Exercise 02 - актуализация UC

UC04. Регистрация клиента (менеджером)

Цель

Зарегистрировать клиента по звонку или при личном обращении, чтобы добавить его в систему, создать учетную запись и отправить уведомление с данными для входа.

Область действия

Барбершоп, оказание услуг клиентам.

Уровень

Цель пользователя.

Роли

- **Менеджер** - выполняет регистрацию через веб-интерфейс.
- **Клиент** - предоставляет минимальные данные (имя и телефон).

Сущности

- **Person** - базовая сущность (логин, пароль (хэш), имя, телефон).
- **Client** - связь персоны с клиентом (дата регистрации).
- **ClientNotification** - уведомление клиенту о регистрации.

Предусловия

- Менеджер авторизован в **веб-клиенте (front-end)**.
- Доступен **сервер приложений (API)**.
- Подключен **сервис уведомлений** (SMS / Telegram / WhatsApp / VK).
- Актуальны справочники **Channels** и **NotificationTemplates**.

Постусловия

- Созданы записи **Person** и **Client**.
- Запланировано уведомление (**ClientNotification**) с временным паролем.
- В журнал аудита записано событие регистрации.

Основной сценарий

№	Шаг	Приложение	Роль	Интеграция (метод/шина)	Вход / Выход
1	Менеджер открывает форму «Новый клиент»	Клиент (Web UI)	Менеджер	-	-

№	Шаг	Приложение	Роль	Интеграция (метод/шина)	Вход / Выход
2	Вводит **телефон** и **имя** клиента	Клиент (Web UI)	Менеджер	-	Ввод: phone, first_name
3	Проверка уникальности телефона	Сервер (API)	Система	**GET /api/persons?phone=**	exists=true/false
4	При отсутствии телефона -генерация **login** и **временного пароля**, хэширование	Сервер (API)	Система	-	login, password_hash, temp_password
5	Создать **Person**	Сервер (API)	Система	**POST /api/persons**	person_id
6	Создать **Client**, связав с **Person**	Сервер (API)	Система	**POST /api/clients**	client_id
7	Записать уведомление о регистрации	Сервер (API)	Система	**POST /api/notifications**	notification_id
8	Отправить уведомление клиенту	Сервис уведомлений	Система	**sendSMS()** / **sendTelegram()**	msg_id, status
9	Отобразить результат менеджеру	Клиент (Web UI)	Система	-	client_id, результат регистрации

Альтернативные сценарии

A1. Телефон уже зарегистрирован

- Система сообщает: «Клиент уже существует».
- Менеджер может:
  - Перейти в карточку клиента,**A1.1**,
  - Или обновить данные существующей записи,**A1.2**.
- Интеграция: **GET /api/clients?phone=**.

A2. Некорректный телефон

- Проверка формата E.164 не проходит.
- Менеджер получает сообщение: «Проверьте номер».
- Регистрация не выполняется.

A3. Ошибка создания записи

- При сбое на сервере (**DB error**, **network timeout**) - откат транзакции.

- Сообщение менеджеру: «Временная ошибка, попробуйте снова».
- Лог: **ERROR_REGISTER_CLIENT**.

A4. Ошибка отправки уведомления

- Запись в **ClientNotification** есть, но канал возвращает **failed**.
- Система повторяет отправку 3 раза (ретрай).
- При неудаче - уведомление менеджеру, предложить другой канал.

A5. Согласие клиента

- Менеджер фиксирует согласие клиента на обработку данных.
- Если согласие не дано - отправляются только сервисные уведомления.

A6. Повторная регистрация (идемпотентность)

- Повторная форма с тем же телефоном в течение 5 минут -> ответ**idempotent OK**, возвращаются существующие **person_id** и **client_id**.

A7. Канал уведомлений недоступен

- Канал (**sms**, **telegram**, …) помечен **is_active=false**.
- Менеджеру предлагается выбрать другой канал или отложить уведомление.

Бизнес-правила

- **phone** - обязательное уникальное поле (формат E.164).
- **login** - по умолчанию совпадает с телефоном.
- **password** - хранится в виде хэша.
- Регистрация - атомарная (создание **Person** и **Client** в одной транзакции).
- Уведомления - асинхронные, отправляются через очередь.

Справочники

Справочник	Назначение
Channels	Каналы уведомлений (SMS, Telegram, WhatsApp, VK).
NotificationTemplates	Шаблоны сообщений о регистрации, напоминания и др.
NotificationStatus	Возможные статусы уведомлений (**scheduled**, **sent**, **delivered**, **failed**).
ConsentTypes	Типы согласий (по желанию, если включено в политику).

Интеграционные методы

- **POST /api/persons/check** - проверка телефона. **POST /api/persons** - создание персоны.
- **POST /api/clients** - создание клиента.
- **POST /api/notifications** - создание уведомления. **notifications.send** - очередь/шина для рассылки.
- **POST /api/notifications/{id}/delivery-callback** - обновление статуса доставки.

Передача справочных данных

Вариант 1 - справочники актуализируютсядо UC

Предусловие:Справочники **Channels** и **NotificationTemplates** синхронизированы заранее (например,

раз в 15 минут).

- UI отображает только доступные каналы.
- При отсутствии шаблона -> срабатывает A7 (предложить альтернативу).

Вариант 2 - справочник передаётсяв сообщении

- Менеджер передаёт **channel='sms'**, **template_code='REG_WELCOME'**.
- Сервер валидирует запись в реальном времени.

Вариант 3 - обновление справочникапо необходимости

- При отсутствии записи система создаёт дефолтный шаблон.
- Используется только при низком риске рассинхронизации.

Нефункциональные требования

-**Идемпотентность:** повторная регистрация по одному телефону не создаёт дубликатов.
-**Аудит:** логирование шагов 3–9 и статусов уведомлений.
-**Безопасность:** пароль только в хэше; телефон маскируется в логах.
-**Производительность:** индекс по **person.phone_number**.
-**Надёжность:** повторная отправка уведомлений при сбое.

Краткое резюме для разработки

- Реализовать REST-методы: **check**, **create person**, **create client**, **create notification**.
- Форма регистрации с валидацией телефона и выбором канала.
- Очередь уведомлений и колбэки доставки.
- Регулярная синхронизация справочников (**Channels**, **Templates**).