Содержание

Вариант 1

## ….User story: Запись на пробный урок в онлайн-школе программирования

Вариант 2

## ... Use Case: Запись на пробный урок в онлайн-школе программирования

**Краткое описание:**

Пользователь сайта онлайн-школы программирования записывается на пробный урок, заполняя форму с именем и телефоном. После успешной отправки кнопка меняет состояние на "Заявка отправлена".

**Акторы:**

Основной актор: Пользователь

Второстепенные акторы: Система, почтовый сервис

**Предусловия:**

Пользователь зашел на сайт vecus.ru.

**Ожидаемый результат:**

Пользователь записан на пробный урок.

**Основной поток событий (Basic Flow):**

1.Пользователь нажимает на кнопку "Записаться"

2.Система отображает форму "Записаться на пробный урок с преподавателем".

3.Пользователь вводит Имя и Телефон.

4.Пользователь отправляет данные формы.

5.Система подтверждает правильность введенных данных.

6.Система передает данные в почтовый сервис для уведомления администрации школы.

7. Система информирует пользователя об успешной операции

### **Альтернативный поток (Alternative Flow):**

1.Пользователь нажимает кнопку "Записаться" в разделе "Курсы". Дальнейшие шаги аналогичны основному сценарию.

**Исключения (Exceptions):**

**3а. Пользователь оставляет поле "Имя" или "Телефон" пустым.**

**3а.1** Система информирует пользователя "Пожалуйста, заполните все поля".

**5а. Введены некорректные данные**

**5а.1.** Система убеждается, что введённый номер телефона не соответствует шаблону из последовательности цифр от 6 до 11 цифр, содержит буквы или другие символы  
 **5а.2.** Система информирует пользователя о некорректности введенных данных

**6a. Ошибка при передаче данных в почтовый сервис:**

**6.a1** Система не может передать данные в почтовый сервис.

**6.a2** Пользователь получает сообщение: "Ошибка при отправке данных, попробуйте позже".

**6b.Задержка ответа от системы**

**6b1**.Система долго обрабатывает запрос.

**6b.2** Пользователь видит индикатор загрузки и по истечении тайм-аута получает сообщение: "Время ожидания истекло, попробуйте позже".

**7.a Отсутствие подтверждения успешной отправки:**

**7a.1** Данные отправлены, но пользователь не получает уведомление об успешной операции.

**7а.2** Система повторно отправляет данные или отображает соответствующее уведомление.

**Постусловия:**

После успешной отправки кнопка меняет вид на "Заявка отправлена".

Данные о заявке успешно отправлены на почту администрации.

## ….Use Case: Покупка курса в онлайн-школе программирования

### **Название:**

Покупка курса

### **Краткое описание:**

Пользователь выбирает нужный ему курс и совершает оплату. После успешной транзакции он получает доступ к урокам.

### **Акторы:**

**Основной актор:** Пользователь

**Второстепенные акторы:** Система, платежный сервис

### **Предусловия:**

1. Пользователь зашел на сайт **vecus.ru**.
2. Пользователь авторизован в системе.
3. Я бы добавил сюда третьим предусловие "Пользователь выбрал курс для покупки" *(и это предусловие раскрыл в отдельном сценарии)*. А в текущем варианте использования оставил бы лишь именно ПОКУПКУ курса.

### **Основной поток событий (Basic Flow):**

1. Пользователь выбирает нужный ему курс в разделе "Курсы"
2. Пользователь нажимает на кнопку "Купить курс".
3. Система отображает страницу оплаты с доступными способами платежа.
4. Пользователь выбирает способ оплаты
5. Пользователь вводит платежные данные и подтверждает оплату.
6. Система отправляет запрос в платежный сервис.
7. Платежный сервис проверяет данные и выполняет транзакцию.
8. Система получает подтверждение успешного платежа.
9. Система отправляет пользователю уведомление об успешной покупке.
10. Пользователь получает доступ к курсу в личном кабинете.

### **Альтернативный поток (Alternative Flow):**

1. Пользователь выбирает другой курс в каталоге.
2. Дальнейшие шаги аналогичны основному сценарию.

### **Исключения (Exceptions):**

**2.a Неудачная авторизация пользователя:**

**2.a** Пользователь не вошел в аккаунт перед оплатой.

**2.b** Система перенаправляет пользователя на страницу авторизации перед оплатой.

**5.a Ошибка ввода платежных данных:**

**5.a.1** Пользователь вводит некорректные данные карты (неверный номер, срок действия и т. д.).

**5.a.2** Система отправляет сообщение: "Некорректные данные карты. Проверьте введенные данные и попробуйте снова."

**5.b Проблемы с интернет-соединением:**

**5.b.1** Пользователь теряет интернет-соединение в процессе оплаты.

**5.b.2** Система выводит сообщение: "Ошибка соединения. Проверьте подключение к интернету и попробуйте снова."

**7.a Недостаточно средств на карте:**

**7.a.1** Платежный сервис отправляет ответ об отсутствии средств.

**7.a.2** Пользователь получает сообщение: "Недостаточно средств для оплаты. Пополните счет или используйте другую карту."

