**Техническое описание проекта по курсу ООАД**

**КиноЛист**

**Студенты ФИТ НГУ:**

**Скопинцев Никита Алексеевич**

**Леонтьев Данил Вячеславович**

**Бурым Максим Юрьевич**

**группа 21210**

**Версия 1.0.4**

Содержание

Оглавление

[1. Введение 3](#_Toc154048413)

[1.1 Цель 3](#_Toc154048414)

[1.2 Область действия 3](#_Toc154048415)

[1.3 Определения и сокращения 3](#_Toc154048416)

[1.4 Ссылки 3](#_Toc154048417)

[1.5 Краткое описание 3](#_Toc154048418)

[2. Предметная область проекта 4](#_Toc154048419)

[2.1 Существующие проблемы 4](#_Toc154048420)

[2.2 Предполагаемое решение 4](#_Toc154048421)

[3. Требования к программному решению 4](#_Toc154048422)

[3.1 Роли 4](#_Toc154048423)

[3.2 Функциональные требования для роли неавторизованного пользователя 4](#_Toc154048424)

[3.2.1 Авторизация 4](#_Toc154048425)

[3.3 Функциональные требования для роли авторизованного пользователя 5](#_Toc154048426)

[3.3.1 Добавление в список желаемого 5](#_Toc154048427)

[3.3.2 Добавление в список просмотренного 5](#_Toc154048428)

[3.3.3 Добавление в список отслеживаемого 6](#_Toc154048429)

[3.3.4 Посмотреть плейлист 7](#_Toc154048430)

[3.3.5 Редактирование списка желаемого 7](#_Toc154048431)

[3.3.6 Удаление фильма/сериала из списка отслеживания 8](#_Toc154048432)

[3.3.7 Удаление фильма/сериала из списка просмотренного 8](#_Toc154048433)

[3.3.8 Выбор рандомного фильма/сериала из списка желаемого 9](#_Toc154048434)

[3.3.9 Выбор рандомного фильма/сериала из об 10](#_Toc154048435)

[3.3.10 Возврат на главный экран из любого места 10](#_Toc154048436)

[3.4 Нефункциональные требования 10](#_Toc154048437)

[4. Обзор архитектуры 11](#_Toc154048438)

[4.1.1 Компонентная модель системы 11](#_Toc154048439)

[4.1.2 Компоненты сторонних производителей 11](#_Toc154048440)

[4.1.3 Схема развертывания приложения 11](#_Toc154048441)

[5. Допущения и ограничения 11](#_Toc154048442)

[6. Известные проблемы 12](#_Toc154048443)

[6.1 Невысокая производительность приложения 12](#_Toc154048444)

Техническое описание проекта по курсу ООАД

# Введение

## Цель

Данный документ представляет собой техническое описание проекта *КиноЛист* и содержит основные требования к разрабатываемой в рамках проекта программной системе и описание архитектуры программного решения.

## Область действия

Документ разработан в рамках проекта *КиноЛист* на основе стандартного шаблона и предназначен для использования студентами ФИТ и преподавателями дисциплины ООАД.

## Определения и сокращения

**Таблица 1: Определения и сокращения**

|  |  |
| --- | --- |
| **Термин** | **Описание** |
| «Телеграм бот" или просто «Бот» | Телеграм бот – это роботизированный аккаунт в мессенджере «Telegram», который запрограммирован на автоматическое совершение действий |

## Ссылки

В тексте содержатся ссылки на следующие документы:

1. P-Cources.OOAD.Kinolist model.asta, v1.0 – UML – диаграммы.

Ссылки приводятся в виде [N], где N – номер документа в вышеприведенном списке.

## Краткое описание

Содержание данного документа построено таким образом, чтобы дать ответ на следующие вопросы:

* Какие проблемы предметной области должен решать будущий программный продукт
* Посредством какой функциональности системы будут достигнуто решение проблем предметной области
* Какова архитектура программного решения

Описание предметной области и проблем, для решения которых предназначен будущий программный продукт, приведены в разделе 2.

Раздел 3 содержит описание требований к программному решению, раздел 4 – описание архитектуры выбранного решения.

# Предметная область проекта

Проект "КиноЛист" решает актуальные проблемы кинолюбителей, предоставляя инновационное решение для управления кинематографическим опытом. Пользователи часто сталкиваются с забывчивостью относительно просмотренного контента, неопределенностью при выборе фильма или сериала на вечер, а также упущением новых серий.

