

# Требования к проекту Movie Recommendation Bot

## Общее описание

Целью проекта является разработка Telegram-бота для подбора фильмов на основе предпочтений/случайным образом или по определенным категориям (жанр, год выпуска, актеры, рейтинг и т.д.). Доступна возможность добавления отзывов и оценок фильмам в списке просмотренных, а также добавление фильмов в список "Буду смотреть".

## Описание требований

1. Источники и артефакты
  - 1.1. Контроль версий и совместная работа
    - 1.1.1. Весь код должен быть зафиксирован в репозитории GitHub.
    - 1.1.2. Репозиторий должен включать файл README.md с инструкцией по:
      - 1.1.2.1. Настройке проекта локально.
      - 1.1.2.2. Сборке и запуску приложения.
      - 1.1.2.3. Развертыванию приложения с использованием Docker.
      - 1.1.2.4. Взаимодействию с Telegram-ботом.
    - 1.1.3. История коммитов должна отражать значимые вклады всех членов команды.
  - 1.2. Целевые артефакты
    - 1.2.1. Приложение должно производить запускаемый fat JAR как основной артефакт.
    - 1.2.2. Приложение должно быть развернуто в виде Docker-контейнера.
    - 1.2.3. Использовать инструмент сборки Maven для управления зависимостями, сборки и упаковки приложения.
2. Технологический стек
  - 2.1. Основные технологии
    - 2.1.1. Telegram API: приложение должно интегрироваться с Telegram API для обеспечения функциональности бота.

- 2.1.2. Язык программирования: Java 23
  - 2.1.3. Фреймворк: Spring 6.2.3 (WebFlux, JOOQ, Spring Rest Docs)
  - 2.1.4. База данных: PostgreSQL
  - 2.1.5. Тестирование: JUnit
- 2.2. Развертывание и контейнеризация
  - 2.2.1. Использовать Docker для контейнеризации приложения.
  - 2.2.2. Предоставить файл Docker Compose для локальной разработки и тестирования.
  - 2.2.3. Опубликовать Docker-образ в Docker Hub для развертывания.
- 3. Качество кода
  - 3.1. Стиль кода
    - 3.1.1. Придерживаться признанного руководства по стилю кода на Java (Google Java Style Guide).
  - 3.2. Современные языковые особенности
    - 3.2.1. Использовать современные языковые конструкции и особенности, доступные в выбранном языке.
  - 3.3. Принципы чистого кода
    - 3.3.1. Следовать принципам чистого кода: значимые имена переменных, правильная инкапсуляция и модульный подход.
    - 3.3.2. Избегать дублирования кода и обеспечить правильную обработку ошибок.
- 4. Дополнительные замечания
  - 4.1. Документация: обязательна правильная документация (комментарии в коде, README и т.д.).
  - 4.2. Тестирование: написание модульных тестов.
- 5. Функциональные требования
  - 5.1. Авторизация
    - 5.1.1. через Telegram (только пользователи)
    - 5.1.2. через HTTP (только администраторы)
  - 5.2. Роли
    - 5.2.1. Пользователь
    - 5.2.2. Администратор
  - 5.3. Проверка состояния бота (авторизация не требуется)
    - 5.3.1. /healthcheck
      - 5.3.1.1. Состояние сервера
      - 5.3.1.2. Список авторов (студентов)

#### 5.4. Проверка пользователей (только для администраторов)

##### 5.4.1. Список всех пользователей (их telegramId)

### 6. Основные функции

#### 6.1. Общие

6.1.1. [Идентифицировать пользователя](#) (автоматическая по telegramId, дополнительная регистрация не требуется).

6.1.2. [Получить информацию о фильме.](#)

6.1.3. [Предпочтения.](#)

6.1.3.1. [Показать предпочтения пользователя.](#)

6.1.3.2. [Добавить предпочтения \(жанр, актёры, год выпуска, страна\).](#)

6.1.3.3. [Удалить предпочтения](#) (жанр, актёры, год выпуска, страна).

6.1.4. [Поиск фильмов](#)

6.1.4.1. [Найти фильм с фильтрами \(жанры, актёры, год выпуска, рейтинг, длительность фильма\).](#)

6.1.4.2. [Найти фильм по предпочтениям пользователя \(используются предпочтения пользователя, если они добавлены\).](#)

6.1.4.3. [Найти случайный фильм \(фильтры и предпочтения игнорируются\).](#)

6.1.5. [Список "Буду смотреть"](#)

6.1.5.1. [Добавить фильм в список "Буду смотреть".](#)

6.1.5.2. [Показать список "Буду смотреть".](#)

6.1.5.3. [Удалить фильм из списка "Буду смотреть".](#)

6.1.6. [Список просмотренных фильмов](#)

6.1.6.1. [Добавить фильм в список просмотренных.](#)

6.1.6.2. [Показать все просмотренные фильмы за период](#) (три месяца, последний месяц, неделя, конкретный период).