**6.a Технический сбой платежного сервиса:**

**6.a.1** Платежный сервис временно недоступен.

**6.a.2** Система выводит сообщение: "Сбой в работе платежного сервиса. Попробуйте позже."

**8.a Ошибка оплаты:**

**8.a.1** Платежный сервис отправляет ошибку транзакции.

**8.a.2** Система информирует пользователя: "Оплата не прошла. Попробуйте еще раз или используйте другой метод оплаты."

**8.b Отклонение платежа банком:**

**8.b.1** Система получает отказ от банка.

**8.b.2** Пользователь получает сообщение: "Ваш банк отклонил платеж. Свяжитесь с банком или используйте другой способ оплаты."

Какой писать пункт, возможно всегда! **Сбой системы:**

В случае технических проблем система отправляет пользователю сообщение: "Возникла техническая ошибка. Попробуйте позже."

### 

### **Постусловия:**

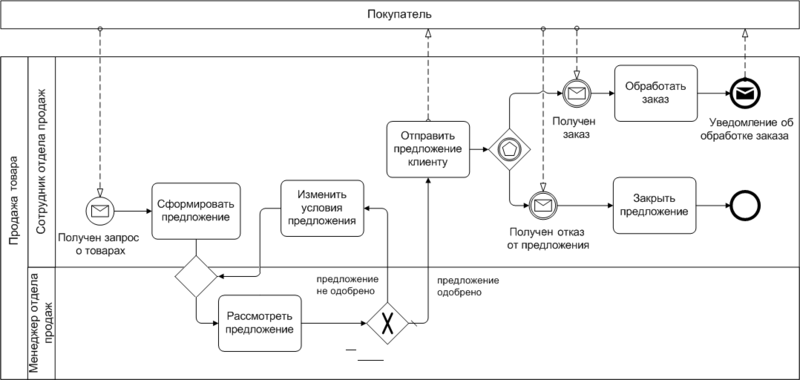
**1.** Пользователь успешно оплатил курс и получил доступ к урокам.

**1.**Система отправила уведомление на почту пользователя о подтверждении покупки.

**1.** Оплата зачислена на счет школы.

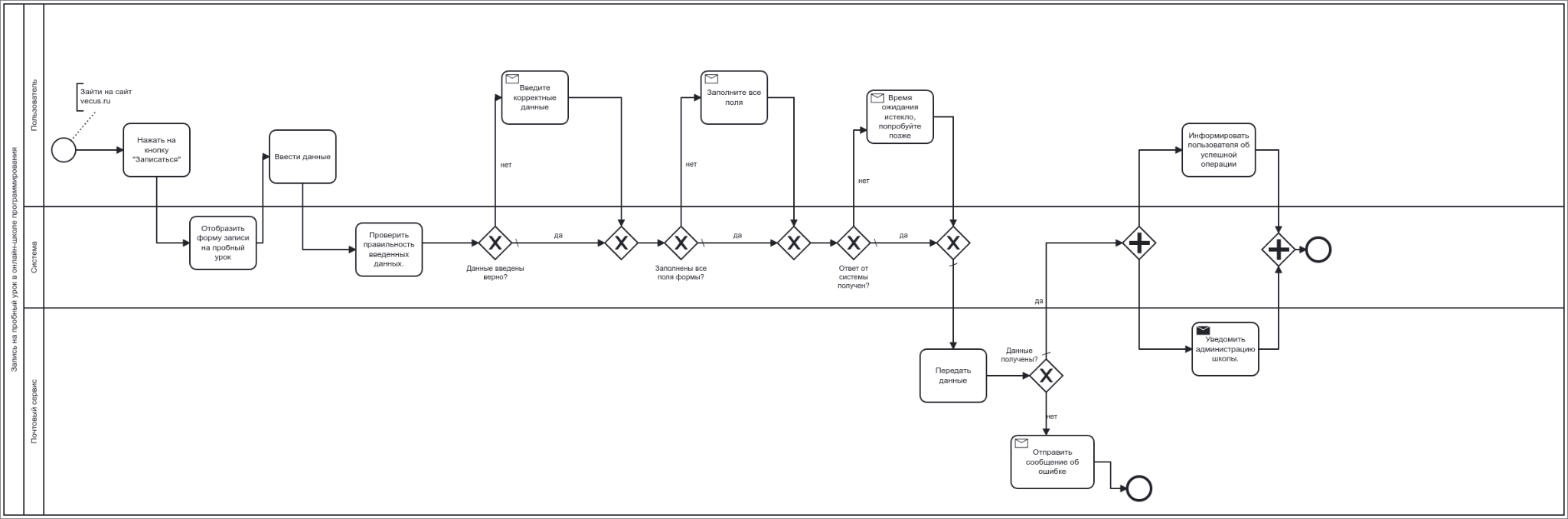
## Вопросы

1.Лучше, когда пользователь отдельно рисуется(как в примере ниже) или в одном пуле, как у меня?



