

# Биллинг

- Биллинг как ключевой элемент любого бизнеса.
- Большинство биллинговых систем сложны в использовании за счет универсальности.
- Определение конкретных сценариев для задач онлайн-кинотеатра позволяет реализовать простой и надежный механизм функционирования сервиса.

## Функциональные требования

- Оплата подписки через банковские карты
- Возврат денежных средств
- Контроль оплаты пользователей

## Нефункциональные требования

- Гарантия выполнения оплаты
- Исключение двойных списаний

# Технический стек

- Django
- PostgreSQL
- FastAPI
- Flask
- Stripe
- Docker
- Pytest
- Allure

# Планирование работ

## 1. Неделя

- Изучение предмета
- Составление пользовательских сценариев
- Разработка архитектуры и диаграмм движения

## 2. Неделя

- Продолжение изучения предмета
- Корректировка архитектуры
- Реализация структуры сервисов

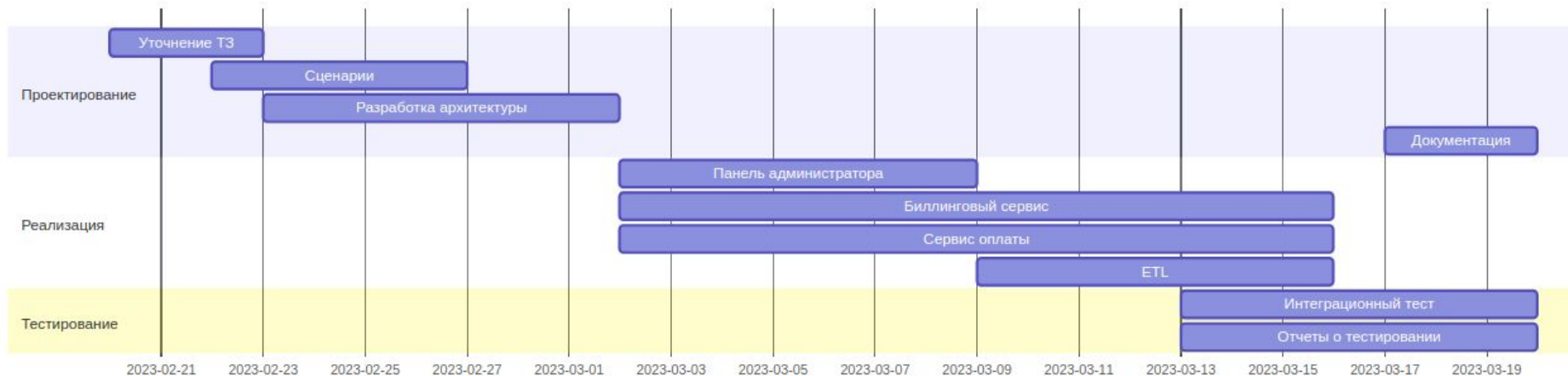
## 3. Неделя

- Реализация сервисов

## 4. Неделя

- Разработка интеграционных тестов
- Отладка взаимодействия сервисов
- Правка документации проекта

