

## Техническое задание (БА+СА) аналитик

### Задача 1

Сделайте модель в нотации BPMN 2.0 по описанию рабочего потока ниже.

#### Описание:

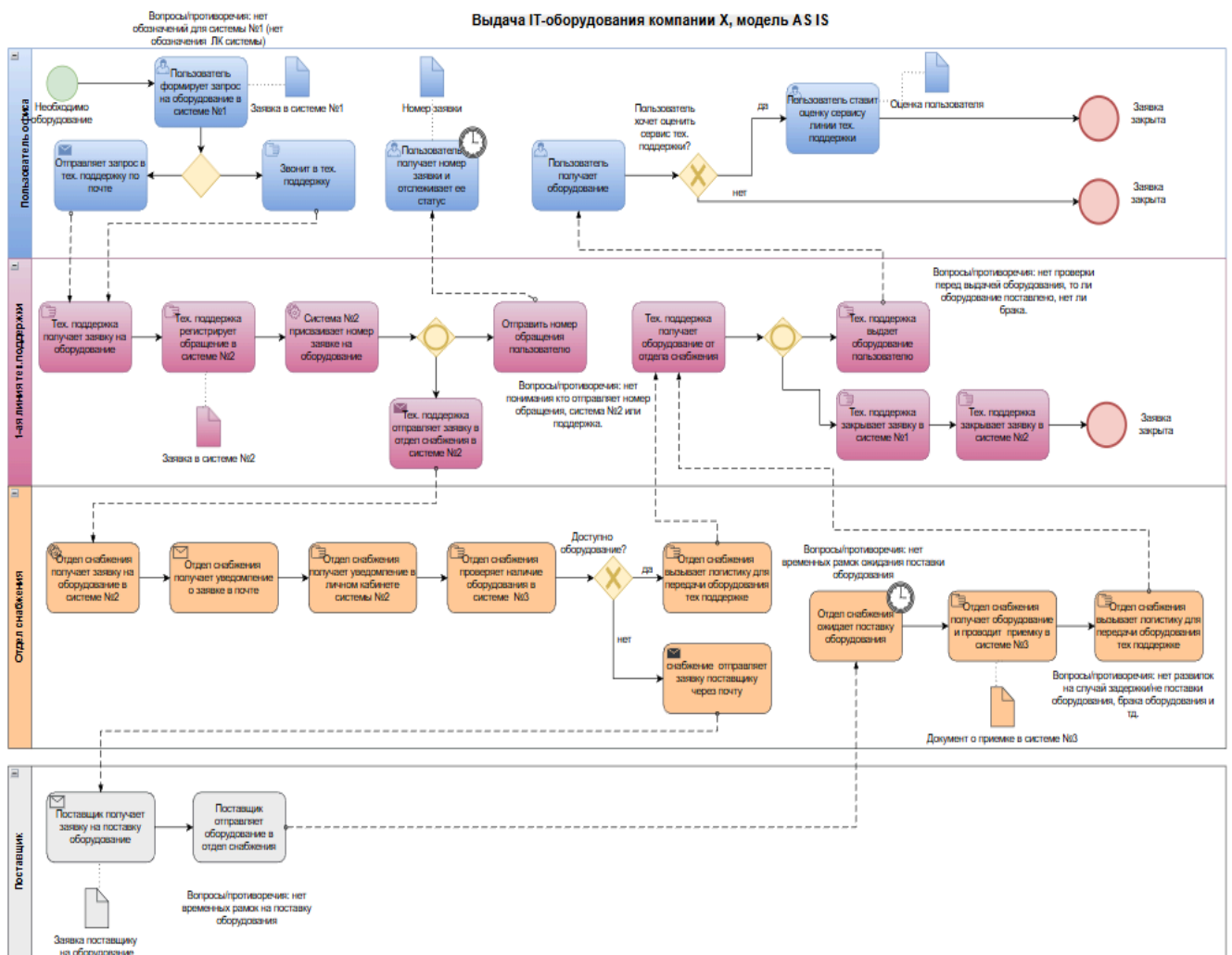
В компании «Х» работает выдача ит-оборудования пользователям по заявке, которую надо оформлять в системе №1. Пользователь офиса, когда ему нужно, делает заявку через почту или по телефону в 1-ую линию технической поддержки на оборудование. Он получает номер обращения, благодаря которому может отследить, что происходит с его заявкой. Первая линия технической поддержки делает заявку на оборудование (на основании заявки от пользователя) в системе №2 отделу снабжения. Отдел снабжения принимает заявку в системе №2. Отдел снабжения видит, что есть заявка на оборудование по уведомлению из почты и также - на экране своего личного кабинета системы №2. Далее отдел снабжения проверяет наличие запрашиваемого пользователем оборудования в системе №3.

