

Тест-кейсы по Use Case

Use case:

Название:	Тарификация абонентов оператора «Ромашка».
Триггер:	Получение CDR-файла.
Предусловия:	Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT. Существование тарифов «Классика» и «Помесячный» в БД на HRS.
Результат:	Успешная тарификация абонентов оператора «Ромашка».
Основной поток:	<ol style="list-style-type: none"> 1. Сервис CDR выступает в роли producer и отправляет CDR файл в topic. 2. Сервис BRT как consumer принимает данные с информацией о звонке в текстовом формате из topic. 3. Сервис BRT проводит авторизацию: проверяет номер абонентов из CDR файла, сравнивая их с данными, хранящимися в БД оператора «Ромашка», и добавляя идентификатор тарифа. 4. Сервис BRT выступает в роли producer и отправляет сообщение с данными о звонке и тарифе в формате JSON в topic. 5. Сервис HRS как consumer принимает данные в JSON формате из topic. Данные содержат информацию о длительности звонков абонентов. 6. Сервис HRS производит расчет суммы списания для каждого абонента. 7. Сервис HRS выступает в роли producer и отправляет сообщение в формате JSON в topic, которое содержит данные о сумме списания с баланса каждого абонента. 8. Сервис BRT как consumer принимает данные о сумме списания в формате JSON из topic. 9. Сервис BRT списывает полученную сумму с каждого абонента.

Альтернативные потоки:	<p>1. а Ошибка передачи CDR-файла 1.а.1 Возвращается код ошибки 4xx/5xx.</p> <p>2. а Ошибка считывания CDR-файла 2.а.1 Возвращается код ошибки 4xx/5xx.</p> <p>3.а Ошибка авторизации абонентов 3.а.1 Сервис BRT возвращает код ошибки 4xx/5xx.</p> <p>4.а Ошибка передачи данных о звонке и тарифе абонентов 4.а.1 Возвращается код ошибки 4xx/5xx.</p> <p>5.а Ошибка считывания данных о звонке и тарифе абонентов 5.а.1 Возвращается код ошибки 4xx/5xx.</p> <p>6.а Ошибка расчета суммы списания 6.а.1 Сервис HRS возвращает код ошибки 4xx/5xx.</p>
	<p>7.а Ошибка передачи данных о сумме списания 7.а.1 Возвращается код ошибки 4xx/5xx.</p> <p>8.а Ошибка считывания данных о сумме списания 8.а.1 Возвращается код ошибки 4xx/5xx.</p> <p>9.а Ошибка списания с баланса 9.а.1 BRT возвращает код ошибки 4xx/5xx.</p>

Тест-кейс №1.1:

Идентификатор: T-UC1-01

Заголовок тест-кейса: Тарификация исходящего звонка абонента оператора «Ромашка» по тарифу «Классика» внутри сети

Приоритет: 1

Связь с требованием: Use Case «Тарификация абонентов оператора ‘Ромашка’»

Модуль и подмодуль ПО: CDR, BRT, HRS

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT;
- Существование тарифов «Классика» и «Помесячный» в БД на HRS;
- Известен номер абонент, который совершает вызов, и его исходный баланс (не меньше 100 у.е.):
SELECT Phone_number, Balance

FROM USER
WHERE ID_tariff = '11' AND Balance >= 100;

Обозначим для удобства Phone_number_1, Balance_1

- Известен абонент, которому поступает вызов:
SELECT Phone_number, Balance
FROM USER
WHERE ID_tariff = '11' AND Phone_number <> 'Phone_number_1';

Обозначим для удобства Phone_number_2, Balance_2

- В CDR-файле 10 записей;
- CDR-файл придерживается следующих требований:
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601);
 - дата и время окончания звонка (ISO 8601);
 - разделитель данных – запятая;
 - разделитель записей – перенос строки;
 - данные обязательно формируются в хронологическом порядке;
 - CDR может быть обычным txt/csv.

*CEILING используется для округления в большую сторону (необходимо по ТЗ)

Шаги тест-кейса:	Ожидаемый результат:
1. Эмуляция генерации CDR-файла с предусловием 5, 6 между абонентами Phone_number_1 и Phone_number_2 длительностью N минут;	В BRT появился CDR-файл с зеркальными записями о звонке
2. Проверка корректности CDR-файла: SELECT ID, Call_type, Phone_caller, Phone_called, CEILING(TIMESTAMPDIFF(MINUTE, t.Start_time, t.Finish_time)) AS Duration FROM TRANSACTION WHERE (Phone_caller = 'Phone_number_1' AND Phone_called = 'Phone_number_2') AND (Phone_caller = 'Phone_number_2' AND Phone_called = 'Phone_number_1');	Вывело две записи из CDR-файла. Допустим, что выведенное время звонка = 16 минут
3. Посчитать сумму, которую должен списать BRT по формуле (Длительность звонка * 1,5)	16 * 1,5 = 24 у.е Итоговый баланс: Balance_1 – 24 Произошел расчет средств в HRS, результат на списание отправлен BRT
4. Сравнить полученную сумму с суммой, которую списал BRT: SELECT Phone_number, Balance FROM USER	В BRT списана корректная сумма

WHERE ID_tariff = '11' AND Phone_number = 'Phone_number_1';	
--	--

Тест-кейс №1.2:

Идентификатор: T-UC1-02

Заголовок тест-кейса: Тарификация исходящего звонка абонента оператора «Ромашка» по тарифу «Классика» абоненту другого оператора

Приоритет: 1

Связь с требованием: Use Case «Тарификация абонентов оператора «Ромашка»»

Модуль и подмодуль ПО: CDR, BRT, HRS

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT;
- Существование тарифов «Классика» и «Помесячный» в БД на HRS;
- Известен номер абонент, который совершает вызов, и его исходный баланс (не меньше 100 у.е.):

```
SELECT Phone_number, Balance
FROM USER
```

```
WHERE ID_tariff = '11' AND Balance >= 100;
```

Обозначим для удобства Phone_number_1, Balance_1

- Известен абонент, которому поступает звонок: Phone_number_2;
- В CDR-файле 10 записей;
- CDR-файл придерживается следующих требований:
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601);
 - дата и время окончания звонка (ISO 8601);
 - разделитель данных – запятая;
 - разделитель записей – перенос строки;
 - данные обязательно формируются в хронологическом порядке;
 - CDR может быть обычным txt/csv.

