

Разработка приложения: “Древо Рода”

1.MVP

[1.1 Существующие проблемы:](#)

[1.2 Цель разработки:](#)

[1.3 Use Cases](#)

[1.4 User Stories](#)

[1.5 Описание](#)

[1.6 Структура базы данных:](#)

[1.7 Процесс “Создание персонажа”](#)

1.MVP

1.1 Существующие проблемы:

- Сложная визуализация генеалогического древа, рисунок/excel
- Нет полного понимания о генетических рисках по здоровью себя (пользователя), последующих поколений

1.2 Цель разработки:

- Упростить жизнь пользователя, позволить автоматизировать процесс визуализации.
- Сформировать пул генетических рисков для себя и для потомков.

1.3 Use Cases

Основной актер: Пользователь (член семьи, генеалог).

1. Создание персонажа

- **Предусловие:** Пользователь открыл приложение.
- **Основной поток:**
 1. Пользователь нажимает "Добавить человека".
 2. Система отображает форму с полями: имя, дата рождения, дата смерти, возраст (если дата неизвестна).

3. Пользователь заполняет обязательное поле "Имя".
 4. Пользователь добавляет заболевания (опционально).
 5. Пользователь нажимает "Сохранить".
 6. Система создает персонажа и автоматически генерирует связи с "?" для родителей/детей.
- **Постусловие:** Персонаж отображается в древе.
-

2. Добавление связи между персонажами

- **Предусловие:** В древе есть минимум один персонаж.
 - **Основной поток:**
 1. Пользователь выбирает персонажа.
 2. Нажимает "+" рядом с узлом.
 3. Выбирает тип связи: "родитель", "ребенок", "супруг", "брат/сестра".
 4. Если связь с существующим человеком: выбирает из списка.
 5. Если связь неизвестна: оставляет "?".
 6. Система добавляет связь и обновляет древо.
 - **Альтернативный поток:** Если выбранный человек уже связан, система показывает ошибку.
-

3. Управление заболеваниями

- **Предусловие:** Пользователь редактирует существующего персонажа.
 - **Основной поток:**
 1. Пользователь нажимает "Добавить заболевание".
 2. Вводит название заболевания (или выбирает из списка).
 3. Система сохраняет заболевание и отображает иконку 🩺 на узле.
 - **Постусловие:** Заболевание отображается в профиле персонажа.
-

4. Визуализация древа

- **Предусловие:** В системе есть минимум один персонаж.
 - **Основной поток:**
 1. Пользователь открывает главный экран.
 2. Система отображает древо в стиле Иггдрасиль (предки — внизу, потомки — вверху).
 3. Пользователь может масштабировать и перемещаться по дереву.
 4. При нажатии на узел отображается детальная информация.
-

1.4 User Stories

1. Как пользователь, я хочу добавлять новых людей в дерево, чтобы сохранять историю семьи.

- **Критерии приемки:**
 - Обязательное поле — имя.
 - Автоматическое создание связей с "?" для родителей/детей.

2. Как пользователь, я хочу указывать родственные связи, чтобы отображать структуру семьи.

- **Критерии приемки:**
 - Поддержка связей: родитель-ребенок, супруги, братья/сестры.
 - Возможность отметить связь как "неизвестную".

3. Как пользователь, я хочу отмечать наследственные заболевания, чтобы отслеживать медицинскую историю.

- **Критерии приемки:**
 - Иконка заболевания отображается на узле.
 - Заболевания можно добавлять из списка или создавать новые.

1.5 Описание

Базовая структура

- **Центральный узел** (ствол): текущий фокусный человек (изначально пустой квадрат с "?").
- **Корни** (вниз): родители, дедушки/бабушки.
- **Ветви** (вверх): дети, внуки.
- **Горизонтальные ветви** (по бокам): супруги, братья/сестры.

Пример интерфейса:

Клик на узел → меню с кнопками:

- "Добавить родителя",
- "Добавить ребенка",
- "Добавить супруга",
- "Удалить ветку".