## Существующие проблемы

2.1.1 **Суть проблемы:**

Кинолюбители часто сталкиваются с трудностями в управлении своим кинематографическим опытом. Забывчивость относительно просмотренных фильмов, неопределенность в выборе контента на вечер, а также упущение выхода новых серий сериалов - все эти факторы снижают удовлетворение от просмотра.

2.1.2 **Причины и их влияние на участников:**

* Забывчивость: Пользователи часто забывают, что они хотели посмотреть или уже посмотрели.
* Неопределенность в выборе: Отсутствие системы управления списками ведет к затруднениям в определении, что посмотреть.
* Упущение новых серий: Отсутствие оповещений приводит к упущению выхода новых серий сериалов.

2.1.3 **Пути решения:**

* Система управления списками: Создание механизма для удобного добавления и отслеживания фильмов и сериалов.
* Рекомендации и оповещения: Внедрение системы рекомендаций и автоматических уведомлений о выходе новых серий.

## Предполагаемое решение

* Добавление в список: Возможность добавления фильмов и сериалов в различные списки.
* Поиск и рандомизация: Пользователи могут искать контент по названию, а так же получить рандомный фильм или сериал по жанрам или из списка желаемого.
* Автоматические уведомления: Система оповещений о новых сериях сериалов из списка отслеживаемого.

# Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта *КиноЛист.*

## Роли

В предлагаемой к разработке системе идентифицированы следующие роли:

1. Неавторизованный пользователь – это пользователь, у которого нет главного интерфейса, у него есть одно действие – зарегистрироваться
2. Авторизованный пользователь – это пользователь, которому доступны все функции нашего приложения без каких-либо ограничений

## Функциональные требования для роли неавторизованного пользователя

### Авторизация

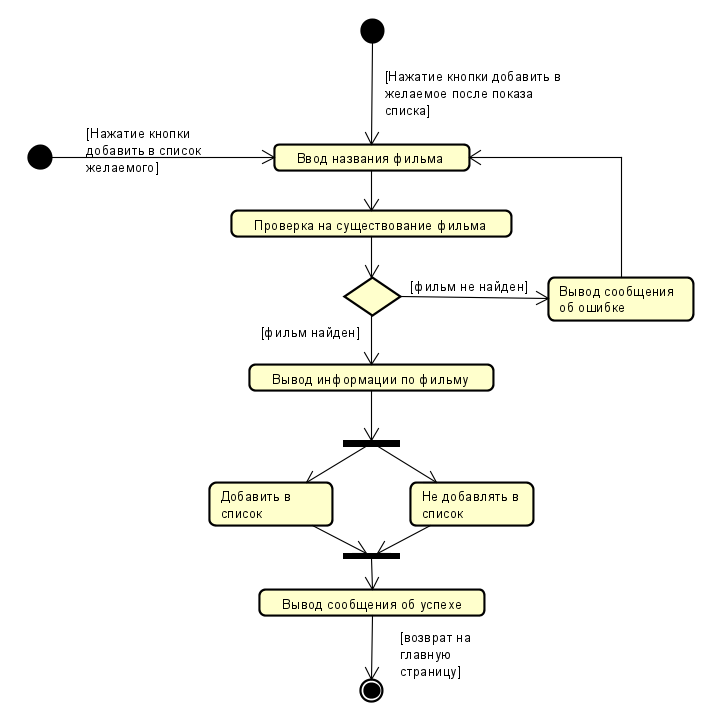
*Неавторизованному пользователю необходимо пройти регистрацию (нажать кнопку “Start”) для перехода в состояние авторизованного пользователя*