# Диаграмма Ганта



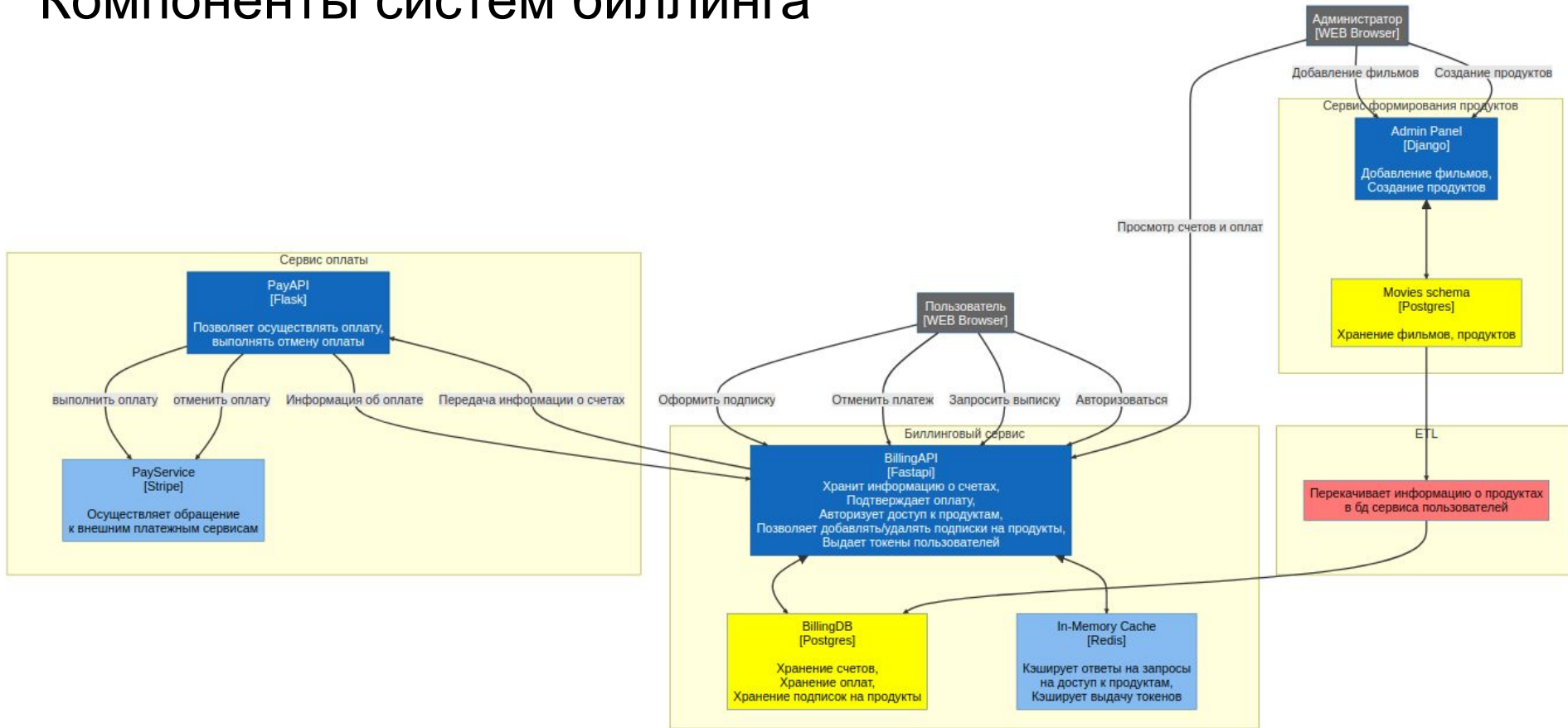
# Компоненты систем биллинга

- Панель администратора
- Биллинговый сервис
- Сервис оплаты
- ETL

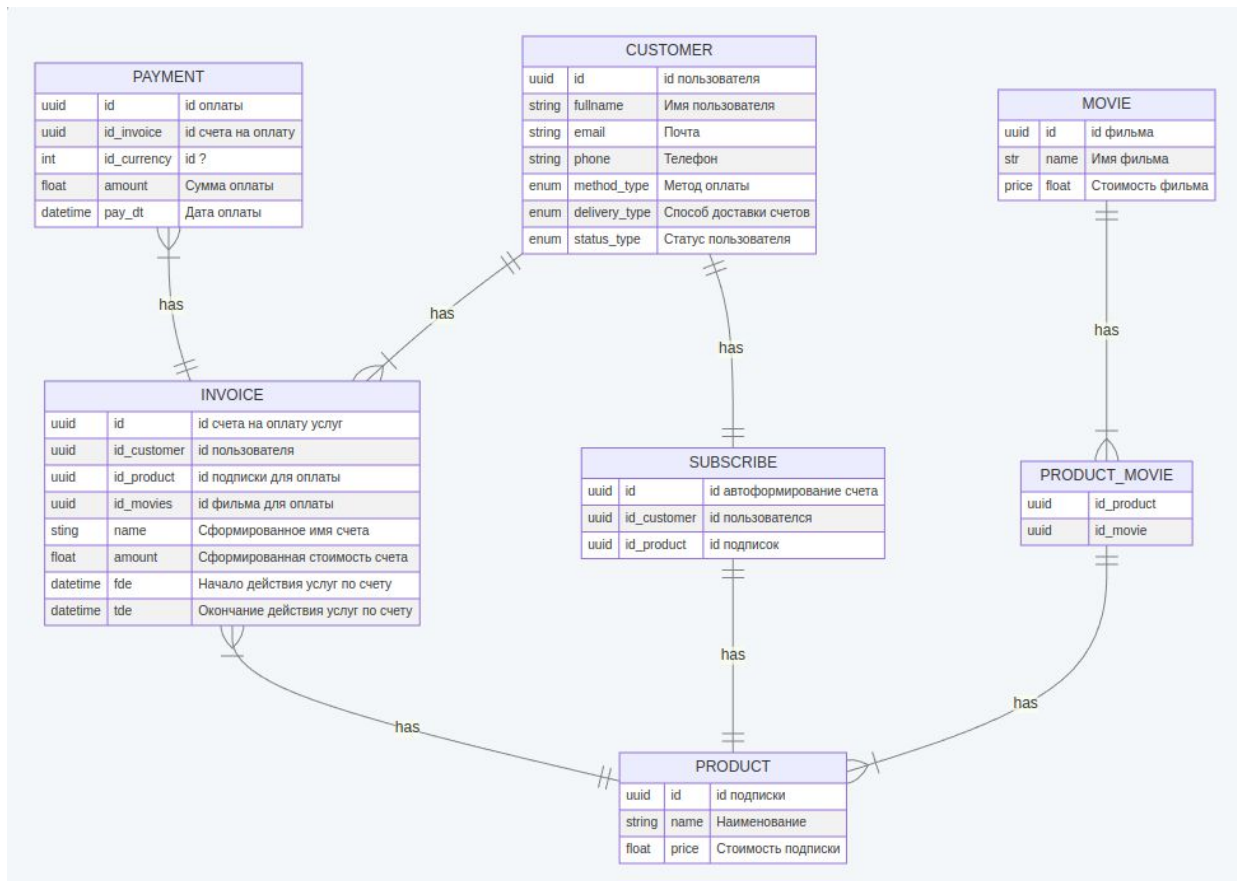
## Реализованный функционал

- Внесение подписок менеджером онлайн-кинотеатра
- Конвертирование данных о доступных подписках в биллинговый сервис
- Оплата/отмена оплаты подписки пользователем
- Просмотр оплат

