**Singleview Test specialist Qualification Interview tasks (25 mins)**

**Task 1**

The contract for a companion product is created with binding period (configured period).

Binding period is also used to define charge override that can be applied to Customers for a specified period of time.

When a customer terminates the contract before the end date of the contract(binding period), there may be a penalty(cancellation) charge applied to the customer. For every contract defined, there is a specific field ‘A’. The field can be updated by an authorized CSR by putting a new value.

If the field is updated, the value on the field will be used to apply the cancellation charge to the customer.

Otherwise, the cancellation charge will be calculated as described below: Calculate how many months left until the end date of the contract Multiply the number of months left \* the recurring charge

**Task:**

a) Calculate Cancellation charge if:

Field 'A' = 100EUR

Binding Period = 12months

Start Date = 01/01/2011

Recurring Charge = 20EUR

End Date = 01/03/2011

**Total:** 20x12=240

**Debt:** 240-40+100=300

b) Calculate Cancellation charge if:

Field 'A' = empty

Binding Period = 12months

Start Date = 01/01/2011

Recurring Charge = 20EUR

End Date = 28/02/2011

**Total:** 12x20=240

**Debt:** 240-40=200

**Task 2**

Cuckoo Clock

Cuckoo Clock configuration is based on the following cycles:

a) every 3 hours (when hour hand hits 12, 3, 6 and 9 hours)

b) every 6 hours (when hour hand hits 12, 6 hours)

Cuckoo cycle can be changed any moment.

**Task:**

1. Think of all possible scenarios of testing for the Cuckoo Clock.

2. Imagine that on a B Cycle cuckoo is not working at 6. Report it to developers (Problem Description; Test Steps to reproduce the problem; Expected Results and Actual Results).

**Equivalences partitions:**

11:59,12:00,12:01

2:59,3:00,3:01

5:59,6:00,6:01

8:59,9:00,9:01

**Problem Description**:

Часы с кукушкой работают не верно, кукушка не кукует на 6 часах.

Убедиться что часы в рабочем состоянии, а так же установлены согласно инструкции и ничего не мешают им функционировать.

Механический завод в исправности и заведён, если часы электронные, тогда проверить исправность батарейки.

Часы имеют 2 цикла настроек, кукушка кукует каждые 3 часа или каждые 6 часов, который можно переключить в любой момент.

Что бы сократить кол-во тестов, применяем попарную технику тест дизайна.

**Test Steps to reproduce the problem:**

1. Повернуть минутную стрелку часов на 5:59
2. Подождать 1 минуту
3. На 6:00 часы не кукуют
4. Подождать 1 минуту, до 6:01 (убедиться, что часы не кукуют)
5. Повернуть минутную стрелку часов на 11:59
6. Подождать 1 минуту
7. На 12:00 часы не кукуют
8. Подождать 1 минуту, до 12:01 (убедиться, что часы не кукуют и через 6 часов)
9. Повернуть минутную стрелку часов на 2:59
10. Подождать 1 минуту
11. На 3:00 часы не кукуют
12. Подождать 1 минуту, до 3:01 (убедиться, что часы не кукуют и через 3 часа в другом цикле)
13. Повернуть минутную стрелку часов на 8:59
14. Подождать 1 минуту
15. На 9:00 часы не кукуют
16. Подождать 1 минуту, до 9:01 (убедиться, что часы не кукуют и через 6 часов в другом цикле)

**Additional notes:**

Первая проверка была проведена согласно циклу B на 6:00, следующая проверка проводится что бы проверить цикл кукушки через 6 часов на 12:00,

Что бы убедиться в каком цикле находятся часы, проводим проверку через 3 часа на 3:00 и заключительная проверка через 6 часов на 9:00.

За 4 теста мы покрыли проверки цикла А - every 3 hours (when hour hand hits 12, 3, 6 and 9 hours) и цикла В - every 6 hours (when hour hand hits 12, 6 hours)

**Ожидаемый результат:** кукушка кукует на 6 часах, на 12 часах и в другом цикле через 3 часа на 3:00, 6:00, 9:00, 12:00

**Фактический результат:** кукушка не кукует в 6:00, 12:00, 3:00, 9:00

**Task 3**

Testing Object: Water Cooler.

Test Data: Water Cooler has two buttons - Hot Water and Cold Water and Display with the temperature.

Task:

a) Write requirements for the Water Cooler.

b) Write Test Case for each requirement.

**User story:**

Я как пользователь, хочу налить холодной воды из кулера для утоления жажды

- Пользователь может использовать кулер и налить холодной воды нажав на кнопку Холодная вода

- Пользователь может увидеть температуру воды на дисплее кулера

**Requirements:**

1. Кулер должен быть подключён к системе подачи очищенной воды и электричеству 220 В с заземлением.
2. Кулер должен быть установлен внутри помещения на ровной, твёрдой поверхности.
3. У кулера есть отсек для установки посуды макс. в 300 мл.
4. Термостат отрегулирован для подогрева и охлаждения воды.
5. Производительность: Горячая вода => 85 C, холодная вода <= 10 C.
6. Дисплей кулера показывает данные Горячая вода +90 C, холодная вода +5 С.
7. Кулер может налить горячую воду нажав красную кнопку.
8. Кулер может налить холодную воду нажав синюю кнопку.