**

## Функциональные требования для роли авторизованного пользователя

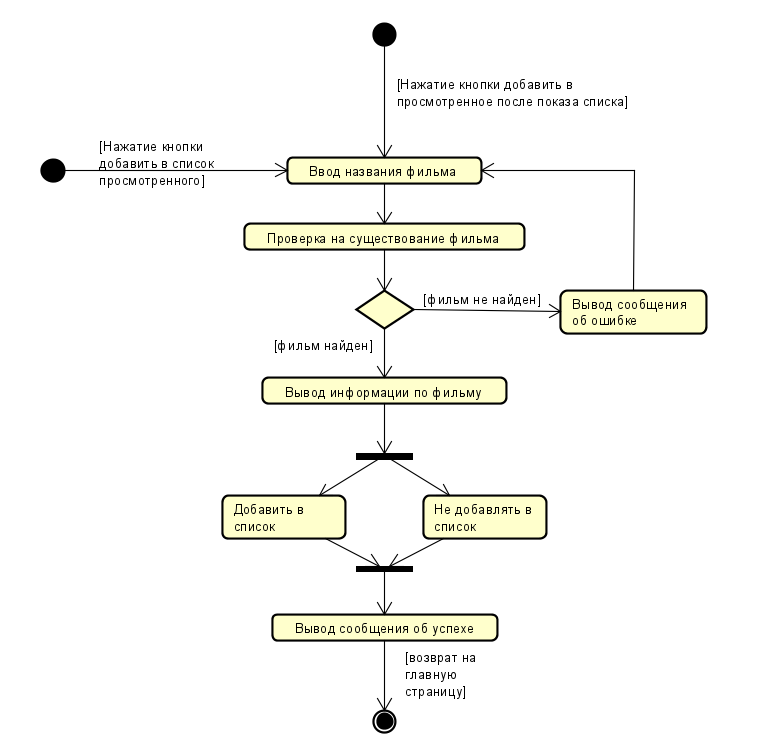
### Добавление в список желаемого

*Пользователь добавляет выбранный им фильм в список желаемых фильмов/сериалов к* просмотру

**

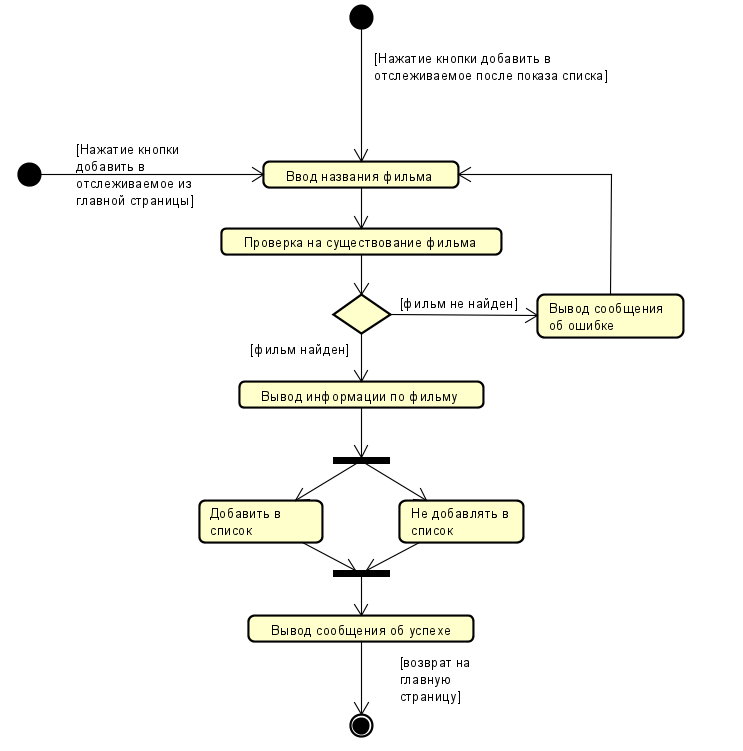
### Добавление в список просмотренного

*Пользователь добавляет выбранный им фильм в список желаемых фильмов/сериалов к просмотру*