Шаги тест-кейса:	Ожидаемый результат:
1. Эмуляция генерации CDR-файла с предусловием 5, 6 между абонентами Phone_number_1 и Phone_number_2 длительностью N минут;	В BRT появился CDR-файл записью о звонке
2. Проверка корректности CDR-файла: SELECT ID, Call_type, Phone_caller, Phone_called, CEILING(TIMESTAMPDIFF(MINUTE, t.Start_time, t.Finish_time)) AS Duration FROM TRANSACTION WHERE Phone_caller = 'Phone_number_1' AND	Вывело одну запись из CDR-файла. Допустим, что выведенное время звонка = 16 минут

Phone_called = 'Phone_number_2';	
3. Посчитать сумму, которую должен списать BRT по формуле (Длительность звонка * 2,5)	16 * 2,5 = 40 у.е Итоговый баланс: Balance_1 – 40 Произошел расчет средств в HRS, результат на списание отправлен BRT
4. Сравнить полученную сумму с суммой, которую списал BRT: SELECT Phone_number, Balance FROM USER WHERE ID_tariff = '11' AND Phone_number = 'Phone_number_1';	В BRT списана корректная сумма

Тест-кейс №1.3:

Идентификатор: T-UC1-03

Заголовок тест-кейса: Тарификация исходящего звонка абонента оператора «Ромашка» по тарифу «Помесячный» с достаточным количеством минут для оплаты звонка

Приоритет: 1

Связь с требованием: Use Case «Тарификация абонентов оператора «Ромашка»»

Модуль и подмодуль ПО: CDR, BRT, HRS

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT.
- Существование тарифов «Классика» и «Помесячный» в БД на HRS.
- Известен номер абонент, который совершает вызов, и его исходный баланс (не меньше 100 у.е.):
SELECT u.Phone_number, u.Balance, ts.Quantity_spent
FROM USER u
INNER JOIN TARIFF_STATUS ts ON u.Phone_number = ts.Phone_number
WHERE u.ID_tariff = '12' AND u.Balance >= 100 AND 30<=ts.Quantity_spent <= 50;

Обозначим для удобства Phone_number_1, Balance_1, Minute (ts.Quantity_spent)

- Известен абонент, которому поступает звонок: Phone_number_2;
- В CDR-файле 10 записей;
- CDR-файл придерживается следующих требований:
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601);
 - дата и время окончания звонка (ISO 8601);
 - разделитель данных – запятая;
 - разделитель записей – перенос строки;
 - данные обязательно формируются в хронологическом порядке;
 - CDR может быть обычным txt/csv.

Шаги тест-кейса:	Ожидаемый результат:
1. Эмуляция генерации CDR-файла с предусловием 5, 6 между абонентами Phone_number_1 и Phone_number_2 длительностью N минут;	В BRT появился CDR-файл записью о звонке
2. Проверка корректности CDR-файла: SELECT ID, Call_type, Phone_caller, Phone_called, CEILING(TIMESTAMPDIFF(MINUTE, t.Start_time, t.Finish_time)) AS Duration FROM TRANSACTION WHERE Phone_caller = 'Phone_number_1' AND Phone_called = 'Phone_number_2';	Вывело одну запись (или две записи, не указано точно в предусловии какому оператору принадлежит номер 2) из CDR-файла. Допустим, что выведенное время звонка = 16 минут
3. Посчитать количество средств, которые должны списаться	По условию минут хватает, переход на тариф «Классика» не произошел Остаток минут: Minute - 16 Итоговый баланс: Balance_1 Произошел расчет средств в HRS
4. Сравнить полученное количество минут с минутами, которые списал HRS: SELECT u.Phone_number, u.Balance, ts.Quantity_spent FROM USER u INNER JOIN TARIFF_STATUS ts ON u.Phone_number = ts.Phone_number WHERE u.ID_tariff = '12';	В HRS списано корректное количество минут.

Не совсем уверена по поводу того, где хранятся минуты. В документации от аналитика написано HRS, разработчик на данный момент еще не реализовал это все, поэтому сужу по документации.

Тест-кейс №1.4:

Идентификатор: T-UC1-04

Заголовок тест-кейса: Тарификация исходящего звонка абонента оператора «Ромашка» по тарифу «Помесячный» с количеством минут, хватившим на часть звонка, внутри сети

Приоритет: 1

Связь с требованием: Use Case «Тарификация абонентов оператора «Ромашка»»

Модуль и подмодуль ПО: CDR, BRT, HRS

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT.
- Существование тарифов «Классика» и «Помесячный» в БД на HRS.