Если оборудование есть - снабжение вызывает логистику и отправляет оборудование первой линии технической поддержки. Техническая поддержка выдаёт это оборудование пользователю и закрывает заявку пользователя в системе №1 и закрывает заявку в системе №2. Пользователь получает оборудование и может выставить оценку сервиса технической поддержке.

Если оборудования нет - снабжение делает заявку на закупку поставщику через почту. Ждёт оборудование. Поставщик его привозит. Далее снабжение проводит приём оборудования у себя в системе №3. Далее вызывает логистику и передаёт оборудование технической поддержке. Техническая поддержка выдаёт это оборудование пользователю и закрывает заявку пользователя в системе №1 и закрывает заявку в системе №2. Пользователь получает оборудование и может выставить оценку сервиса технической поддержке.

На модели можно указать вопросы, противоречия к описанию рабочего потока. (Чего не хватает для грамотного отражения модели «как есть»)

#### Решение задачи №1



## Задача 2

Представьте, что вы аналитик Личного кабинета продавца.

Вам поступила задача сформулировать требования для реализации новой фичи (функциональности), которая позволит опубликовать свой товар на маркетплейсе (витрине товаров).

Перед тобой следующие задачи:

- Сделать верхнеуровневую постановку в формате User story и описать требования к новой функциональности в формате вариантов использования (Use Case).
- Опционально: нарисовать диаграмму процесса в удобной нотации.

## Решение задачи №2

**Верхнеуровневая постановка в формате User story и описание требований к новой функциональности в формате вариантов использования (Use Case).**

### 1) Верхнеуровневая постановка в формате User story

Как продавец, я хочу иметь возможность опубликовывать свои товары на витрине маркетплейса через свой личный кабинет, чтобы после отображения товаров на маркетплейсе начать продавать товары покупателям.

Критерии приемки User story:

- Продавец имеет возможность загружать информацию и фото товара в свой ЛК.
- После публикации товара, продавец может вносить изменения в карточку товара.
- Система должна оповещать продавца о статусе публикации:
  1. На модерации
  2. Опубликован
  3. Отклонён
- При ошибке продавец получает сообщение с указанием причины.
- Все обязательные поля проверяются на корректность до отправки.

### 2) Описание требований к новой функциональности в формате вариантов использования (Use Case)

UC-01: Создание карточки товара

Акторы: Продавец

Предусловия:

- Продавец авторизован в системе
- Продавец вошел в личный кабинет (ЛК) продавца

Основной сценарий:

1. Продавец выбирает "Добавить товар" в ЛК
2. Система отображает форму создания товара
3. Продавец заполняет обязательные поля:
  - Название товара (3-100 символов)
  - Описание (10-2000 символов)
  - Категория (выбор из каталога)
  - Бренд (выбор из списка или ввод нового)
  - Артикул производителя
  - Цена товара
  - Дополнительные характеристики товара (10-2000 символов)
  - Фото товара (1-5 фото)
4. Система валидирует данные
5. Система сохраняет товар и присваивает номер
6. Продавец сохраняет товар как черновик или отправляет на модерацию

Альтернативные сценарии:

- Не заполнены обязательные поля:
  1. Система выделяет незаполненные поля красным.
  2. Выводит подсказку: "Заполните все обязательные поля".
- Ошибка загрузки фото:

1. Система показывает: "Файл не загружен, недопустимый формат. Допустимые форматы: JPG, PNG".
2. После исправлений карточка отправляется на повторную валидацию.