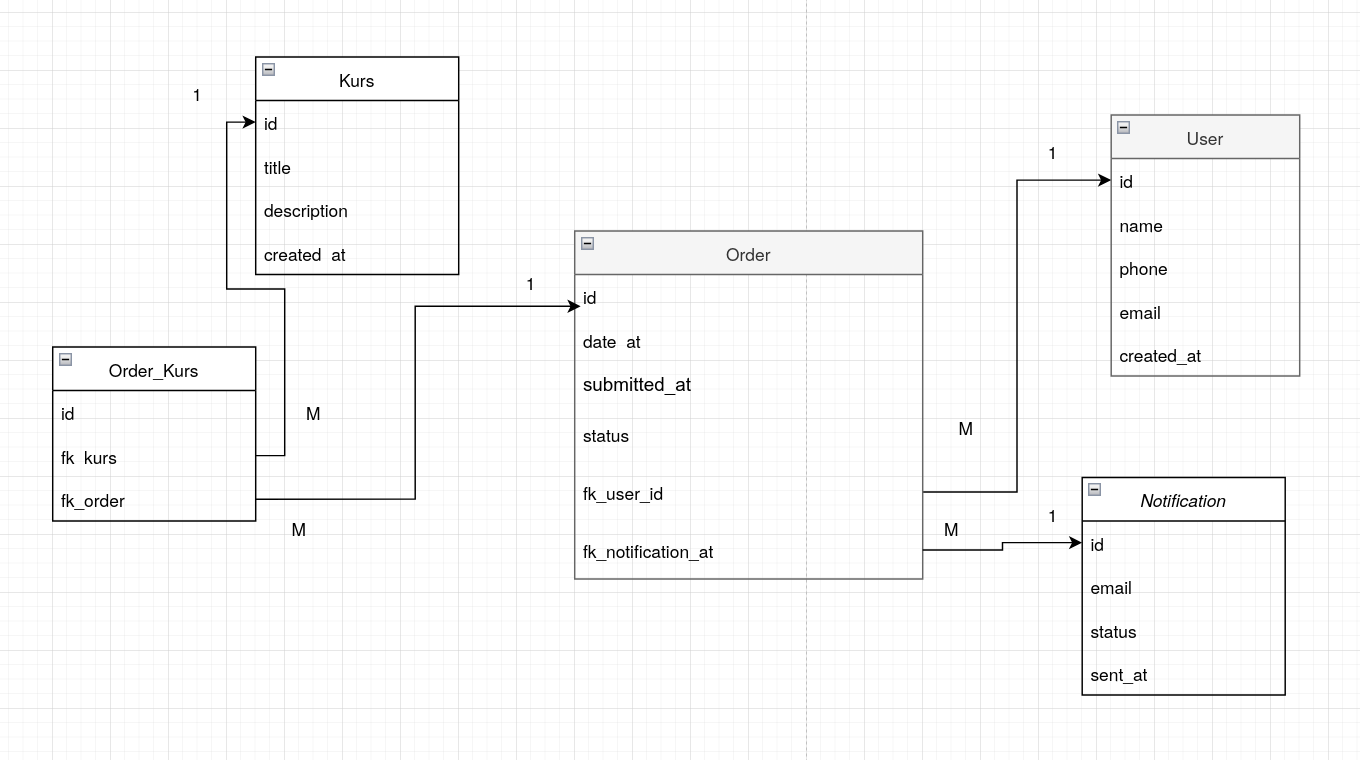
2.Альтернативный сценарий и исключения всегда надо показывать?

[Запись на пробный урок](https://stormbpmn.com/app/diagram/249474d4-774c-460a-8660-37a0896215db?overlays=e30=)



## [ER-диаграмма](https://drive.google.com/file/d/1ojtb6sMHsuajLMhGHZ2MLRPL2iNONPUL/view?usp=sharing)

Для описанного сценария можно выделить несколько сущностей и атрибутов для проектирования базы данных. Эти сущности обеспечат хранение информации о пользователях, заявках на пробные уроки и взаимодействии с системой.