- Известен номер абонент, который совершает вызов, и его исходный баланс (не меньше 100 у.е.):
SELECT u.Phone_number, u.Balance, ts.Quantity_spent
FROM USER u
INNER JOIN TARIFF_STATUS ts ON u.Phone_number = ts.Phone_number
WHERE u.ID_tariff = '12' AND u.Balance >= 100 AND 1<=ts.Quantity_spent <= 10;

Обозначим для удобства Phone_number_1, Balance_1, Minute (ts.Quantity_spent)

- Известен абонент, которому поступает вызов:
SELECT Phone_number, Balance
FROM USER
WHERE ID_tariff = '11' AND Phone_number <> 'Phone_number_1';

Обозначим для удобства Phone_number_2, Balance_2

- В CDR-файле 10 записей;
- CDR-файл придерживается следующих требований:
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601);
 - дата и время окончания звонка (ISO 8601);
 - разделитель данных – запятая;
 - разделитель записей – перенос строки;
 - данные обязательно формируются в хронологическом порядке;
 - CDR может быть обычным txt/csv.

Шаги тест-кейса:	Ожидаемый результат:
1. Эмуляция генерации CDR-файла с предусловием 5, 6 между абонентами Phone_number_1 и Phone_number_2 длительностью N минут;	В BRT появился CDR-файл записью о звонке
2. Проверка корректности CDR-файла: SELECT ID, Call_type, Phone_caller, Phone_called, CEILING(TIMESTAMPDIFF(MINUTE, t.Start_time, t.Finish_time)) AS Duration FROM TRANSACTION WHERE (Phone_caller = 'Phone_number_1' AND Phone_called = 'Phone_number_2') AND (Phone_caller = 'Phone_number_2' AND Phone_called = 'Phone_number_1');	Вывело две записи из CDR-файла. Допустим, что выведенное время звонка = 16 минут
3. Посчитать количество средств, которые должны списаться	По условию минут хватает лишь на оплату части звонка, переход на тариф «Классика» произошел Допустим, что известное количество минут = 10 Остаток минут: 10 – 16 = -6, списываем 10 минут, 6 минут перерасчитываем по

	<p>тарифу «Классика» Необходимо рассчитать итоговый баланс: Итоговый баланс: $\text{Balance_1} - 1,5 * 6$</p> <p>Произошел расчет средств в HRS, результат на списание отправлен BRT</p>
<p>4. Сравнить полученное количество минут с минутами, которые списал HRS и сравнить итоговый баланс:</p> <pre>SELECT u.Phone_number, u.Balance, ts.Quantity_spent FROM USER u INNER JOIN TARIFF_STATUS ts ON u.Phone_number = ts.Phone_number WHERE u.ID_tariff = '12';</pre>	<p>В HRS списано корректное количество минут, в BRT списана корректная сумма</p>

Тест-кейс №1.5:

Идентификатор: T-UC1-05

Заголовок тест-кейса: Тарификация исходящего звонка абонента оператора «Ромашка» по тарифу «Помесячный» с количеством минут, хватившим на часть звонка, при звонке абоненту другого оператора связи

Приоритет: 1

Связь с требованием: Use Case «Тарификация абонентов оператора «Ромашка»»

Модуль и подмодуль ПО: CDR, BRT, HRS

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT.
- Существование тарифов «Классика» и «Помесячный» в БД на HRS.
- Известен номер абонент, который совершает вызов, и его исходный баланс (не меньше 100 у.е.):

```
SELECT u.Phone_number, u.Balance, ts.Quantity_spent
FROM USER u
INNER JOIN TARIFF_STATUS ts ON u.Phone_number = ts.Phone_number
WHERE u.ID_tariff = '12' AND u.Balance >= 100 AND 1<=ts.Quantity_spent <= 10;
```

Обозначим для удобства Phone_number_1, Balance_1, Minute (ts.Quantity_spent)

- Известен абонент, которому поступает вызов: Phone_number_2;
- В CDR-файле 10 записей;
- CDR-файл придерживается следующих требований:
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601);
 - дата и время окончания звонка (ISO 8601);
 - разделитель данных – запятая;
 - разделитель записей – перенос строки;

- данные обязательно формируются в хронологическом порядке;
- CDR может быть обычным txt/csv.

Шаги тест-кейса:	Ожидаемый результат:
1. Эмуляция генерации CDR-файла с предусловием 5, 6 между абонентами Phone_number_1 и Phone_number_2 длительностью N минут;	В BRT появился CDR-файл записью о звонке
2. Проверка корректности CDR-файла: SELECT ID, Call_type, Phone_caller, Phone_called, CEILING(TIMESTAMPDIFF(MINUTE, t.Start_time, t.Finish_time)) AS Duration FROM TRANSACTION WHERE Phone_caller = 'Phone_number_1' AND Phone_called = 'Phone_number_2';	Вывело одну запись из CDR-файла. Допустим, что выведенное время звонка = 16 минут
3. Посчитать количество средств, которые должны списаться	По условию минут хватает лишь на оплату части звонка, переход на тариф «Классика» произошел Допустим, что известное количество минут = 10 Остаток минут: $10 - 16 = -6$, списываем 10 минут, 6 минут перерасчитываем по тарифу «Классика» Необходимо рассчитать итоговый баланс: Итоговый баланс: $Balance_1 - 2,5 * 6$ Произошел расчет средств в HRS, результат на списание отправлен BRT
4. Сравнить полученное количество минут с минутами, которые списал HRS и сравнить итоговый баланс: SELECT u.Phone_number, u.Balance, ts.Quantity_spent FROM USER u INNER JOIN TARIFF_STATUS ts ON u.Phone_number = ts.Phone_number WHERE u.ID_tariff = '12';	В HRS списано корректное количество минут, в BRT списана корректная сумма

Тест-кейс №1.6:

Идентификатор: T-UC1-06

Заголовок тест-кейса: Тарификация исходящего звонка абонента оператора «Ромашка» по тарифу «Помесячный» с законченным пакетом минут