Постусловия:

- Продавец получает уведомление о статусе

UC-02: Публикация товара

Актеры: Продавец, Система модерации

Предусловия:

- Заполнены все обязательные поля товара
- Загружена минимум 1 фотография товара

Основной сценарий:

1. Продавец нажимает "Опубликовать товар"
2. Система выполняет автоматическую проверку:
3. Проверка цен (соответствие рекомендованной ценовой политике)
4. Проверка категории (правильность выбора)
5. Если проверка пройдена, то присваивается статус "Опубликован"
6. Система индексирует товар для поиска
7. Товар появляется в каталоге маркетплейса
8. Продавец получает уведомление об успешной публикации

Альтернативный сценарий: Требуется ручная модерация

1. Система отправляет товар на ручную проверку
2. Статус меняется на "На проверке"
3. Администратор маркетплейса проверяет публикуемый товар в течение 24 часов
4. При одобрении администратора маркетплейса - публикация, при отклонении - отправка сообщения продавцу с указанием причин

UC-03: Внесение изменений в карточку товара

Актеры: Продавец

Предусловия:

- Продавец авторизован в системе
- Продавец вошел в личный кабинет (ЛК) продавца
- Товар опубликован на маркетплейсе и имеет статус "Опубликован"

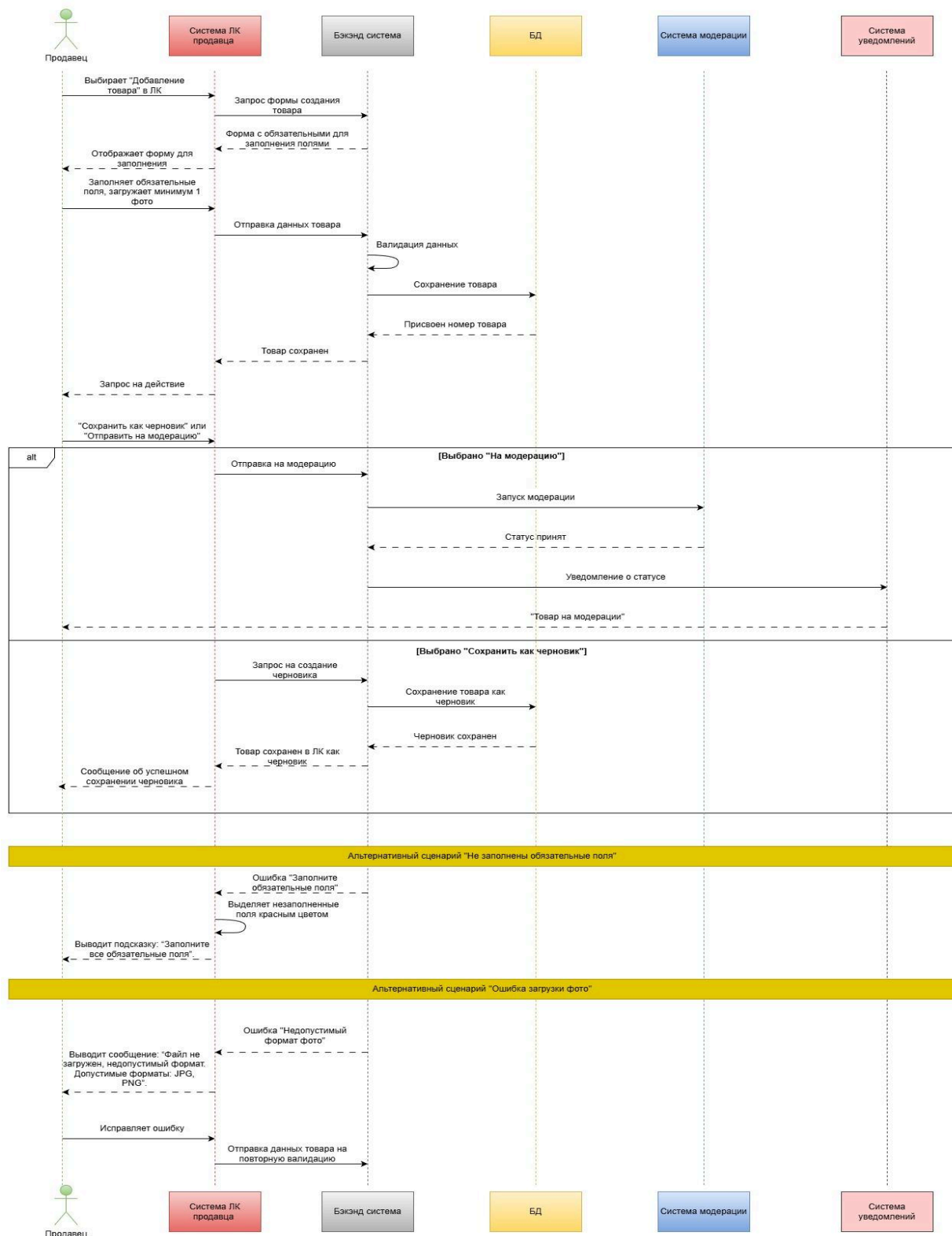
Основной сценарий:

