**Документация, содержащая описание функциональных характеристик модуля программного обеспечения**

**Модуль «Telegram бот»**

**г. Москва 2025**

**Содержание**

[1. Назначение системы 3](#_heading=h.u8baoeuj932l)

[2. Функциональные требования 3](#_heading=h.4mr2pyaam4qn)

[3. Архитектура Telegram бота 3](#_heading=h.616p9t2xihav)

[4. Схема работы 3](#_heading=h.tu49x9fx2dcs)

[5. Используемые языки программирования и фреймворки 4](#_heading=h.f1km0ndxyaxb)

[Реализация 4](#_heading=h.hvcmxrhql0p)

[6. Последовательность потоков данных 4](#_heading=h.l03dhxvdlbq6)

[7. Описание алгоритмов и математического обеспечения 4](#_heading=h.xqpcx5yilvap)

[8. Логическая модель данных и справочники 4](#_heading=h.tavnv3sj88j9)

[9. Роли 5](#_heading=h.7tissdisr173)

## 1. Назначение модуля «Telegram бот»

Модуль «Telegram бот» предназначен для предоставления актуальных отчетов о состоянии бизнеса онлайн-кинотеатра «Премьер» в интерфейсе мессенджера Telegram. Он обеспечивает удобный доступ к аналитическим данным менеджменту всех уровней.

## 2. Функциональные требования

Модуль «Telegram бот» обеспечивает удобную интеграцию бизнес-аналитики с Telegram, предоставляя оперативные отчеты менеджменту. Для обеспечения надежной работы рекомендуется проводить регулярное тестирование и мониторинг процессов передачи данных.

Модуль выполняет следующие функции:

* Получение актуальных и исторических данных по оплатам подписок (биллингу), смотрению, посещению и действиям пользователей.
* Получение актуальных и исторических метаданных по контенту.
* Хранение данных в базе.
* Агрегация данных для формирования отчетов.
* Отправка агрегированных отчетов через Telegram бот.
* Управление пользователями, имеющими доступ к информации в боте.

## 3. Архитектура Telegram бота

Модуль Telegram бота включает следующие компоненты:

Сервер, размещенный на виртуальной машине в Yandex Cloud.

Мессенджер Telegram, через который пользователи получают отчеты.

База данных ClickHouse, используемая для хранения и обработки данных.

Платформа данных, отвечающая за сбор и подготовку данных.



*Рис.1 Схема работы Модуля*

## 

## 4. Схема работы

Данные из «Платформы данных» загружаются в ClickHouse.

По расписанию или по запросу агрегированные данные подготавливаются для отправки.

Telegram бот получает промпты и данные для формирования отчетов.

Бот отправляет сообщения пользователям согласно их ролям и доступам.

## 5. Используемые языки программирования и фреймворки

### Реализация

Язык программирования: Python

Используемые библиотеки:

* + Pandas
  + Telebot
  + Telegram Bot calendar

## 6. Последовательность потоков данных

Ежедневное обновление витрин в ClickHouse.

Формирование агрегатов и структурированных отчетов.

Передача данных Telegram боту для отправки пользователям.

## 7. Описание алгоритмов и математического обеспечения

Используются методы агрегации, оконных сдвигов, объединений и соединений таблиц.

Все вычисления происходят на стороне «Платформы данных».

## 8. Логическая модель данных и справочники

episode\_views\_by\_viewer

Основные поля:

dt – дата смотрения

passport\_id – идентификатор пользователя

subscription\_kind – тип подписки

subscription\_id – идентификатор подписки

tv\_id – идентификатор тайтла

season\_num, episode\_num – сезон и эпизод

cnt\_views, cnt\_views\_3min, view\_duration\_s – просмотры и длительность смотрения

age\_restriction, genre\_name, geo\_region\_title\_ru – метаданные контента

Extended\_billing\_aggregate

Основные поля:

passport\_id – id пользователя

start\_date, end\_date – даты начала и окончания подписки

money – стоимость подписки

LTV, LT\_total – жизненная ценность пользователя

content\_dash\_table\_till\_user\_driver\_allocation\_from\_2020\_history

Основные поля:

tv\_id, title\_ru, season\_num, dt – идентификаторы контента

watching\_time, alloc\_revenue – данные о выручке и времени просмотра

## 9. Роли

| Пользователь | Получение отчетов в Telegram |
| --- | --- |
| Администратор | Управление пользователями |
| Сервер | Обработка и отправка отчетов |
| Платформа данных | Сбор и подготовка информации |