| **Отчет по тестированию телекоммуникационной биллинговой системы "Ромашка"** |
| --- |
| **История редактирования**   | Версия | Описание | Автор | Дата | | --- | --- | --- | --- | | 1.0 | Первый отчет по тестированию | Михайлова Алина | 11.05.2025 | |

# Итог тестирования

В ходе тестирования были выявлены 7 дефектов различной степени критичности:

| ID дефекта | Название | Критичность | Статус |
| --- | --- | --- | --- |
| #DEF-001 | Тестирование API: Запрос от менеджера на добавление абонента с некорректным номером телефона msisdn | Высокая | Открыт |
| #DEF-002 | Тестирование API: Запрос от менеджера на добавление абонента другого оператора | Высокая | Открыт |
| #DEF-003 | Тестирование API: Запрос от менеджера на добавление абонента с некорректным тарифом | Высокая | В работе |
| #DEF-004 | Тестирование API: Запрос от менеджера на добавление абонента с суммой пополнения 0 | Высокая | В работе |
| #DEF-005 | Автотест: на отправку CDR файлов на валидацию в BRT через брокер RabbitMQ | Средняя | Закрыт |
| #DEF-006 | Тест-кейс: #HRS-TC-006 Тарификация с несуществующим ID тарифа | Высокая | Закрыт |
| #DEF-007 | Тест-кейс: #BRT-TC-012 Абонент с таким ФИО и паспортными данными уже существует | Высокая | Закрыт |

Наиболее проблемными областями являются:

* Обработка внешних данных (абоненты другого оператора);
* Отсутствие проверки на уникальность, на существование данных;
* Валидация входных данных(CDR файлы, пополнение баланса на 0).

Рекомендации по исправлениям:

1. Приоритет для дефектов:
   1. DEF-001 (некорректный номер), DEF-002 (чужой оператор), DEF-007 (дублирование абонентов) - требуют исправления в первую очередь, так как влияют на целостность данных.
   2. DEF-003 (несуществующий тариф), DEF-004 (пополнение 0) - важны для корректной бизнес-логики.
2. Доработка валидации:
   1. Добавить проверку формата номера (msisdn), уникальности данных абонентов, существования тарифов в БД;
   2. Реализовать обработку ошибок для возврата корректных HTTP-статусов (400 вместо 200 при невалидных данных, 404 при не нахождении данных);

Также стоит учесть момент с нагрузочным тестированием - его нужно запланировать, чтобы проверить стабильность работы проекта.

# Результаты тестирования

В данной таблице представлены результаты тестирования (таблица 2).

| Категория | Выполнено тестов | Успешно | Провалено | Пропущено |
| --- | --- | --- | --- | --- |
| Тест-кейсы по файлу «Тест-кейсы UC.xlsx» | 34 | 28 | 6 | 0 |
| Сквозные кейсы «Ручные тесты по ТЗ.xlsx» | 6 | 4 | 4 | 0 |
| API (Postman) | 26 | 22 | 4 | 0 |
| Автотесты | 34 | 26 | 8 | 0 |
| Чек-листы по «Ручные тесты по ТЗ.xlsx» | 13 | 8 | 5 | 0 |

Таблица 2 – Результаты тестирования проекта

Тем самым, всего было проведено 113 тестов, успешных 88 теста – 78%, проваленных 25 – 22%.

Основной функционал микросервисов работает корректно, ключевые бизнес-сценарии успешно пройдены, но перед релизом стоит устранить дефекты, особенно при негативных сценариях(некорректный номер телефона, несуществующие тарифы).

# Найденные дефекты

Логика тестов повторялась, поэтому дефекты, найденные при ручных тестах, были схожи с дефектами, найденные при автотестах или тестирования API.

| DEF-001 | Тестирование API: Запрос от менеджера на добавление абонента с некорректным номером телефона msisdn | | | |
| --- | --- | --- | --- | --- |
| Тип запроса (Error) | Приоритет | | 1 | POST {{BASE\_URL}}/crm/manager/subscriber |
| Шаги воспроизведения | | | | |
| 1 | Отправка запроса body с генерированными данными, где номер изначально некорректный | const firstName = pm.variables.replaceIn("{{$randomFirstName}}");  const surname = pm.variables.replaceIn("{{$randomLastName}}");  const msisdn = '79' + Math.floor(Math.random() \* 90000 + 10000);  pm.variables.set('firstName', firstName);  pm.variables.set('surname', surname);  pm.variables.set('msisdn', msisdn); | | |
| 2 | Получение ответа от сервера | {      "id": 32,      "msisdn": "7969537",      "firstName": "Tevin",      "secondName": null,      "surname": "Schuster",      "tariffId": null,      "balance": 100  } | | |
| Замечание | | Ожидалось: «Статус 400 - Формат msisdn введен некорректно» (Invalid Format)  Было получено: Статус 200 OK | | |

| DEF-002 | Тестирование API: Запрос от менеджера на добавление абонента другого оператора | | | |
| --- | --- | --- | --- | --- |
| Тип запроса (Error) | Приоритет | | 1 | POST {{BASE\_URL}}/crm/manager/subscriber |
| Шаги воспроизведения | | | | |
| 1 | Отправка запроса body с генерированными данными, где номер изначально принадлежит другому оператору | const firstName = pm.variables.replaceIn("{{$randomFirstName}}");  const surname = pm.variables.replaceIn("{{$randomLastName}}");  const msisdn = '70' + Math.floor(Math.random() \* 900000000 + 100000000);  pm.variables.set('firstName', firstName);  pm.variables.set('surname', surname);  pm.variables.set('msisdn', msisdn); | | |
| 2 | Получение ответа от сервера | {      "id": 33,      "msisdn": "70609237266",      "firstName": "Charlene",      "secondName": null,      "surname": "Kerluke",      "tariffId": null,      "balance": 100  } | | |
| Замечание | | Ожидалось: «Статус 400 - msisdn не является клиентом опретора 'Ромашка'»  (Invalid Format)  Было получено: Статус 200 OK | | |