1. Продавец открывает карточку нужного товара в разделе "Мои товары"
2. Нажимает кнопку "Редактировать"
3. Изменяет нужные поля
4. Нажимает "Сохранить изменения и отправить на модерацию"
5. Объявление переходит в статус "На модерации"
6. После успешной модерации, карточка товара публикуется с присвоением статуса "Опубликован".

Альтернативный сценарий: Не заполнены обязательные поля:

1. Система выделяет незаполненные поля красным.
2. Выводит подсказку: "Заполните все обязательные поля".

**Диаграмма процесса - UML Sequence Diagram**



### Задача 3

На картинке (скачать можно по ссылке -

[https://drive.google.com/file/d/1fi9JLRvO6bhS1uJJ2PUX8ZTOq0Eq\\_At0/view?usp=drive\\_link](https://drive.google.com/file/d/1fi9JLRvO6bhS1uJJ2PUX8ZTOq0Eq_At0/view?usp=drive_link)) изображен интерфейс регистрации Пользователя (как успешный экран создания Пользователя, так и варианты с ошибками).

#### Задание 3.1:

Требуется описать REST api интерфейс на стороне бэк сервиса, который должен вызываться фронтендом при нажатии на кнопку "Register". Описание должно включать:

- http метод + URL
- входные параметры (наименование, тип, обязательность, ограничения)
- выходные параметры при успешном ответе (наименование, тип, обязательность, ограничения)
- выходные параметры при ответе с ошибкой (наименование, тип, обязательность, ограничения)
- описание ошибок - коды ответов (главное верно указать клиентские или серверные) + текст сообщения
- пример запроса
- пример ответа (для успешного сценария и сценария с ошибкой).

Требований к формату нет (может быть в виде таблицы или просто текстом).

Допускается описание в формате openAPI спецификации. Полученный yaml приложить в виде текста к решению или в виде отдельного файла (обязательно не забыть открыть доступ).

#### Задание 3.2

Описать подробный пошаговый алгоритм, создания Пользователя на стороне бэк сервиса при вызове, получившегося в Задачи 3.1 метода. Алгоритм должен подробно описывать какие проверки требуется выполнять бэк сервису после получения запроса на создание нового Пользователя (при нажатии кнопки "Register"), прежде чем создать Пользователя в БД. Описание может быть в виде текста или диаграммы (например, activity)/блок-схемы.