Приоритет: 1

Связь с требованием: Use Case «Тарификация абонентов оператора «Ромашка»»

Модуль и подмодуль ПО: CDR, BRT, HRS

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT.
- Существование тарифов «Классика» и «Помесячный» в БД на HRS.
- Известен номер абонент, который совершает вызов, и его исходный баланс (не меньше 100 у.е.):

```
SELECT u.Phone_number, u.Balance, ts.Quantity_spent
FROM USER u
INNER JOIN TARIFF_STATUS ts ON u.Phone_number = ts.Phone_number
WHERE u.ID_tariff = '12' AND u.Balance >= 100 AND ts.Quantity_spent = 0;
```

Обозначим для удобства Phone_number_1, Balance_1, Minute (ts.Quantity_spent)

- Известен абонент, которому поступает звонок: Phone_number_2;
- В CDR-файле 10 записей;
- CDR-файл придерживается следующих требований:
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601);
 - дата и время окончания звонка (ISO 8601);
 - разделитель данных – запятая;
 - разделитель записей – перенос строки;
 - данные обязательно формируются в хронологическом порядке;
 - CDR может быть обычным txt/csv.

Шаги тест-кейса:	Ожидаемый результат:
1. Эмуляция генерации CDR-файла с предусловием 5, 6 между абонентами Phone_number_1 и Phone_number_2 длительностью N минут;	В BRT появился CDR-файл записью о звонке
2. Проверка корректности CDR-файла: SELECT ID, Call_type, Phone_caller, Phone_called, CEILING(TIMESTAMPDIFF(MINUTE, t.Start_time, t.Finish_time)) AS Duration FROM TRANSACTION WHERE Phone_caller = 'Phone_number_1' AND Phone_called = 'Phone_number_2';	Вывело одну запись (или две записи, не указано точно в предусловии какому оператору принадлежит номер 2) из CDR-файла. Допустим, что выведенное время звонка = 16 минут
3. Посчитать количество средств, которые должны списаться	По условию минут не хватает, произошел переход на тариф Остаток минут: 0 Итоговый баланс(если внутри сети): Balance_1 – 16*1,5 Итоговый баланс(если не внутри сети): Balance_1 – 16*2,5 Произошел расчет средств в HRS,

	передано на списание в BRT
<p>4. Сравнить списание средств:</p> <pre>SELECT u.Phone_number, u.Balance, ts.Quantity_spent FROM USER u INNER JOIN TARIFF_STATUS ts ON u.Phone_number = ts.Phone_number WHERE u.ID_tariff = '12';</pre>	В BRT списано корректное количество у.е..

Тест-кейс №1.7:

Идентификатор: T-UC1-07

Заголовок тест-кейса: Тарификация абонентов оператора «Ромашка» с некорректным CDR-файлом (формат файла не txt/csv)

Приоритет: 2

Связь с требованием: Use Case «Тарификация абонентов оператора «Ромашка»»

Модуль и подмодуль ПО: CDR: эмулятор, коммутатор

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT;
- Существование тарифов «Классика» и «Помесячный» в БД на HRS;
- Известен номер абонент, который совершает вызов: Phone_number_1;
- Известен абонент, которому поступает вызов: Phone_number_2;
- В CDR-файле 10 записей;
- Формат CDR-файла doc;
- CDR-файл придерживается формата (
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601);
 - дата и время окончания звонка (ISO 8601);
 - разделитель данных – запятая;
 - разделитель записей – перенос строки).

Шаги тест-кейса:	Ожидаемый результат:
<p>1. Эмуляция генерации CDR-файла с предусловием 5, 6, 7</p> <p>Проверка корректности CDR-файла: SELECT * FROM TRANSACTION</p>	<p>BRT выдает ошибку (400) «Некорректный формат CDR-файла»</p>

Описание эмуляции на данный момент неизвестно, разработчик не реализовал данный функционал.

Тест-кейс №1.8:

Идентификатор: T-UC1-08

Заголовок тест-кейса: Тарификация абонентов оператора «Ромашка» с некорректным CDR-файлом (отрицательное время звонка)

Приоритет: 2

Связь с требованием: Use Case «Тарификация абонентов оператора ‘Ромашка’»

Модуль и подмодуль ПО: CDR: эмулятор, коммутатор

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT.
- Существование тарифов «Классика» и «Помесячный» в БД на HRS.
- В CDR-файле 10 записей;
- CDR-файл придерживается формата (
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601): «2024-11-16T04:25:03Z» ;
 - дата и время окончания звонка (ISO 8601): «2020-11-16T04:25:03Z»;
 - разделитель данных – запятая;
 - разделитель записей – перенос строки;
 - в рамках задания CDR может быть обычным txt/csv).

Шаги тест-кейса:	Ожидаемый результат:
1. Эмуляция генерации CDR-файла с предусловием 3,4 Проверка корректности CDR-файла: SELECT * FROM TRANSACTION	BRT выдает ошибку (400) «Некорректное содержание CDR-файла»

Описание эмуляции на данный момент неизвестно, разработчик не реализовал данный функционал.