# Компоненты систем биллинга

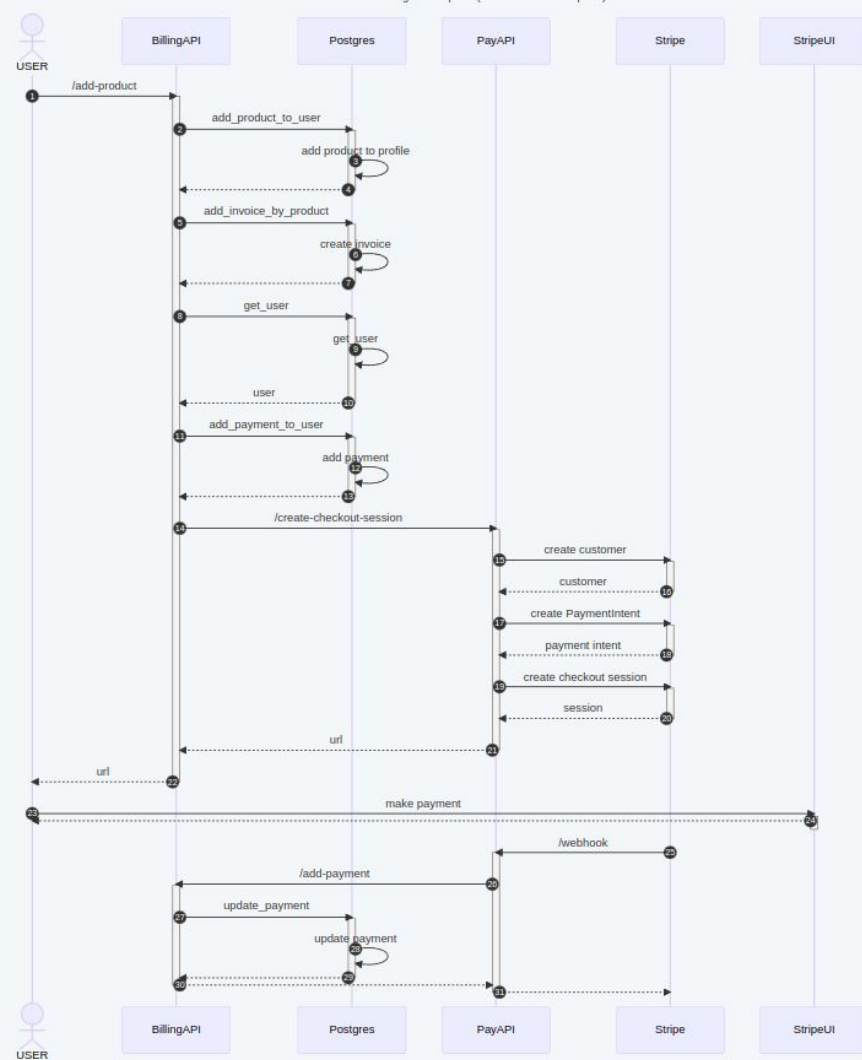


# Организация хранилища

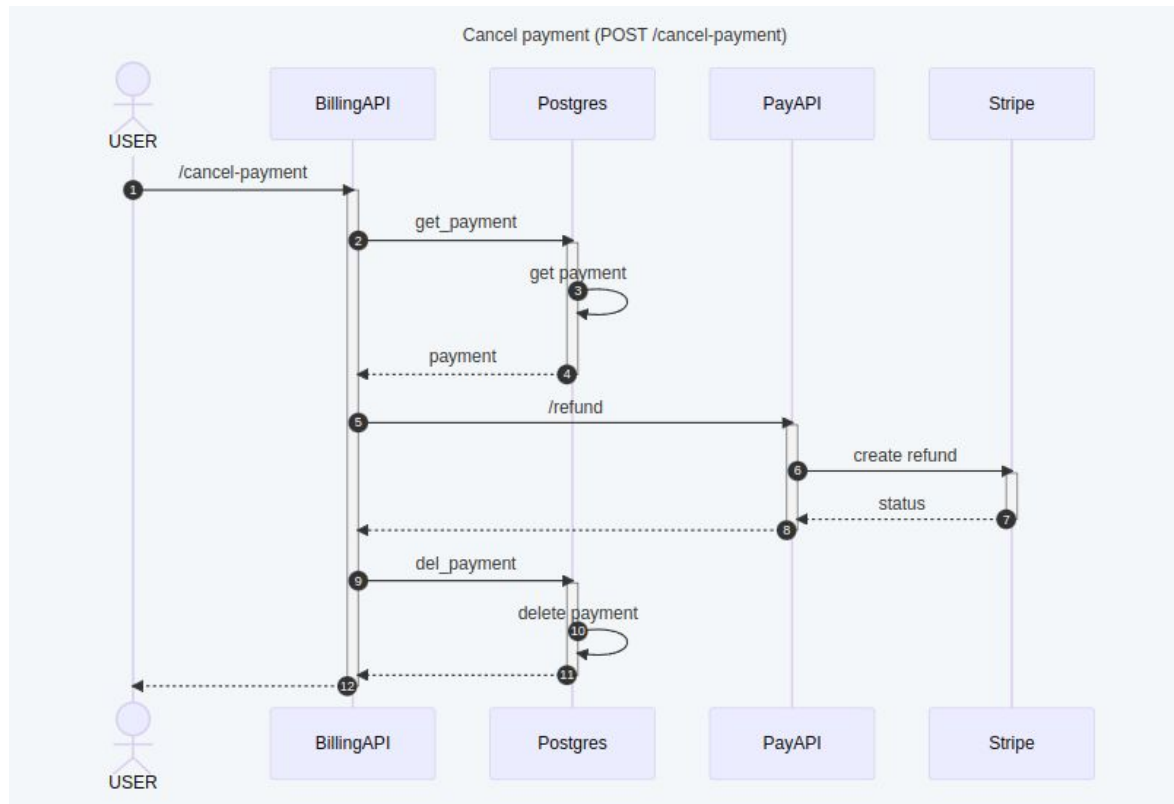






















































# Диаграмма последовательности проведения оплаты



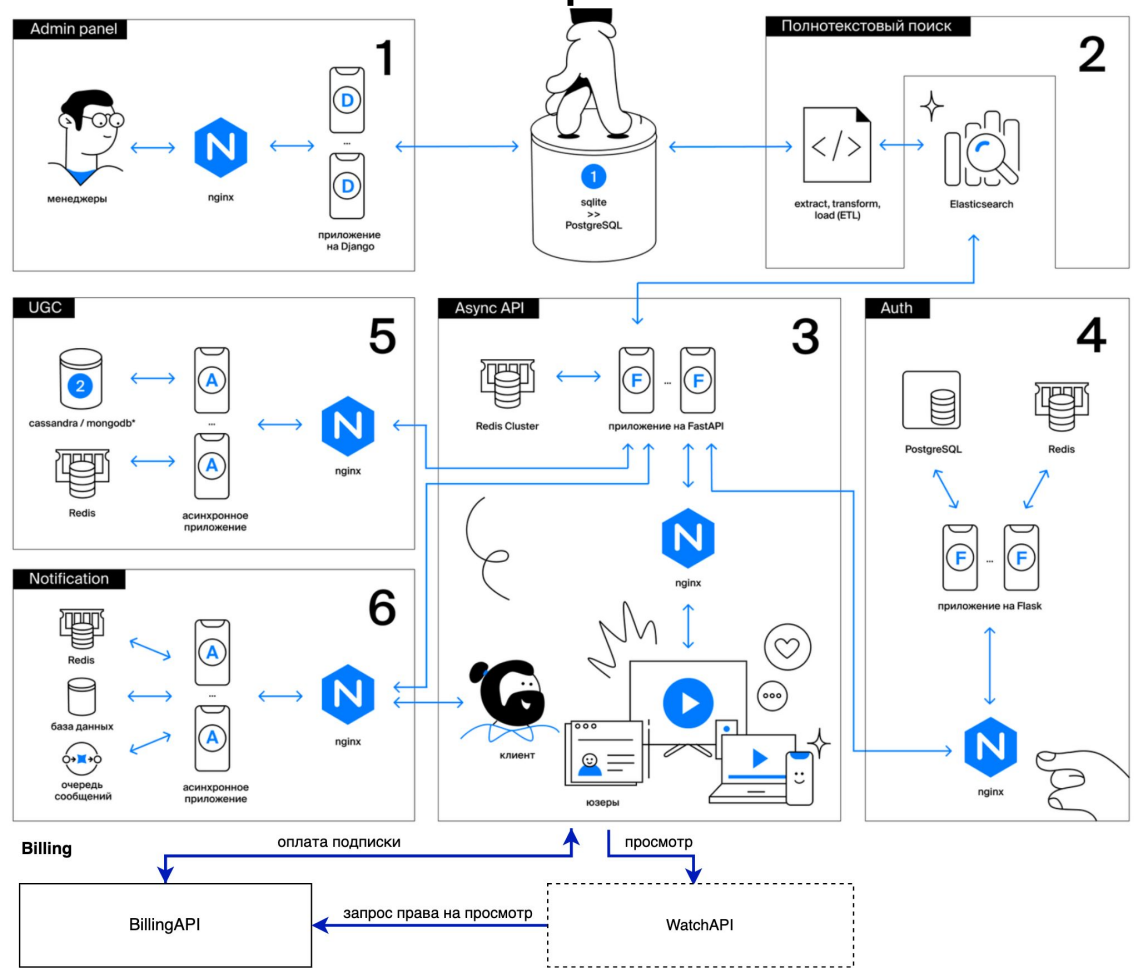
# Диаграмма последовательности отмены оплаты



# Docker-контейнеры

<input type="checkbox"/> Name ↓↑	State ↓↑ Filter ▾	Quick Actions	Stack ↓↑	Image ↓↑
<input type="checkbox"/> portainer	running	    	-	portainer/portainer-ce
<input type="checkbox"/> django	running	    	graduate_work	graduate_work-admin
<input type="checkbox"/> integration_test	running	    	graduate_work	integration_test
<input type="checkbox"/> etl	running	    	graduate_work	etl
<input type="checkbox"/> billing	running	    	graduate_work	billing
<input type="checkbox"/> postgres_container_test	running	    	graduate_work	postgres:13
<input type="checkbox"/> redis	running	    	graduate_work	bitnami/redis:latest
<input type="checkbox"/> pay-api	running	    	graduate_work	payapi
<input type="checkbox"/> allure	healthy	    	graduate_work	frankescobar/allure-docker-service:latest
<input type="checkbox"/> stripe-cli	running	    	graduate_work	stripe/stripe-cli

# Интеграция с онлайн-кинотеатром



# Демо

Проверка работы сервиса по типовому сценарию пользователя:

- Заполнение базы данных тестовыми данными - фильмы и подписка
- Авторизация пользователя для получения токена для дальнейшего взаимодействия с сервисом
- Получение списка продуктов, на которые можно оформить подписку
- Проверка отсутствия доступа к контенту
- Оформление подписки и получение ссылки на оплату
- Ожидание оплаты
- Проверка успешного доступа к контенту
- Отмена оплаты за подписку
- Проверка отсутствия доступа к контенту

# Интеграционный тест

```
@allure.suite("Интеграционные тесты")
@allure.title("Проверка оплаты пользователем")
def test_1(db_session, initial_data, billing_client, send_telegram_notify):
    product, movies = initial_data

    with step("Получение списка доступных продуктов"):...

    with step("Проверка отсутствия прав у пользователя на просмотр фильма"):...

    with step("Запрос пользователем на покупку подписки"):...

    with step("Проверка отсутствия прав у пользователя на просмотр фильма после оформления подписки"):...

    with step("Ожидание оплаты подписки пользователем в течении 2-х минут"):...

    with step("Проверка успешного доступа к контенту после оформления подписки"):...

    with step("Отмена платежа"):...

    with step("Проверка отсутствия прав у пользователя на просмотр фильма"):...
```

# Проведение тестирования

```
Attaching to integration_test
integration_test | ===== test session starts =====
integration_test | platform linux -- Python 3.10.8, pytest-7.2.2, pluggy-1.0.0
integration_test | rootdir: /opt/app
integration_test | plugins: allure-pytest-2.13.0, anyio-3.6.2
integration_test | collected 1 item
integration_test |
integration_test | tests/test_1.py . [100%]
integration_test |
integration_test | ===== 1 passed in 66.76s (0:01:06) =====
integration_test exited with code 0
```

# Проведение оплаты

←  Test **TEST MODE**

Фильмы для детей

**RUB 1,000.00**

Powered by **stripe** | [Terms](#) [Privacy](#)

## Pay with card

Email mail@example.com

### Card information

4242 4242 4242 4242

VISA

04 / 28

777



### Name on card

IVAN PETROV

### Country or region

Russia



Pay



By confirming your payment, you allow Test to charge your card for this payment and future payments in accordance with their terms.



# Отчет о тестировании

Overview

Categories

Suites

Graphs

Timeline

Behaviors

Packages

Suites

order name duration status

Status: 0 0 1 0 0 Marks: [icons]

tests

Интеграционные тесты

#1 Проверка оплаты пользователем 1m 04s

tests.test\_1#test\_1

Passed Проверка оплаты пользователем

Overview History Retries

Severity: normal

Duration: 1m 04s

Execution

Set up

Подключение к API Биллинг сервиса 2s 078ms

Подключение к БД 1ms

Заполнение БД тестовыми данными 29ms

Подключения к телеграм 0s

Test body

Получение списка доступных продуктов 131ms

Проверка отсутствия прав у пользователя на просмотр фильма 209ms

Запрос пользователем на покупку подписки 1s 994ms

Проверка отсутствия прав у пользователя на просмотр фильма после оформления подписки 223ms

Ожидание оплаты подписки пользователем в течении 2-х минут 1m 00s

Проверка успешного доступа к контенту после оформления подписки 16ms

Отмена платежа 1s 461ms

Проверка отсутствия прав у пользователя на просмотр фильма 114ms

log 5.8 KB

stdout 2.5 KB

stderr 6.1 KB

Tear down

## Что не успели

- Система лояльности (Индивидуальные скидки и промокоды)
- Рекуррентные платежи
- Интеграция тестов в CI