---

▎Сущности и их атрибуты

▎1. Пользователь (User)

Эта сущность хранит информацию о пользователях, которые заполняют форму для записи на пробный урок.

Атрибуты:

id — Уникальный идентификатор пользователя (первичный ключ).

name — Имя пользователя.

phone — Телефон пользователя.

email - Email пользователя

created\_at — Дата и время создания записи.

---

▎2. Заявка (Order)

Эта сущность хранит данные о заявках на пробные уроки, которые отправляют пользователи.

Атрибуты:

• id — Уникальный идентификатор заявки (первичный ключ).

• fk\_user\_id — Внешний ключ, связывающий заявку с пользователем

• status — Статус заявки (например, "отправлена", "обработана").

• submitted\_at — Дата и время отправки заявки.

• notified\_at — Дата и время уведомления администрации через почтовый сервис.

---

▎3. Уведомления (Notification)

Эта сущность хранит информацию об отправке уведомлений через почтовый сервис.

Атрибуты:

• id — Уникальный идентификатор уведомления (первичный ключ).

• fk\_notification\_at — Внешний ключ, связывающий уведомление с заявкой

• email — Email, на который было отправлено уведомление.

• status — Статус отправки уведомления (например, "успешно", "ошибка").

• sent\_at — Дата и время отправки уведомления.

---

▎4. Курс (Kurs)

Атрибуты:

• id — Уникальный идентификатор курса (первичный ключ).

• title — Название курса.

• description — Описание курса.

• created\_at — Дата и время создания записи.

---

▎5. Привязка заявки к курсу Order\_Kurs)

Если заявка связана с конкретным курсом, создаётся промежуточная таблица для связи "многие ко многим" между заявками и курсами.

Атрибуты:

•id — Уникальный идентификатор записи.

• fk\_order — Внешний ключ, связывающий запись с заявкой

• fk\_kurs — Внешний ключ, связывающий запись с курсом

---

▎Связи между сущностями

1. User ↔ Order: Один пользователь может отправить несколько заявок (связь 1:N).

2. Order ↔ Notificationg: Одной заявке может соответствовать одно или несколько уведомлений (связь 1:N).

3. Order ↔ Kurs: Заявка может быть связана с одним или несколькими курсами (связь M:N через таблицу Order\_Kurs).

#### **1.Регистрация пользователя**

**Запрос:**

POST /api/users/register

Описание входных параметров

| **Атрибут** | **Тип данных** | **Обязательность** | **Где передаётся** | **Описание** |
| --- | --- | --- | --- | --- |
| name | строка | да | body | имя пользователь |
| phone | строка | да | body | телефон |
| email | строка | да | body | адрес email |
| created\_at | строка | да | body | Дата и время создания записи |
|  |  |  |  |  |

| {  "id": 1,  "name": "Иван Иванов",  "phone": "+79991234567",  "email": "ivan@example.com",  "created\_at": "2025-03-02T12:00:00Z"  } |
| --- |

**Ответ:**

Описание выходных параметров

**Статус**

**HTTP-код: 201 Created**

| {  "id": 1,  "name": "Иван Иванов",  "phone": "+79991234567",  "email": "ivan@example.com",  "created\_at": "2025-03-02T12:00:00Z"  } |
| --- |

**2. Получение списка курсов**

**Запрос:**

GET /api/kurs/list HTTP/1.1

Описание входных параметров

| **Атрибут** | **Тип данных** | **Обязательность** | **Где передаётся** | **Описание** |
| --- | --- | --- | --- | --- |
| title | строка | да | path | Название курса |
| description | строка | да | path | Описание курса |

**Ответ**

**HTTP-код: 200 OK**

Пример:

| {  “title”:”Scratch”;  “description”:”Создание игр и анимации на Scratch ”  } |
| --- |

**3.Запись на пробный урок**

**Описание: Создает запись на пробный урок для ученика.**

POST /api/order/{order\_id}/register/ HTTP/1.1

Описание входных параметров

| **Атрибут** | **Тип данных** | **Обязательность** | **Где передаётся** | **Описание** |
| --- | --- | --- | --- | --- |
| status | строка | да | body | Статус заявки |
| submitted\_at | строка | да | body | Дата и время отправки заявки. |
| fk\_user\_id | строка | да | body | Связь заявки с пользователем |
|  | строка | да | body |  |

**Ответ:**

Описание выходных параметров

**Статус**

**HTTP-код: 201 Created**

@startuml

title "Запись на пробный урок"

autonumber

actor User as u

participant Frontend as f

database PostgreSQL as p

participant Email as e

participant Admin as a

u -> f: Заполняет форму записи

alt

f -> p: post /api/user/registr

p -> e:request

par

e -> a: Поступила заявка

e -> u:Заявка отправлена

end

else

f--> u:Данные не валидны

end