Тест-кейс №1.9:

Идентификатор: T-UC1-09

Заголовок тест-кейса: Разделение звонка на две записи: до и после полуночи

Приоритет: 3

Связь с требованием: Use Case «Тарификация абонентов оператора ‘Ромашка’»

Модуль и подмодуль ПО: CDR, BRT, HRS

Предусловия:

- Установленный список абонентов «Ромашка» (не менее 10) хранится в БД на BRT;
- Существование тарифов «Классика» и «Помесячный» в БД на HRS;
- Сервис запущен;

- Абоненты свободны;
- Известен номер абонент, который совершает вызов, и его исходный баланс (не меньше 100 у.е.):

```
SELECT Phone_number, Balance
FROM USER
WHERE ID_tariff = '11' AND Balance >= 100;
```

Обозначим для удобства Phone_number_1, Balance_1

- Известен абонент, которому поступает вызов: Phone_number_2;
- В CDR-файле 10 записей;
- CDR-файл придерживается следующих требований:
 - тип вызова (01 - исходящие, 02 - входящие);
 - номер обслуживаемого абонента;
 - номер абонента, с которым осуществлялась связь;
 - дата и время начала звонка (ISO 8601);
 - дата и время окончания звонка (ISO 8601);
 - разделитель данных – запятая;
 - разделитель записей – перенос строки;
 - данные обязательно формируются в хронологическом порядке;
 - CDR может быть обычным txt/csv.

Шаги тест-кейса:	Ожидаемый результат:
1. Эмуляция генерации CDR-файла с предусловием 7, 8 между абонентами Phone_number_1 и Phone_number_2 длительностью N минут, который начался до 0:00:00 и продолжается некоторое время после	В BRT появился CDR-файл с двумя записями о звонке
2. Проверка корректности CDR-файла: <pre>SELECT ID, Call_type, Phone_caller, Phone_called, CEILING(TIMESTAMPDIFF(MINUTE, t.Start_time, t.Finish_time)) AS Duration FROM TRANSACTION WHERE Phone_caller = 'Phone_number_1' AND Phone_called = 'Phone_number_2'</pre>	<p>Вывело две записи из CDR-файла. Допустим, что выведенное время звонка = 16 минут</p> <p>Duration с двух записей должно в сумме давать 16 минут</p>

№1 Use case:

Название:	Изменение тарифа абонента оператора «Ромашка».
Триггер:	Менеджер отправляет запрос на смену тарифа абонента.
Предусловия:	Абонент существует в БД абонентов оператора «Ромашка». Новый тариф существует в БД тарифов.
Результат:	Тариф абонента оператора «Ромашка» успешно изменен.
Основной поток:	1. Менеджер авторизуется в системе через Postman, указывая логин и пароль. 2. Postman проверяет данные о менеджере и успешно авторизует его в системе. 3. Менеджер отправляет запрос на смену тарифа абонента через Postman, указывая номер телефона и идентификатор нового тарифа. 4. Postman передает запрос на смену тарифа CRM. 5. CRM, получая запрос, формирует информацию об абоненте и новом идентификаторе тарифа. 6. CRM передает информацию в БД абонентов оператора «Ромашка». 7. БД проводит операцию смены тарифа для указанного абонента.
Альтернативные потоки:	2.а Ошибка авторизации менеджера 2.а.1 Возвращается ошибка. 3.а Ошибка отправки запроса на смену тарифа абонента 3.а.1 Возвращается ошибка. 4.а Ошибка передачи запроса на смену тарифа абонента 4.а.1 Возвращается ошибка. 6.а Ошибка отправки запроса с информацией о новом тарифе 6.а.1 Возвращается ошибка. 7.а Ошибка смены тарифа абонента 7.а.1 Возвращается ошибка.

Тест-кейс №2.1.**Идентификатор:** T-UC2-01**Заголовок тест-кейса:** Изменение тарифа абонента оператора «Ромашка»**Приоритет:** 2**Связь с требованием:** №1 Use case «Изменение тарифа абонента оператора «Ромашка»»**Модуль и подмодуль ПО:** CRM-система, HRS, BRT**Предусловия:**

- Имеется учетная запись роли менеджер;
- Известен логин и пароль учетной записи роли менеджер;
- Найти действующего абонента оператора «Ромашка», зафиксировать номер телефона абонента;
 SELECT Phone_number, ID_tariff
 FROM USER
 WHERE Phone_number = 'номер действующего абонента «Ромашки»'
- Новый тариф не соответствует нынешнему тарифному плану, зафиксировать id тарифного плана;
 SELECT ID_tariff
 FROM TARIFF
 WHERE ID_tariff <>
 (SELECT ID_tariff
 FROM USER
 WHERE Phone_number = 'номер действующего абонента «Ромашки»')
- Авторизоваться с учётной записью из условия 1,2

Шаги тест-кейса:	Ожидаемый результат:
1. Отправить запрос на смену тарифного плана, учитывая успешное выполнение последнего предусловия <i>Входные данные:</i> номер телефона абонента, id тарифа (из предусловия 3,4)	Получен код ответа 2xx
2. Проверить данные в ответе	Получен код ответа 2xx
3. Отправить SQL-запрос в BRT	Id тарифа успешно сменен

```

SELECT Phone_number, ID_tariff
FROM USER
WHERE Phone_number = 'номер действующего абонента «Ромашки»'

```

Тест-кейс №2.2.

Идентификатор: T-UC2-02

Заголовок тест-кейса: Авторизации менеджера в CRM-системе с неверным паролем.

Приоритет: 3

Связь с требованием: №1 Use case «Изменение тарифа абонента оператора «Ромашка»»

Модуль и подмодуль ПО: CRM-система

Предусловия:

- Имеется учетная запись роли менеджер;
- Известен логин учетной записи роли менеджер.

Шаги тест-кейса:	Ожидаемый результат:
1. Отправить запрос авторизации	«Введите логин»
2. Вводится логин из предусловия	«Введите пароль»

3. Вводится неверный (неизвестный пароль)	Возвращена ошибка авторизации «Неверный пароль»
---	--

Тест-кейс №2.3.