Динамическое добавление узлов

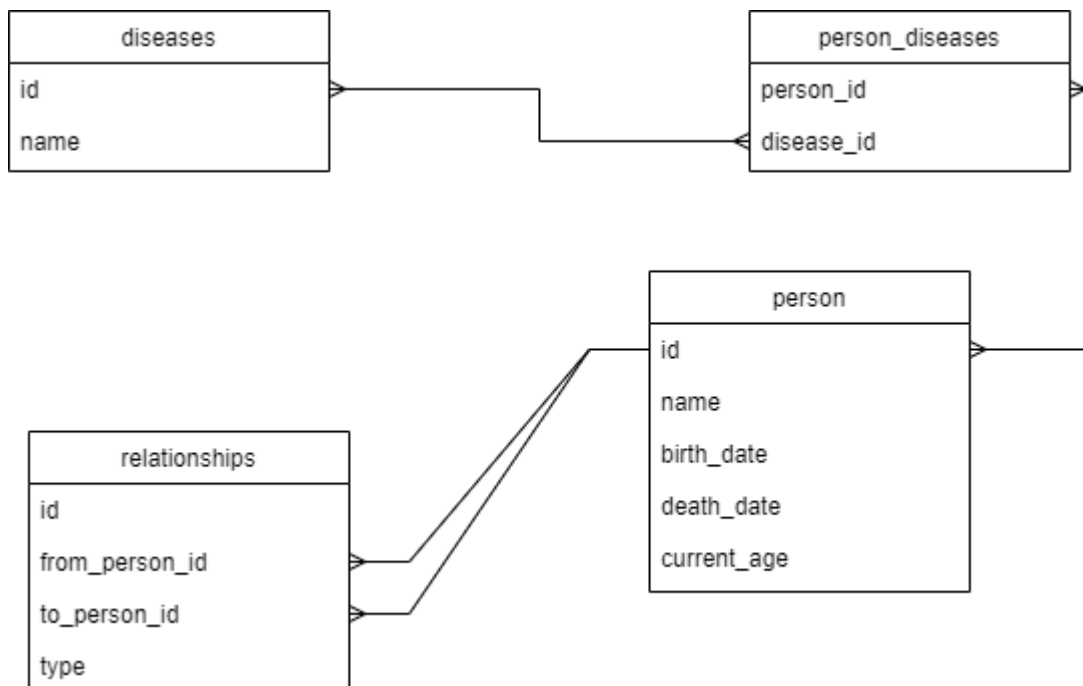
- **Правила:**
 - При заполнении данных в узле автоматически появляются:
 - **2 узла снизу** (родители) с "?".

- **1 узел сверху** (дети) с "?". При заполнении ребенка автоматически добавляется еще один узел с "?" от тех же родителей.
- Горизонтальные связи (супруги) добавляются **вручную** через кнопку "+" на узле.

Элементы управления

1. **Правый клик на узле** → контекстное меню:
 - "Редактировать",
 - "Добавить родителя",
 - "Добавить ребенка",
 - "Добавить супруга",
 - "Свернуть ветку",
 - "Удалить".
2. **Панель инструментов:**
 - Кнопки: "Экспорт в JSON", "Импорт из JSON".
3. **Форма редактирования узла:**
 - Обязательные поля: Имя.
 - Опциональные: Дата рождения, Дата смерти, Заболевания.