**Test Cases:**

1. Кулер должен подключён к сети и к подаче воды
2. Взять 300 мл стакан, поставить в отсек для подачи воды
3. Удерживать синюю кнопку для заполнения стакана холодной водой
4. На дисплее отображается температура воды +5С
5. Отпустить синюю кнопку для прекращения подачи холодной воды
6. Взять термометр и проверить температуры воды в стакане на соответствие заданным характеристикам (холодная вода <= 10 C)

**Acceptance creteria:**

- Когда пользователь нажимает на синюю кнопку с холодной водой, на дисплее экрана экрана отображается температура воды +5 С

- Пользователь может одновременно нажать только на одну кнопку

- Пользователь может поставить максимальный размер посуды 300 мл в отсек для набора воды

- Пользователь может налить стакан холодной воды, удерживая синюю кнопку с холодной водой.

**Task 4**

Read and elaborate on the idea of the request:

Regret period is a period during which the customer is allowed to regret the purchase of the product without receiving any product charges for the period the product was active.

Cancelling the product within the regret period will not result in cancellation fees for breaking the contract.

Regret functionality allows the charging of a recurring charge to be delayed until the customer has accepted the product for the configured regret period.

Acceptance is considered once the product has been active for the regret period. The period of regret is stored on the product instance.

Idea: Создать систему бронирования продукта для клиента, в течение которого в установленное время не будет взиматься плата, в этот момент продукт резервируется в системе для данного клиента и у клиента есть время для принятия решения. Клиент не несёт никаких финансовых потерь в статусе бронирования продукта и может отказаться от продукта. По истечению установленного времени статуса бронирования продукта, статус переходит в подтверждённый заказ.

User story: Я как клиент хочу забронировать продукт, для того что бы принять решение о покупке и согласовать оплату с руководством, при этом не нести финансовых рисков во время бронирования продукта.

- Клиент может бесплатно использовать систему бронирования для резервирования продукта в системе, статус продукта: «забронировано»

- Клиент может отказаться от продукта в статусе «забронировано» и не несёт финансовые потери

- Клиент может использовать систему бронирования до тех пор, пока продукт не перейдёт в статус «подтверждено» по истечению установленного времени в зависимости от уровня клиента:

1 уровень: стандартный (от 1 до 10 покупок в месяц) статус «забронировано» 1 день

2 уровень: продвинутый (от 11 до 50 покупок в месяц) статус «забронировано» 2 дня

3 уровень: специальный (от 51 и более покупок в месяц) статус «забронировано» 3 дня

- Клиент принимает условия автоматического перехода от статуса «забронировано» до статуса «подтверждено» по истечению установленного времени согласно уровню клиента.

Продукт со статусом «забронировано» отображается в базе данных в таблице orders, в колонке order\_reserve (datetime). Настроить интервал даты и времени для каждого клиента, согласно его уровня покупок в месяц.

**Task 5**

Database consists of 3 tables with the content presented below

|  |  |  |
| --- | --- | --- |
| **Table**  **Name** | **Field Name** | **Possible Values** |
| **ACCOUNT** | ⌂ ACCOUNTID |  |
|  | BALANCE |  |
|  | ACCOUNT\_STATE | 1- installed, 2- active, 3- inactive, 4- expired |
| **PRODUCT** | ⌂ ACCOUNTID |  |
|  | ⌂ OIDVAL |  |
|  | PRICEPLAN | "Spring", "Summer", "Winter" |
|  | PHONE\_NUM |  |
|  | VOICEMAIL | 0/1 |
| **SIMCARD** | ⌂ OIDVAL |  |
|  | SIM\_NUMBER |  |
|  | SIM\_STATUS | 1 - sim card is in use, 2 - not in use, 3 - disconnected |

Please write SQL statements giving output questioned below:

1. Select list of Phone numbers, current balances and sim card numbers using 'Spring' and 'Summer' priceplans and having Voicemail option enabled. One customer can have many sim cards, you should take into account only currently used one

2. Calculate and provide the count of accounts that have negative balance together with the total sum of negative balances. Group them by current status (account\_state).

SQL query 1:

SELECT PRODUCT.PHONE\_NUM AS Phone numbers,

ACCOUNT.BALANCE AS current balances,

SIMCARD.SIM\_NUMBER AS sim card numbers

FROM ACCOUNT

INNER JOIN PRODUCT

ON ACCOUNT.ACCOUNTID = PRODUCT.ACCOUNTID

INNER JOIN SIMCARD

ON PRODUCT.OIDVAL = SIMCARD.OIDVAL

WHERE (PRICEPLAN = "Spring" AND PRICEPLAN = "Summer")

AND VOICEMAIL = 1

AND SIM\_STATUS = 1;

SQL query 2:

SELECT ACCOUNT\_STATE, SUM(BALANCE)

FROM ACCOUNT

GROUP BY ACCOUNT\_STATE

HAVING SUM(BALANCE) < 0;