

## Задание 1.

В приложении есть сущность “Карточка товара”, у которой в свою очередь есть параметр “Цена”. Карточку товара можно создавать, редактировать и удалять. Вы получаете новую доработку. В задаче у вас требование: “При резком изменении цены (на 30%) товара на попытку сохранить изменения должна отображаться нотификация “Резкое изменение цены”

- 1) Какие виды тестирования необходимо провести для тестирования этого требования?
  - ❖ **Функц.:** нотификация отображается только при изменении цены на 30% и только после попытки сохранить изменения (нажать на кнопку), как отображается (UI), как будет отображаться если нотификаций несколько
  - ❖ **Нефункц.:**
  - ❖ **Произв.:** нагрузочное – если есть возможность массово создавать, редактировать, удалять. Скорость – система не замедляется при отображении нотификации
  - ❖ **Совместимость:** проверка отображения в разных браузерах и на устройствах
  - ❖ **Локализация**
  - ❖ **Регрессионное:** проверка что функционал не поменялся
  - ❖ **UX:** проверка удобства
- 2) Напишите чеклист атомарных(низкоуровневых) проверок, необходимых для тестирования этого требования.

### Проверка граничных значений:

- ✚ Изменить цену товара на 30% и сохранить изменения – нотификация отображается
- ✚ Изменить цену товара на 29% и сохранить изменения - нотификация не отображается
- ✚ Изменить цену товара на 31% и сохранить изменения - нотификация отображается
- ✚ Не изменять цену и сохранить - нотификация не отображается

### Проверка отображения:

- ✚ Проверить отображение после нажатия на кнопку - нотификация отображается только после нажатия на кнопку
- ✚ Проверить отображение на разных устройствах - нотификация отображается корректно на всех устройствах
- ✚ Проверить отображение в разных браузерах - нотификация отображается корректно во всех браузерах
- ✚ Проверить отображение в разных версиях приложения под разные языки - нотификация отображается корректно во всех версиях с другой локализацией
- ✚ Проверить стиль и расположение - нотификация соответствует дизайну
- ✚ Проверить, что не мешает другим элементам UI - нотификация не перекрывает другие элементы UI

## Задание 2.

В некотором приложении есть функция «Получение оповещений». Пользователь может указать адрес эл почты в поле «email», адрес эл почты в поле «Дополнительный email», телефон (в поле «Телефон») или не указать ничего. Настройка «Получать оповещения» может быть отключена.

Логика получения оповещений следующая:

- Если указаны адрес эл почты (любой или оба), то пользователь получает письма на каждый из них
- Если указаны email (любой или оба) и телефон, то оповещения отправляются только на email,
- Если указан только телефон, то пользователь получает оповещения на него.
- Если ничего не указано или «Получать оповещения» выключено - не получает оповещений совсем.

A	B	C	D	E
	Email	Dop. Email	Phone	None
Если почта	+	+	-	-
Если почта и тел.	+	+	-	-
Если тел.	-	-	+	-
Если ничего	-	-	-	+

1) Напишите чек-лист всех проверок и ожидаемых результатов для каждой.

- ✓ Указать только почту в поле email и включить оповещения - оповещения отправляются на указанный email
- ✓ Указать только почту в поле доп. email и включить оповещения - оповещения отправляются на указанный email
- ✓ Указать почту в обоих полях и включить оповещения - оповещения отправляются на оба указанных email
- ✓ Указать почту и телефон и включить оповещения - оповещения отправляются только на указанный email
- ✓ Указать только телефон и включить оповещения - оповещения отправляются на указанный телефон
- ✓ Не указать ничего и включить оповещения - оповещения не отправляются
- ✓ Указать только телефон и отключить оповещения - оповещения не отправляются
- ✓ Указать только email и отключить оповещения – оповещения не отправляются
- ✓ Указать только доп. email и отключить оповещения - оповещения не отправляются
- ✓ Указать обе почты и телефон, отключить оповещения - оповещения не отправляются

2) Можно ли сократить количество проверок? Да

Если да, то какие не будете проводить?

