| **План тестирования телекоммуникационной биллинговой системы "Ромашка"** |
| --- |
| **История редактирования**   | Версия | Описание | Автор | Дата | | --- | --- | --- | --- | | 1.0 | Первое описание плана тестирования | Михайлова Алина | 11.05.2025 | |

# Введение

## Основная информация

Проект представляет собой распределенную микросервисную систему биллинга для телекоммуникационной компании "Ромашка". Система обрабатывает данные о вызовах абонентов, тарифицирует их согласно тарифным планам и производит расчеты.

## Цель тестирования

Тест-план проекта преследует следующие цели:

* Проверить корректность работы каждого микросервиса;
* Убедиться в правильности взаимодействия между сервисами;
* Проверить точность расчетов тарификации;
* Обеспечить стабильность работы системы под нагрузкой;
* Подтвердить соответствие бизнес-требованиям.

# Рамки проекта

## 2.1 Область тестирования

В объем работ по тестированию входит тестирование следующих микросервисов и их функций:

1. **CDR-сервис**:
   * Генерация записей о вызовах;
   * Отправка пакетов CDR в BRT-сервис через RabbitMQ.
2. **BRT-сервис**:
   * Обработка вызовов в реальном времени;
   * Управление балансами абонентов;
   * Взаимодействие с HRS-сервисом.
3. **HRS-сервис**:
   * Применение правил тарификации;
   * Управление тарифными планами и пакетами;
   * Расчет стоимости услуг.
4. **CRM-сервис**:
   * Авторизация в роли менеджера: смена тарифа и пополнение баланса любого абонента, создание нового абонента, запрос на доступную информацию по абоненту и его текущему тарифу;
   * Авторизация в роли абонента: может пополнить счет.

## 2.2 Ключевые бизнес-процессы:

* Обработка входящих/исходящих звонков;
* Смена тарифного плана;
* Автоматическое продление тарифного периода;
* Начисление абонентской платы;
* Учет пакетных минут.

## 2.3 Исключения

* Тестирование безопасности;
* UI-тестирование;
* Инфраструктурные сервисы (Config Server, Eureka Server, API Gateaway).

# Критерии входа и выхода

## 3.1 Критерии входа (Entry Criteria)

* Развернуто тестовое окружение (Docker-compose);
* Заполнены тестовые данные (тарифы, абоненты);
* Доступны все микросервисы;
* Настроен мониторинг RabbitMQ;
* Доступны инструменты тестирования.

## 3.2 Критерии выхода (Exit Criteria)

* 100% прохождение критических тест-кейсов;
* Все блокирующие дефекты исправлены;
* Автоматизированные тесты покрывают ≥80% основных сценариев;
* Достигнуто согласованное покрытие тестами (по метрикам Allure).

# Ресурсы

## 4.1 Инструменты

* Автоматизация: Java 17 + RestAssured + JUnit5;
* Ручное тестирование: Postman;
* Мониторинг**:** RabbitMQ Management, Spring Boot Admin.

## 4.2 Окружение

* Тестовый стенд с Docker-контейнерами;
* Отдельные БД для каждого сервиса;
* Тестовый экземпляр RabbitMQ;
* Mock-сервисы для внешних интеграций.

# План и стратегия тестирования

## 6.1 По типам

1. **Интеграционное тестирование**:
   * Взаимодействие между CDR → BRT → HRS → CRM;
   * Проверка работы через RabbitMQ;
   * Тестирование API CRM.
2. **Функциональное тестирование**:
   * Проверка правильности тарификации;
   * Тестирование смены тарифных планов:
   * Проверка расчета стоимости с учетом пакетных минут.
3. **Нагрузочное тестирование**:
   * Обработка большого количества CDR-записей;
   * Параллельная обработка вызовов.
4. **API-тестирование**:
   * Тестирование REST API каждого микросервиса;
   * Проверка валидации входных данных;
   * Тестирование ошибок и граничных условий.

## 6.2 По подходам

* Автоматизированное тестирование: API-тесты (RestAssured), интеграционные тесты;
* Ручное тестирование: Исследовательское тестирование, проверка сложных сценариев.

# Детализация тестового покрытия

## 7.1 Тестирование по микросервисам (на основе "Тест-кейсы UC.xlsx")

CDR-сервис:

* #CDR-TC-001: Успешная генерация CDR (приоритет 1.0);
* #CDR-TC-002: Ошибка доступа к БД (приоритет 3.0);
* #CDR-TC-003: Звонок через полночь (приоритет 2.0).

HRS-сервис:

* #HRS-TC-001/002: Тарификация для "Классики" (исходящие/входящие);
* #HRS-TC-003/004: Тарификация для "Помесячного" (в рамках лимита/превышение);
* #HRS-TC-005: Отрицательная длительность звонка
* #HRS-TC-006: Несуществующий тариф;
* #HRS-TC-007: Ошибка БД;
* #HRS-TC-008: Некорректный JSON.

BRT-сервис:

* #BRT-TC-001: Успешная обработка CDR;
* #BRT-TC-002/003: Обработка некорректных CDR;
* #BRT-TC-005/006: Списание средств (успешное/ошибки);
* #BRT-TC-007/008: Пополнение баланса;
* #BRT-TC-011/012: Добавление абонента;
* #BRT-TC-013/014: Смена тарифа.

CRM-сервис:

* #CRM-TC-001/002: Авторизация менеджера;
* #CRM-TC-004/005: Авторизация абонента;
* #CRM-TC-006: Пополнение баланса абонентом.

## 7.2 Сквозное тестирование (на основе "Ручные тесты по ТЗ.xlsx")

Ключевые сценарии:

* CHB-TC-001: Исходящий звонок ("Классика", внутри сети);
* CHB-TC-002: Исходящий звонок ("Классика", внешняя сеть);
* CHB-TC-004: Превышение лимита ("Помесячный");
* CHB-TC-005: Звонок в рамках лимита ("Помесячный").

Валидация данных:

* Проверка формата номеров;
* Корректность временных меток;
* Обработка пустых файлов.

## 7.3 API тестирование (Postman)

Менеджер:

* Создание/изменение абонентов;
* Смена тарифов;
* Пополнение баланса;
* Просмотр информации.

Абонент:

* Пополнение собственного баланса;
* Просмотр информации.

# Метрики успеха

* Покрытие основных сценариев ≥75%;
* Успешное прохождение всех критических тест-кейсов;
* Отсутствие блокирующих дефектов в финальном отчете.

# Риски и ограничения

В данной таблице представлены возможные риски при тестировании (см. таблица 1):

| Риск | Вероятность | Влияние | Митригация |
| --- | --- | --- | --- |
| Нестабильность тестового окружения | Средняя | Высокое | Резервные копии конфигураций |
| Нехватка тестовых данных | Низкая | Среднее | Скрипты генерации тестовых данных |
| Проблемы с производительностью RabbitMQ | Средняя | Высокое | Мониторинг очередей, алерты |
| Недостаток времени на полное тестирование | Высокая | Критическое | Приоритезация тест-кейсов |

Таблица 1 – Риски и ограничения

# Приложения

* 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=287012551893>]

## 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>]