### **Решение задачи №3**

Задание 3.1. Описание REST api интерфейса на стороне бэк сервиса.

```

1  Входные параметры. Описание REST API для регистрации пользователя.
2  Метод: POST
3  URL:/api/v1/register
4  Пример запроса на сервер: https://{домен}/api/v1/register
5
6  Наименование      Тип      Обязательность      Ограничения
7  first_name         str.     да                  Имя пользователя. 2-50 символов, только буквы латиницы, пробелы и дефисы
8  last_name          str.     да                  Фамилия пользователя. 2-50 символов, только буквы латиницы, пробелы и дефисы
9  user_name           str.     да                  Логин пользователя. 2-50 символов, допустимы буквы латиницы, пробелы, дефисы, числа
10 password           str.     да                  Пароль. Минимум 8 символов, должен содержать минимум 1 цифру, 1 заглавную букву и 1 спец. символ
11 captcha_token       boolean   да                  Токен, полученный от reCAPTCHA (длина ~200 символов)
12
13
14
15 Выходные параметры при успешном ответе. Описание REST API для регистрации пользователя.
16 Метод: POST
17 URL:/api/v1/register
18 Пример запроса на сервер: https://{домен}/api/v1/register
19
20 Наименование      Тип      Обязательность      Ограничения
21 first_name         str.     да                  Имя пользователя согласно запросу
22 last_name          str.     да                  Фамилия пользователя согласно запросу
23 user_name           str.     да                  Логин пользователя согласно запросу
24 password           str.     да                  Пароль согласно запросу
25 captcha_token       boolean   да                  Токен, должно быть true
26 user_id            str.     да                  UUID-идентификатор пользователя
27 user_status        str.     да                  Статус пользователя (active)
28
29
30 Выходные параметры при ответе с ошибкой. Описание REST API для регистрации пользователя.
31 Метод: POST
32 URL:/api/v1/register
33 Пример запроса на сервер: https://{домен}/api/v1/register
34
35 Наименование      Тип      Обязательность      Ограничения
36 error_code         str.     да                  Код ошибки
37 error_message       str.     да                  Сообщение об ошибке
38 error_field        str.     да                  Имя поля с ошибкой
39
40
41
42 Ошибки:
43 Клиентские ошибки:
44 400 Bad Request Требования к Captcha, ошибки валидации полей. Сообщение: Please verify reCaptcha to register!
45 409 Conflict Пользователь уже зарегистрирован в системе. Сообщение: User exists!
46 Серверные ошибки:
47 500 Internal Server Error Внутренняя ошибка сервера (например могут быть локальные ошибки, сбой передачи API, сбой в БД)
48 503 Service Unavailable Сервис временно недоступен (перегрузка сервера)
49
50
51 Пример запроса
52 Метод: POST
53 URL:/api/v1/register
54 Пример запроса на сервер: https://{домен}/api/v1/register
55 Request body:
56 {
57   "first_name": "Ivan",
58   "last_name": "Ivanov",
59   "user_name": "Ivan",
60   "password": "Ivanov@123",
61   "captcha_token": "bo0Kst0re35a"
62 }
63
64
65 Response body:
66 201 Created
67
68 {
69   "first_name": "Ivan",
70   "last_name": "Ivanov",
71   "user_name": "Ivan",
72   "password": "Ivanov@123",
73   "user_id": "123-q354-345g-hjkl",
74   "user_status": "active"
75 }
76
77 Для проверки Captcha
78 POST https://www.google.com/recaptcha/api/siteverify
79 Content-Type: application/x-www-form-urlencoded
80 secret=YOUR_SECRET_KEY&response=bo0Kst0re35a
81
82 JSON результат:
83 {
84   "success": true,
85   "action": "register",
86   "challenge_ts": "2026-01-10T10:00:00Z",
87   "hostname": "example.com"
88 }
89
90
91 Пример ответа с ошибкой:
92
93 {
94   "error_code": 409,
95   "error_message": "User exists!",
96   "error_field": "user_name"
97 }
98

```

Задание 3.2. Блок -схема, описывающая какие проверки требуется выполнять бэк сервису после получения запроса на создание нового Пользователя.