- Те, где оповещения не отправляются, можно сократить до одной: при выключении «получать оповещения» - оповещения не отправляются

3) Какую технику тест-дизайна используете? **Таблица принятия решений, Эквивалентное значение**

### Задание 3.

Есть описание требований к задаче:

“Пользователь имеет возможность перейти в раздел “Товар” и увидеть в нем список товаров. Пользователь может создать новый заказ, нажав на кнопку “Создать”. У товара могут быть поля “Название”, “Описание”, “Цена”, “Количество”, “Доп. характеристика” и до 5 изображений, но не все поля обязательные. Изображения могут быть в формате .jpeg, .png и т.д. У пользователя нет ограничений по кол-ву созданных товаров.”

1) Есть ли ошибки в требованиях?

- Первое предложение не имеет фактического смысла – если его убрать, кол-во нужной информации не уменьшится
- Во втором предложении заказ – это товар?
- Не все поля обязательные – это какие?
- Что содержится в поле “Доп. характеристика”?
- «.jpeg, .png и т.д.» - какие т.д.?
- Есть ли вообще ограничения у остальных полей? Если да, то какие?
- Есть ли ограничение на размер изображения?
- «У пользователя нет ограничений по кол-ву созданных товаров» - это относится к кол-ву единиц товара в поле «количество», или к самому созданию неограниченного кол-ва «карточек» товара?
- Что происходит после нажатия на кнопку «Создать»?

2) Каким критериям качества не соответствуют эти требования?

- Ясность: не все требования ясны, и могут по-разному интерпретироваться.
- Полнота: не всё описано.
- Согласованность: не все термины согласованы (заказ = товар?, «доп. информация»)
- Неинформативность: первое предложение не несёт особой информации

3) Напишите список вопросов к заказчику исходя из 1 пункта.

- Какие поля являются обязательными?
- Есть ли в них ограничения?
- Какой размер изображений допустим? Какие ещё форматы изображений принимаются?
- Какой тип данных в каждом поле?

- Что содержится в поле “Доп. характеристика”?
- Как выглядит ошибка при не заполнении обязательных полей, не того формата и размера изображение?

#### Задание 4.

Есть команда, которая должна отдавать список задач пользователя /issues

Параметры команды:

- session - сессия пользователя
- all - выводит все задачи, по умолчанию выводятся только задачи в статусе “to do”
- type - опционально выдавать задачи только заданного типа (epic, story, task, bug)

Пример запроса

<http://api.ru/issues/?session=kjabkfjbela02839fhw39brwi3r&type=task&all=true>

#### 1) Напишите чек-лист с атомарными проверками для тестирования

Общее:

- Проверить, что API доступен по URL
- Проверить, что запрос без параметров возвращает корректный ответ
- Проверить, что некорректный запрос возвращает ошибку

Проверка session:

- Проверить, что корректно возвращает список задач пользователя
- Проверить, что при пустом значении в параметре выдаёт ошибку
- Проверить, что при неправильном параметре выдаёт ошибку

Проверка all:

- Проверить, что при all=true возвращает все задачи
- Проверить, что при all=false возвращает только задачи в статусе “to do”
- Проверить, что при пустом значении в параметре возвращает задачи в статусе “to do”
- Проверить, что при некорректном значении выдаёт ошибку

Проверка type:

- Проверить, что при корректном типе (type=story) возвращаются только задачи указанного типа
- Проверить, что при некорректном типе (type=type) возвращается ошибка
- Проверить, что при отсутствии параметра возвращаются все задачи

Проверка ответа:

- Проверить, что ответ имеет корректный статус (200)
- Проверить, что ответ возвращается в нужном формате

#### 2) Какой тип данных может передаваться в параметре all?

**Булевы – true/false**

3) Нужно ли что-то уточнить перед тестированием?

- Какие статусы задач существуют в параметре all
- Какие статусы ответов должны выдаваться при ошибках
- Какой ожидаемый формат ответа

4) Какой инструмент будете использовать для тестирования? **Postman**