**

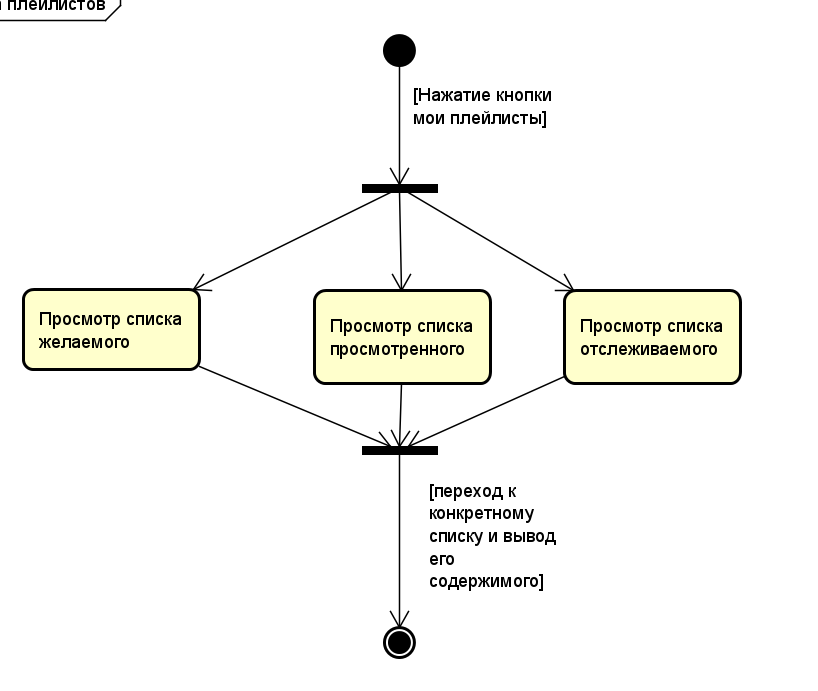
### Добавление в список отслеживаемого

*Пользователь добавляет выбранный им сериал в список отслеживаемого, после чего он становится подписан на уведомления. Как только выходит новая серия или сезон пользователю приходит соответствующее уведомлении о выходе новой серии.*

**

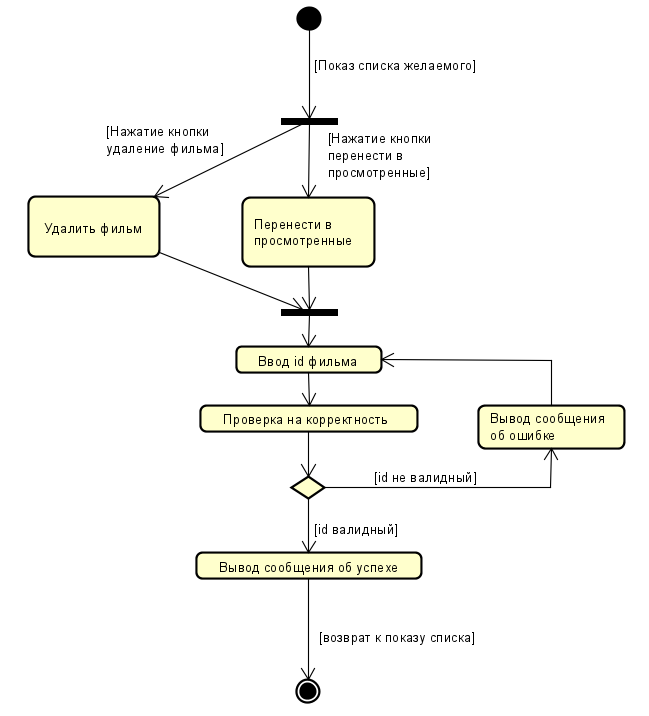
### Посмотреть плейлист

*Пользователю показывается список сериалов и фильмов, которые находятся в определенном плейлисте*

**

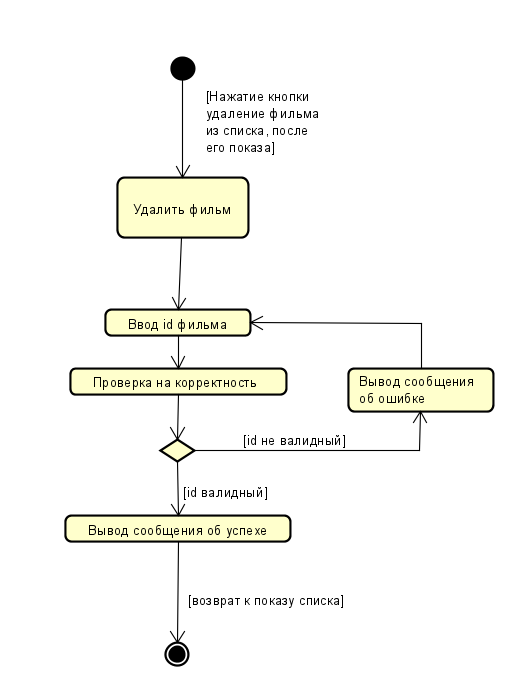
### Редактирование списка желаемого

*Пользователю предоставляется возможность отредактировать список желаемых фильмов/сериалов к просмотру*