| DEF-003 | Тестирование API: Запрос от менеджера на добавление абонента с некорректным тарифом | | | |
| --- | --- | --- | --- | --- |
| Тип запроса (Error) | Приоритет | | 2 | POST {{BASE\_URL}}/crm/manager/subscriber |
| Шаги воспроизведения | | | | |
| 1 | Отправка запроса body с генерированными данными, где такого тарифа не существует | pm.variables.set('tariffId', 3); | | |
| 2 | Получение ответа от сервера | {      "status": 400,      "exceptionType": "BAD\_REQUEST",      "message": "No such active tariff.",      "url": "http://172.18.0.10:8084/tariffs/3"  } | | |
| Замечание | | Ожидалось: «Статус 404 - Тариф не найден»  (Not Found)  Было получено: Статус 400 Bad Request | | |

| DEF-004 | Тестирование API: Запрос от менеджера на добавление абонента с суммой пополнения 0 | | | |
| --- | --- | --- | --- | --- |
| Тип запроса (Error) | Приоритет | | 2 | POST {{BASE\_URL}}/crm/manager/subscriber |
| Шаги воспроизведения | | | | |
| 1 | Отправка запроса body с генерированными данными, где сумма пополнения равна 0 | pm.variables.set('balance', 0); | | |
| 2 | Получение ответа от сервера | "id": 34,      "msisdn": "79631018787",      "firstName": "Elizabeth",      "secondName": null,      "surname": "Rohan",      "tariffId": null,      "balance": 0  } | | |
| Замечание | | Ожидалось: «Статус 400 - Сумма пополнения должна быть положительной, больше 0»  Было получено: Статус 200 ОК | | |

| DEF-005 | Автотест: на отправку CDR файлов на валидацию в BRT через брокер RabbitMQ | | | |
| --- | --- | --- | --- | --- |
| Тип запроса (Error) | Приоритет | | 1 | "cdr\_invalid\_msisdn.txt" |
| Шаги воспроизведения | | | | |
| 1 | Отправка в очередь RabbitMQ файл "cdr\_invalid\_msisdn.txt" |  | | |
| 2 | Получение файла BRT сервисом | Файл прошел проверки | | |
| Замечание | | Ожидалось отклонение файла сервисом BRT | | |

| DEF-006 | Тест-кейс: #HRS-TC-006 Тарификация с несуществующим ID тарифа | | | |
| --- | --- | --- | --- | --- |
| Тип запроса (Error) | Приоритет | | 1 | Нет данных о тарифе с указанным ID в БД |
| Шаги воспроизведения | | | | |
| 1 | Отправить JSON с несуществующим id\_tariff | Сервис возвращает статус 200 ОК | | |
| Замечание | | Ожидалось ответ 404 Not Found | | |

| DEF-007 | Тест-кейс: #BRT-TC-012 Абонент с таким ФИО и паспортными данными уже существует | | | |
| --- | --- | --- | --- | --- |
| Тип запроса (Error) | Приоритет | | 2 | Абонент с такими же данными (ФИО, паспортные данные) уже существует в таблице Subscribers |
| Шаги воспроизведения | | | | |
| 1 | Менеджер предоставляет данные абонента, который уже существует | Произошел запрос с CRM на BRT | | |
| 2 | Запоминание id\_subscriber существующего абонента | BRT отправляет запрос об успешном добавлении абонента | | |
| Замечание | | Ожидалось отклонение запроса.  Было получено: Статус 200 ОК | | |

# Приложение

* Полный список тест-кейсов [<https://docs.google.com/spreadsheets/d/1Xf6a9Kgi7x2SmerEpD49QqqmW8_nD1Ubxse07ykf2j8/edit?usp=sharing>]
* Ручные тесты [<https://docs.google.com/spreadsheets/d/1MPNN_SiL94LIUN5Fmq5z3I0uhovFxXrO7UML9CjQ6Vk/edit?usp=sharing>]
* Коллекция Postman:

[<https://drive.google.com/file/d/1wRuzKj19TIXuV3oCBlLHaYkBvxBZxd-_/view?usp=sharing>]

* Репозиторий с тестированием: [<https://github.com/ove4lo/NexignBootcampTesting>]
* Репозиторий со старыми автотестами:

[<https://github.com/ove4lo/NexignBootcamp>]

* Документация API (Swagger):

[<https://drive.google.com/file/d/1hcNnetLLc6a7u_KRHNzn78LVdWTx0qvW/view?usp=sharing>]

# Исходные требования

**10.1 Интеллект-карта** [<https://miro.com/app/board/uXjVIDVUO2M=/?share_link_id=110641664340>]

## 10.2 ТЗ

* + CDR (ТЗ от аналитика):

[<https://drive.google.com/file/d/1B3mrqJTl9jCXTyhcTWf6udOJ9wiMn1LZ/view?usp=sharing>]

* + BRT (ТЗ от аналитика):

[<https://drive.google.com/file/d/1fAkAdKqmCYhSehRCxo7VN-5LGLuiR87P/view?usp=sharing>]

* + CRM (ТЗ от аналитика):

[<https://drive.google.com/file/d/1X1nuzhTR5w89ihh_SiHd8ciZ2nSj9iYP/view?usp=sharing>]

* + HRS (ТЗ от аналитика):

[<https://drive.google.com/file/d/1H3zN1NhnzB8wz6jYkFHrd_njziQQt8JZ/view?usp=sharing>]