Идентификатор: T-UC2-03

Заголовок тест-кейса: Авторизации менеджера в CRM-системе несуществующим логином

Приоритет: 3

Связь с требованием: №1 Use case «Изменение тарифа абонента оператора ‘Ромашка’»

Модуль и подмодуль ПО: CRM-система

Предусловия:

- Имеется учетная запись роли менеджер.

Шаги тест-кейса:	Ожидаемый результат:
1. Отправить запрос авторизации	«Введите логин»
2. Вводится несуществующий логин	Возвращена ошибка авторизации «Неверный логин»

Тест-кейс №2.4.

Идентификатор: T-UC2-04

Заголовок тест-кейса: Смена тарифного плана абонента на несуществующий в БД HRS

Приоритет: 3

Связь с требованием: №1 Use case «Изменение тарифа абонента оператора ‘Ромашка’»

Модуль и подмодуль ПО: CRM-система, HRS, BRT

Предусловия:

- Имеется учетная запись роли менеджер;
- Известен логин и пароль учетной записи роли менеджер;
- Найти действующего абонента оператора «Ромашка», зафиксировать номер телефона абонента;
SELECT Phone_number, ID_tariff
FROM USER
WHERE Phone_number = 'номер действующего абонента «Ромашки»'
- Тарифный план, id которого нет в бд HRS;
SELECT ID_tariff
FROM TARIFF
- Авторизоваться с учётной записью из условия 1,2

Шаги тест-кейса:	Ожидаемый результат:
------------------	----------------------

1. Отправить запрос на смену тарифного плана, учитывая успешное выполнение последнего предусловия <i>Входные данные:</i> номер телефона абонента (из предусловия 3), id тарифа (12346676)	Получен код 4xx(404) «Тариф не найден»
--	--

Тест-кейс №2.5.

Идентификатор: T-UC2-05

Заголовок тест-кейса: Смена тарифного плана у абонента с другим оператором связи

Приоритет: 3

Связь с требованием: №1 Use case «Изменение тарифа абонента оператора ‘Ромашка’»

Модуль и подмодуль ПО: CRM-система, HRS, BRT

Предусловия:

- Имеется учетная запись роли менеджер;
- Известен логин и пароль учетной записи роли менеджер;
- Найти действующего абонента оператора «Ромашка», выбрать номер телефона абонента, вне базы данных (другой оператор связи);
SELECT Phone_number, ID_tariff
FROM USER
- Новый тариф, выбрать и зафиксировать id тарифного плана;
SELECT ID_tariff
FROM TARIFF
- Авторизоваться с учётной записью из условия 1,2

Шаги тест-кейса:	Ожидаемый результат:
1. Отправить запрос на смену тарифного плана, учитывая успешное выполнение последнего предусловия <i>Входные данные:</i> номер телефона абонента, id тарифа (из предусловия 3,4)	Получена ошибка 4xx «Абонент не найден в системе»

Тест-кейс №2.6.

Идентификатор: T-UC2-06

Заголовок тест-кейса: Смена тарифного плана у абонента с несуществующим номером (номер телефона имеет невалидный вид, несоответствующий маске ввода «+7 (9__)-__-__»)

Приоритет: 3

Связь с требованием: №1 Use case «Изменение тарифа абонента оператора ‘Ромашка’»

Модуль и подмодуль ПО: CRM-система, HRS, BRT

Предусловия:

- Имеется учетная запись роли менеджер;
- Известен логин и пароль учетной записи роли менеджер;
- Номер телефона, имеющий вид: 1234567891012345 (или номер телефона, в котором есть буквы или (за исключением «+»))
- Новый тариф, выбрать и зафиксировать id тарифного плана;
SELECT ID_tariff
FROM TARIFF
- Авторизоваться с учётной записью из условия 1,2

Шаги тест-кейса:	Ожидаемый результат:
<p>1. Отправить запрос на смену тарифного плана, учитывая успешное выполнение последнего условия</p> <p><i>Входные данные:</i> номер телефона абонента, id тарифа (из условия 3,4)</p>	<p>Получена ошибка 4xx «Номер телефона введен некорректно»</p>

№2 Use case:

Название:	Пополнение баланса абонента оператора «Ромашка».
Триггер:	Абонент отправляет запрос на пополнение баланса.
Предусловия:	Абонент существует в БД абонентов оператора «Ромашка».
Результат:	Баланс абонента успешно пополнен.
Основной поток:	<ol style="list-style-type: none">1. Абонент авторизуется в системе через Postman, указывая свой номер телефона.2. Postman проверяет данные об абоненте и успешно авторизует его в системе.3. Абонент отправляет запрос на пополнение баланса через Postman, указывая сумму для пополнения.4. Postman передает запрос на пополнение баланса CRM.5. CRM, получая запрос, формирует информацию об абоненте и сумме для пополнения.6. CRM передает информацию в БД абонентов оператора «Ромашка».7. БД проводит операцию пополнения баланса для указанного абонента.
Альтернативные потоки:	<ol style="list-style-type: none">2.а Ошибка авторизации абонента<ol style="list-style-type: none">2.а.1 Возвращается ошибка.3.а Ошибка отправки запроса на пополнение баланса<ol style="list-style-type: none">3.а.1 Возвращается ошибка.4.а Ошибка передачи запроса на пополнение баланса<ol style="list-style-type: none">4.а.1 Возвращается ошибка.6.а Ошибка отправки запроса на пополнение баланса<ol style="list-style-type: none">6.а.1 Возвращается ошибка.7.а Ошибка пополнения баланса<ol style="list-style-type: none">7.а.1 Возвращается ошибка..

Тест-кейс №3.1.

Идентификатор: T-UC3-01

Заголовок тест-кейса: Пополнение абонентом своего баланса оператора «Ромашка»

Приоритет: 2

Связь с требованием: №2 Use case «Пополнение баланса абонента оператора ‘Ромашка’»

Модуль и подмодуль ПО: CRM-система, BRT

Предусловия:

- Имеется учетная запись абонента, зафиксирован номер телефона абонента;
- Исходный баланс определен:
SELECT Phone_number, Balance
FROM USER
WHERE Phone_number = 'номер действующего абонента «Ромашки»'
- Имеется сумма для пополнения баланса (положительная ненулевая);
- Успешная авторизация абонента с номером телефона из предусловия 1

Шаги тест-кейса:	Ожидаемый результат:
1. Авторизованный абонент вводит сумму из предусловия 3 в строку	Получен код ответа 2xx. Баланс пополнен
2. Отправить SQL-запрос в BRT на проверку изменения баланса	Баланс успешно пополнен

SELECT Phone_number, Balance
FROM USER
WHERE Phone_number = 'номер действующего абонента «Ромашки»'

Тест-кейс №3.2.

Идентификатор: T-UC3-02

Заголовок тест-кейса: Ошибка авторизации абонента (абонент не принадлежит оператору «Ромашка»)

Приоритет: 3

Связь с требованием: №2 Use case «Пополнение баланса абонента оператора «Ромашка»»

Модуль и подмодуль ПО: CRM-система, BRT

Предусловия:

- Запись о данных абонента не существует в БД оператора «Ромашка», номер принадлежит другому оператору.

Шаги тест-кейса:	Ожидаемый результат:
1. Авторизация абонента с номером телефона из предусловия 1	Получен код ответа 4xx. Абонент не принадлежит оператору «Ромашка»

Тест-кейс №3.3.

Идентификатор: T-UC3-03

Заголовок тест-кейса: Ошибка авторизации абонента (номер телефона имеет невалидный вид, несоответствующий маске ввода «+7 (9__)__-__-__»)

Приоритет: 3

Связь с требованием: №2 Use case «Пополнение баланса абонента оператора «Ромашка»»

Модуль и подмодуль ПО: CRM-система, BRT

Предусловия:

- Номер телефона, имеющий вид: 1234567891012345 (или номер телефона, в котором есть буквы или спец.символы (за исключением «+»))

Шаги тест-кейса:	Ожидаемый результат:
1. Авторизация абонента с номером телефона из предусловия 1	Получена ошибка 4xx «Номер телефона введен некорректно»

Тест-кейс №3.4.**Идентификатор:** T-UC3-04**Заголовок тест-кейса:** Пополнение баланса абонента оператора «Ромашка» с некорректной суммой пополнения (отрицательная сумма)**Приоритет:** 3**Связь с требованием:** №2 Use case «Пополнение баланса абонента оператора ‘Ромашка’»**Модуль и подмодуль ПО:** CRM-система, BRT**Предусловия:**

- Имеется учетная запись абонента, зафиксирован номер телефона абонента;
- Исходный баланс определен:
SELECT Phone_number, Balance
FROM USER
WHERE Phone_number = ‘номер действующего абонента «Ромашки»’
- Имеется невалидная сумма для пополнения баланса (например, -300)

Шаги тест-кейса:	Ожидаемый результат:
1. Авторизация абонента с номером телефона из предусловия 1	Получен код ответа 2xx
2. Абонент вводит сумму из предусловия 3	Получен код ответа 4xx. Сумма пополнения не может быть отрицательной.
3. Отправить SQL-запрос в BRT на проверку изменения баланса SELECT Phone_number, Balance FROM USER WHERE Phone_number = ‘номер действующего абонента «Ромашки»’	Баланс не изменен

Тест-кейс №3.5.**Идентификатор:** T-UC3-05**Заголовок тест-кейса:** Пополнение баланса абонента оператора «Ромашка» с некорректной суммой пополнения (нулевая сумма)

Приоритет: 3

Связь с требованием: №2 Use case «Пополнение баланса абонента оператора 'Ромашка'»

Модуль и подмодуль ПО: CRM-система, BRT

Предусловия:

- Имеется учетная запись абонента, зафиксирован номер телефона абонента;
- Исходный баланс определен:
SELECT Phone_number, Balance
FROM USER
WHERE Phone_number = '*номер действующего абонента «Ромашки»*'
- Имеется невалидная сумма для пополнения баланса (0)

Шаги тест-кейса:	Ожидаемый результат:
1. Авторизация абонента с номером телефона из предусловия 1	Получен код ответа 2xx
2. Абонент вводит сумму из предусловия 3	Получен код ответа 2xx. Баланс не изменен.
3. Отправить SQL-запрос в BRT на проверку изменения баланса SELECT Phone_number, Balance FROM USER WHERE Phone_number = ' <i>номер действующего абонента «Ромашки»</i> '	Баланс не изменен

Тест-кейс №3.6.