**

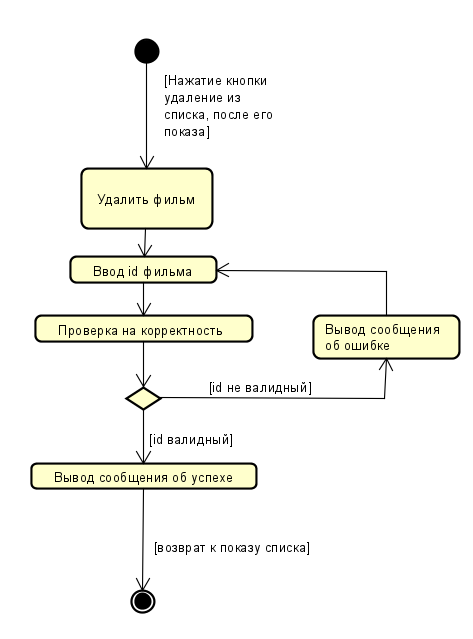
### Удаление фильма/сериала из списка отслеживания

*Пользователь может удалить сериал из своего списка отслеживаемых сериалов, тем самым отменив подписку на выбранный им сериал, отключения оповещений о выходе новых серий*

**

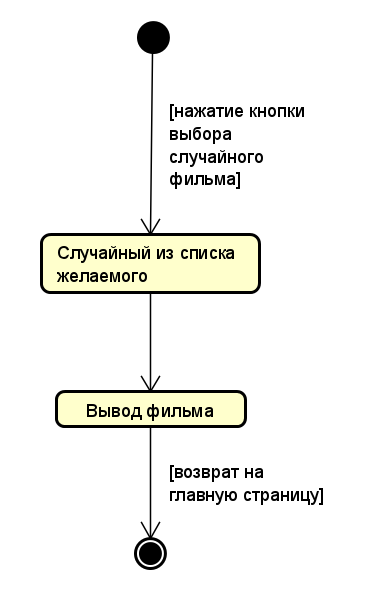
### Удаление фильма/сериала из списка просмотренного

*Пользователь может удалить фильм или сериал из своего списка просмотренного*

**

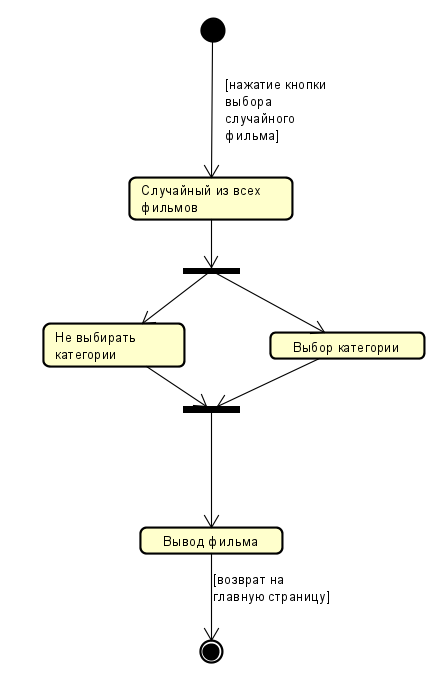
### Выбор рандомного фильма/сериала из списка желаемого

*Пользователь может выбрать фильм или сериал, который выдастся рандомно из списка желаемого пользователя*

**

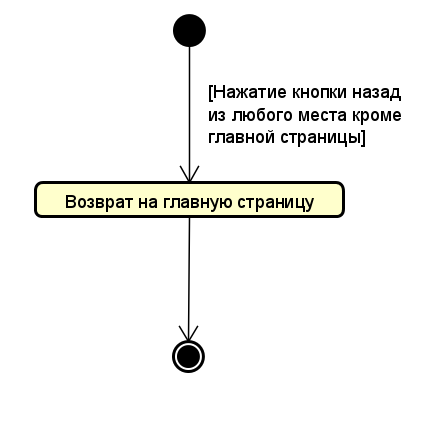
### Выбор рандомного фильма/сериала из об

*Пользователь может выбрать фильм или сериал, который выдастся рандомно из списка всех фильмов/сериалов*

**

### Возврат на главный экран из любого места

*Пользователь может вернуться на главный экран из любого меню бота.*

**

## Нефункциональные требования

*- Серверная часть сервиса должна быть реализована на Java Spring.*

*- Бот должен поддерживать кириллицу.*

# Обзор архитектуры

Этот раздел описывает архитектуру системы.