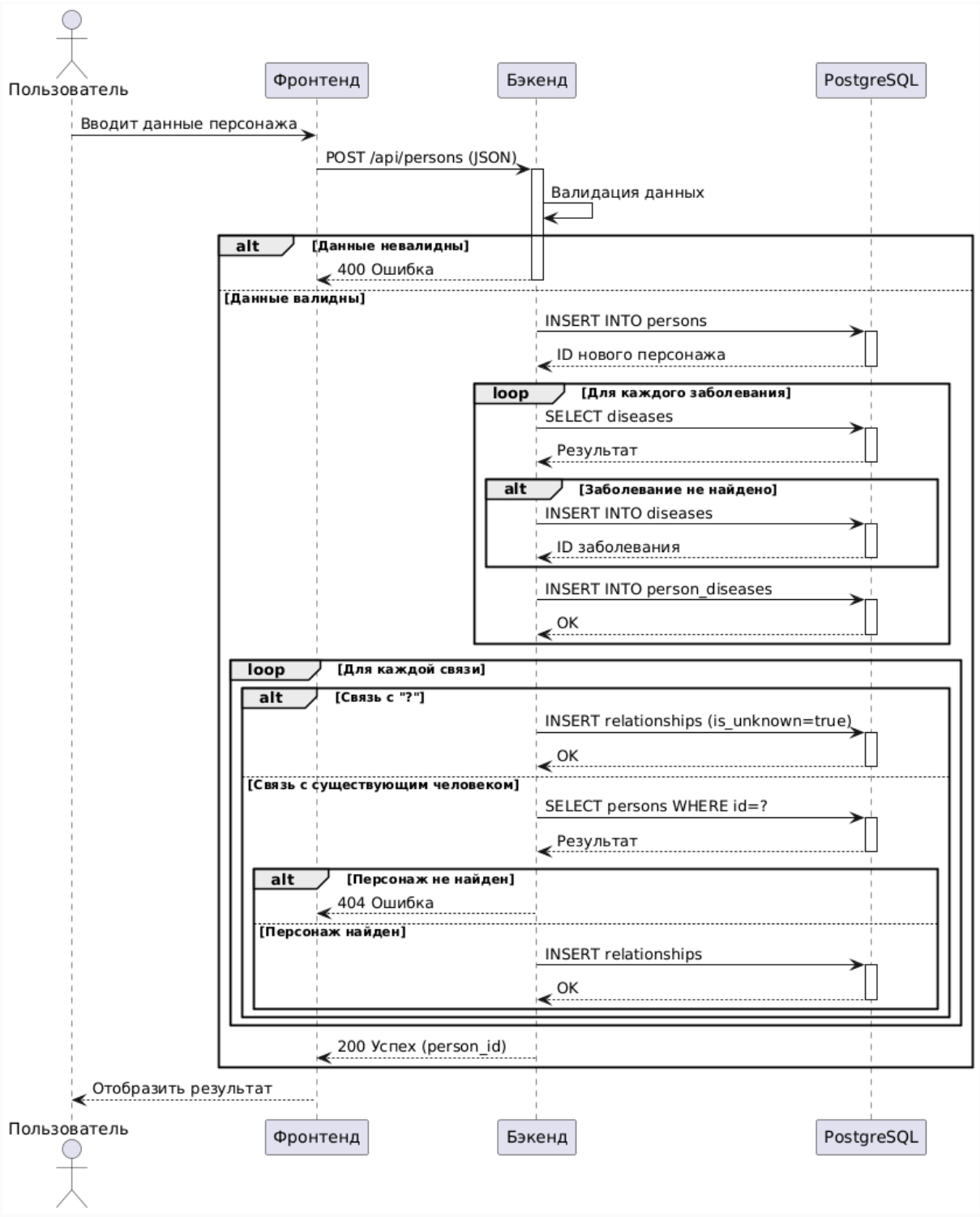
1.6 Структура базы данных:



Поле	Таблица	Тип	Пример	Комментарий	Ограничения
id	person	serial	1		primary key
name	person	varchar	Иван Иванов		обязательное
birth_date	person	timestamp	1990-01-01 00:00:00	дата и время рождения	необязательное, по умолчанию проставляется время 00:00 по utc+3
death_date	person	timestamp	NULL	дата и время рождения	необязательное, по умолчанию проставляется время 00:00 по utc+3
current_age	person	int	35	возраст человека	необязательное, вычисляемое, >=0
id	relationships	serial	1		primary key
from_person_id	relationships	int	1	ID человека, от которого исходит связь,	обязательное
to_person_id	relationships		2	ID человека, к которому направлена связь	обязательное
type	relationships		parent	Тип связи: parent, marriage, child	обязательное
id	diseases	serial	1		primary key
name	diseases		Диабет	название болезни	обязательное, уникальное
person_id	person_diseases	int	1	id человека из таблицы person	обязательное
disease_id	person_diseases	int	1	id	обязательное

	diseases			заболевания из таблицы diseases	
--	----------	--	--	---------------------------------------	--

1.7 Процесс “Создание персонажа”



JSON отправка данных с фронта на бэк:

```
{
  "person": {
    "name": "Иван Петров",
    "birth_date": "1980-05-15T00:00:00Z", // Дата в формате ISO
    "death_date": null,                // null, если человек жив
    "manual_age": null                // null, так как birth_date указана
  },
  "diseases": ["Диабет", "Гипертония"], // Названия заболеваний
  "relationships": [
    {
      "type": "spouse",                // Тип связи: супруг(а)
      "to_person_id": 2,               // ID существующего человека
      "is_unknown": false              // Связь известна
    },
    {
      "type": "parent",                // Тип связи: родитель
      "is_unknown": true               // Родитель неизвестен (?)
    },
    {
      "type": "sibling",               // Тип связи: брат/сестра
      "to_person_id": 3,               // ID существующего человека
      "is_unknown": false
    }
  ]
}
```

JSON ответ с бэка на фронт:

```
{
  "status": "success",
  "person_id": 100,                  // ID нового человека
  "created_diseases": [5, 6],        // ID новых заболеваний (если они
  создавались)
  "relationships_added": 3           // Количество добавленных связей
}
```