Идентификатор: T-UC3-06

Заголовок тест-кейса: Пополнение баланса абонента оператора «Ромашка» с некорректной суммой пополнения (в сумме указаны числа или спец.символы (за исключением «. ,»))

Приоритет: 3

Связь с требованием: №2 Use case «Пополнение баланса абонента оператора 'Ромашка'»

Модуль и подмодуль ПО: CRM-система, BRT

Предусловия:

- Имеется учетная запись абонента, зафиксирован номер телефона абонента;
- Исходный баланс определен:
SELECT Phone_number, Balance
FROM USER
WHERE Phone_number = '*номер действующего абонента «Ромашки»*'
- Имеется невалидная сумма для пополнения баланса («триста у.е»)

Шаги тест-кейса:	Ожидаемый результат:
------------------	----------------------

1. Авторизация абонента с номером телефона из предусловия 1	Получен код ответа 2xx
2. Абонент вводит сумму из предусловия 3	Получен код ответа 4xx. Сумма пополнения должна быть числом

№3 Use case:

Название:	Добавление нового абонента оператора «Ромашка».
Триггер:	Менеджер отправляет запрос на добавление нового абонента.
Предусловия:	Новый абонент еще не добавлен в БД оператора «Ромашка». Тариф для нового абонента существует в БД тарифов.
Результат:	Новый абонент успешно добавлен.
Основной поток:	1. Менеджер авторизуется в системе через Postman, указывая логин и пароль. 2. Postman проверяет данные о менеджере и успешно авторизует его в системе. 3. Менеджер отправляет запрос на добавление нового абонента через Postman, указывая номер телефона и идентификатор тарифа. 4. Postman передает запрос на создание нового абонента CRM. 5. CRM, получая запрос, формирует информацию о новом абоненте. 6. CRM передает информацию в БД абонентов оператора «Ромашка». 7. БД проводит операцию добавления нового абонента.
Альтернативные потоки:	2.а Ошибка авторизации менеджера 2.а.1 Возвращается ошибка. 3.а Ошибка отправки запроса на добавление нового абонента 3.а.1 Возвращается ошибка. 4.а Ошибка передачи запроса на добавление нового абонента 4.а.1 Возвращается ошибка. 6.а Ошибка отправки запроса с информацией о новом абоненте 6.а.1 Возвращается ошибка. 7.а Ошибка добавления нового абонента 7.а.1 Возвращается ошибка.

Тест-кейс №4.1.

Идентификатор: T-UC4-01

Заголовок тест-кейса: Добавление нового абонента оператора «Ромашка».

Приоритет: 2

Связь с требованием: №3 Use case «Добавление нового абонента оператора «Ромашка»»

Модуль и подмодуль ПО: CRM-система, HRS, BRT

Предусловия:

- Имеется учетная запись роли менеджер;
- Известен логин и пароль учетной записи роли менеджер;
- Абонент еще не существует в БД оператора «Ромашка», зафиксировать номер телефона абонента;
 SELECT Phone_number, ID_tariff
 FROM USER
 WHERE Phone_number = 'номер действующего абонента «Ромашки»';
 Результат: 0 совпадений
- Тарифный план существует в БД;
 SELECT ID_tariff
 FROM TARIFF;
- Авторизоваться с учётной записью из условия 1,2

Шаги тест-кейса:	Ожидаемый результат:
1. Отправить запрос на добавление нового абонента, учитывая успешное выполнение последнего предусловия <i>Входные данные:</i> номер телефона абонента, id тарифа (из предусловия 3,4)	Получен код ответа 2xx
2. Проверить данные в ответе	Получен код ответа 2xx
3. Отправить SQL-запрос в BRT	Абонент существует в БД

SELECT Phone_number, ID_tariff
 FROM USER
 WHERE Phone_number = 'номер действующего абонента «Ромашки»';
 Результат: 1 совпадение

Тест-кейс №4.2.

Идентификатор: T-UC4-02

Заголовок тест-кейса: Ошибка при добавлении нового абонента оператора «Ромашка» из-за невалидного номера телефона.

Приоритет: 3

Связь с требованием: №3 Use case «Добавление нового абонента оператора «Ромашка»»

Модуль и подмодуль ПО: CRM-система, HRS, BRT

Предусловия:

- Имеется учетная запись роли менеджер;
- Известен логин и пароль учетной записи роли менеджер;
- Номер телефона, имеющий вид: 1234567891012345 (или номер телефона, в котором есть буквы или спец.символы (за исключением «+»))
- Абонент еще не существует в БД оператора «Ромашка», зафиксировать номер телефона абонента;
 SELECT Phone_number, ID_tariff
 FROM USER
 WHERE Phone_number = 'номер действующего абонента «Ромашки»';
 Результат: 0 совпадений

- Тарифный план существует в БД;
SELECT ID_tariff
FROM TARIFF;
- Авторизоваться с учётной записью из условия 1,2

Шаги тест-кейса:	Ожидаемый результат:
1. Отправить запрос на добавление нового абонента, учитывая успешное выполнение последнего предусловия <i>Входные данные:</i> номер телефона абонента, id тарифа (из предусловия 3,5)	Получен код ответа 4xx. «Номер телефона введен некорректно»

Тест-кейс №4.3.

Идентификатор: T-UC4-03

Заголовок тест-кейса: Добавление уже существующего абонента оператора «Ромашка» в качестве нового абонента.

Приоритет: 3

Связь с требованием: №3 Use case «Добавление нового абонента оператора ‘Ромашка’»

Модуль и подмодуль ПО: CRM-система, HRS, BRT

Предусловия:

- Имеется учетная запись роли менеджер;
- Известен логин и пароль учетной записи роли менеджер;
- Абонент существует в БД оператора «Ромашка», зафиксировать номер телефона абонента;
SELECT Phone_number, ID_tariff
FROM USER
WHERE Phone_number = 'номер действующего абонента «Ромашки»';
Результат: 1 совпадение
- Тарифный план существует в БД;
SELECT ID_tariff
FROM TARIFF;
- Авторизоваться с учётной записью из условия 1,2

Шаги тест-кейса:	Ожидаемый результат:
1. Отправить запрос на добавление нового абонента, учитывая успешное выполнение последнего предусловия <i>Входные данные:</i> номер телефона абонента, id тарифа (из предусловия 3,4)	Получен код ответа 4xx. Абонент уже имеется в БД