### Компонентная модель системы

*[Здесь приводится Component diagram - диаграмма компонентов системы, со связями между компонентами и интерфейсами между ними, а также описание их взаимодействия. Для каждого компонента дается краткое описание его места и предназначения в системе]*

#### Компонент 1

*[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]*

#### Компонент 2

*[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]*

### Компоненты сторонних производителей

API:

* Kinopoisk API Unofficial - api для получения информации о фильмах

Фреймворки:

* Java Spring Boot – это средство с открытым исходным кодом, которое упрощает использование платформ на основе Java для создания микрослужб и веб-приложений.

Библиотеки:

* telegrambots-spring-boot-starter – библиотека для взаимодействия с Telegram API
* Hibernate – библиотека ORM для баз данных

Программное обеспечение:

* PostgreSQL – база данных
* Docker – контейнеризация приложения

### Схема развертывания приложения

*[Здесь приводится Deployment diagram - диаграмма развертывания системы, со связями между узлами и указанием способа связи (протокола). На диаграмме обязательно указать, какие компоненты находятся на том или ином узле]*

# Допущения и ограничения

* 1. Допущения:
  + Ограниченность сторонних ресурсов: Проект предполагает доступность сторонних API для поиска фильмов и получения рандомного фильма. Допущение делается о стабильной работе этих сторонних ресурсов в течение всего периода использования бота.
  + Работа с ограниченным числом пользователей: В текущей архитектуре предусмотрено обслуживание ограниченного числа одновременных пользователей, и в случае необходимости увеличения этого числа, могут потребоваться дополнительные меры по масштабированию.
  1. Ограничения:
* На разработку проекта было выделено 4 месяца
* Проект разрабатывается в команде из трёх человек

# Известные проблемы

Ниже приводятся известные на данный момент проблемы и недоработки выработанного программного решения, а также возможные пути их устранения в последующих итерациях проекта.

## Отсутствие поддержки нескольких языков

|  |  |
| --- | --- |
| **Проблема** | Бот в настоящее время ориентирован только на один язык, что может быть ограничивающим для пользователей из разных стран. |
| **Ранг** | 3 (низкий) |
| **Влияние на проект** | Потенциальная аудитория сокращается из-за отсутствия поддержки различных языков. |
| **Пути решения** | Внедрение мультиязычной поддержки для привлечения более широкой аудитории. |

## Ограничения на количество одновременных пользователей

|  |  |
| --- | --- |
| **Проблема** | С увеличением числа пользователей может возникнуть ограничение на количество одновременных активных чатов, ожидающих ввод пользователя, вызванное ограниченностью ресурсов или инфраструктуры. |
| **Ранг** | 8 (высокий) |
| **Влияние на проект** | Возможные проблемы с производительностью и доступностью для пользователей при превышении установленных ограничений. |
| **Пути решения** | - Разработка стратегии масштабирования для обработки большого числа одновременных пользователей.  - Автоматическое удаление временных данных пользователя из кеш памяти. |

## Зависимость от сторонних ресурсов

|  |  |
| --- | --- |
| **Проблема** | Зависимость от сторонних API для выполнения операций поиска фильмов и генерации случайных рекомендаций. Изменения в работе или недоступность этих сторонних ресурсов могут повлиять на функциональность бота. Так же существует ограничения на количество запросов в день. |
| **Ранг** | 9 (высокий) |
| **Влияние на проект** | Отказ стороннего API может вызвать недоступность определенных функций бота, что приведет к ухудшению пользовательского опыта. |
| **Пути решения** | - Резервные источники: Внедрение резервных источников данных для обеспечения доступности функциональности в случае недоступности основных API.  - Обработка ошибок: Реализация механизмов обработки ошибок и уведомлений для создателей о проблемах с внешними ресурсами. |

Лист регистрации изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| **16.10.2023** | **1.0.1** | **Добавил новые роли, а также функциональные Use-Cases для них.** | **Никита** |
| **19.10.2023** | **1.0.2** | **Добавил Use-Case №3.3.14 и добавил нефункциональные требования.** | **Никита** |
| **02.11.2023** | **1.0.3** | **Добавил скриншоты для Use-case.** | **Никита** |
| **02.11.2023** | **1.0.4** | **Убрали некоторые Use-cases, которые оказались сценарием.** | **Никита, Максим, Данил** |
| **21.12.2023** | **1.0.5** | **Заполнил: «Введение», «Предметная область», «Известные проблемы».** | **Никита** |
| **21.12.2023** | **1.1** | **Заполнил «Обзор архитектуры»**  **Проект готов.** | **Никита** |

Лист регистрации проверок

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |