занят.

Ответ:

Код: 409 Conflict

Тело ответа:

{

"slot\_status": "booked",

"message": "This slot has already been booked."

}

***Шаг 2. Менеджер связывается по телефону с клиентом и договаривается о переносе визита на другое время или к другому мастеру, вносит информацию в систему***

**Приложение:** Веб-интерфейс (Курсив)

**Роли:** Менеджер, Клиент

**Описание:** *после получения уведомления о занятости слота менеджер связывается с клиентом по телефону, чтобы перенести визит. После согласования с клиентом менеджер вносит изменения в систему.*

**Интеграция:** Обновление информации о бронировании.

**Интеграционный метод:**

**HTTP POST** запрос для внесения изменений в данные о бронировании.

**Пример:** POST <https://www.barbershop_example.com/api/bookings/update>

Тело запроса:

{

"booking\_id": 123,

"new\_slot\_time": "2025-06-14T12:00",

"new\_master\_id": 2

}

**Ответ:** Сервер подтверждает обновление данных о бронировании.

**Пример:**

Код: 200 OK

Тело ответа:

{

"status": "success",

"message": "Booking successfully updated."

}

***Шаг 3. Система направляет СМС по телефону клиента об изменении времени и/или мастера визита***

**Приложение:** Сервис SMS-оповещений

**Роли:** Система, Клиент

**Описание:** *Система отправляет клиенту СМС-сообщение о перенесенном визите, изменении времени или мастера.*

**Интеграция:** Взаимодействие с внешним сервисом SMS.

**Интеграционный метод:**

**HTTP POST** запрос к SMS-сервису для отправки сообщения.

**Пример:** POST <https://www.barbershop_example.com/api/sms/send>

Тело запроса:

{

"phone\_number": "+1234567890",

"message": "Your appointment has been rescheduled to 2025-06-14 at 12:00 with master Алексей Сидоров."

}

**Ответ:** Сервер подтверждает успешную отправку СМС-сообщения.

**Пример:**

Код: 200 OK

Тело ответа:

json

Копировать

Редактировать

{

"status": "success",

"message": "SMS successfully sent."

}