6.1.6.3. [Добавить оценку к просмотренному фильму](#) (от 0 до 10) и текстовый отзыв.

### 7. Нефункциональные требования

#### 7.1. Производительность

7.1.1. Максимальное время ответа — 600 мс (исключая задержки внешнего API).

7.1.2. Приложение должно поддерживать до 100000  
одновременных пользователей.

7.2. Надежность

7.2.1. Повторные попытки при ошибках внешнего API (3 попытки с  
интервалом 1 сек).

7.2.2. Валидация ввода пользователя.

# Подробное описание функций

Общая информация:

- при вводе пользователя регистр не учитывается;
- информационные сообщения, представленные ниже могут видоизменяться в ходе реализации бота (без влияния на функциональность).

Идентифицировать пользователя (автоматически по telegramId, дополнительная регистрация не требуется)

**Ожидаемое поведение пользователя:** запускает бота через /start.

Корректный ввод: команда /start.

Некорректный ввод: любой ввод, кроме команды /start.

**Реакция системы:**

1. Бот идентифицирует пользователя по telegramId.
2. Если пользователь новый, создается запись в БД пользователей.
3. Выводится приветственное (информационное) сообщение с доступными командами меню и появляются кнопки начального меню:

[🎬Привет, {username}! Добро пожаловать в кино-бота! 🍿]

Я помогу тебе найти информацию о фильмах, управлять твоими предпочтениями и вести списки просмотренных и запланированных к просмотру фильмов.

📌 **Вот что я умею:**

- ♦ Вывести меню - /menu
- ♦ Получить информацию о фильме — /infoAboutFilm
- ♦ Поиск фильмов — /searchFilm
- ♦ Предпочтения — /preferences
- ♦ Список просмотренных фильмов — /watchedFilmsList
- ♦ Добавить фильм в список просмотренных — /markFilm
- ♦ Добавить оценку фильму — /addScoreToWatchedFilm
- ♦ Добавить текстовый отзыв фильму — /addReviewToWatchedFilm
- ♦ Список "Буду смотреть" — /watchlist

Выбери нужную команду или просто нажми кнопку! 😊"]

## Меню

**Ожидаемое поведение пользователя:** вводит команду /menu.

**Реакция системы:**

1. Бот выводит команды начального меню

Начальное меню:

- 1.1. Получить информацию о фильме (/infoAboutFilm),

- 1.2. Предпочтения (/preferences),
- 1.3. Поиск фильмов (/searchFilm)
- 1.4. Список просмотренных фильмов (/watchedFilmsList)
- 1.5. Добавить фильм в список просмотренных (/markFilm)
- 1.6. Добавить оценку фильму (/addScoreToWatchedFilm)
- 1.7. Добавить текстовый отзыв фильму (/addReviewToWatchedFilm)
- 1.8. Список "Буду смотреть" (/watchlist)
2. Возврат в начальное меню

## Получить информацию о фильме

**Ожидаемое поведение пользователя:** вводит команду /infoAboutFilm или нажимает кнопку "Получить информацию о фильме".

### Реакция системы:

1. Бот запрашивает у пользователя название фильма, о котором пользователь хочет получить информацию.
2. Пользователь вводит название фильма.
  - 2.1. Корректный ввод: название фильма, которое есть в БД фильмов.
  - 2.2. Некорректный ввод: название фильма, которого нет в БД фильмов.
3. Если 2.1: пользователю выводится подробная информация о фильме:  
Основная информация о фильме
  1. Название на русском языке.
  2. Год премьеры в мире.
  3. Рейтинги:
    - 3.1. Рейтинг Кинопоиска.
    - 3.2. Рейтинг IMDb.
    - 3.3. Рейтинг кинокритиков.
  4. Описание и метаданные:
    - 4.1. Полное описание.
    - 4.2. Возрастное ограничение.
  5. Список жанров.
  6. Страны производства.
  7. Список актеров (до 5 штук).
  8. Технические данные:
    - 8.1. Длительность.
    - 8.2. Бюджет.
    - 8.3. Сборы.
  9. Похожие фильмы (до 3 штук, если имеются в БД фильмов).

3. Если 2.2: вывод информационного сообщения о некорректном вводе  
[“Фильм не найден. Проверьте название или попробуйте другой.”] и возврат к шагу 2  
(повторный ввод названия фильма).

## Предпочтения

**Ожидаемое поведение пользователя:** вводит команду */preferences*.

**Реакция системы:**

Пользователю становятся доступными кнопки для работы с предпочтениями и выводится информационное сообщение:

[“Нажмите одну из кнопок или воспользуйтесь одной из команд:

А. Показать предпочтения */showMyPreferences*)

В. Добавить предпочтения *(/addPref)*

С. Удалить предпочтения *(/deletePref)*”].

Если А:

### Показать предпочтения пользователя

**Ожидаемое поведение пользователя:** вводит команду */showMyPreferences* или нажимает кнопку “Показать предпочтения”.

**Реакция системы:**

1. Если предпочтения есть: вывод списка по каждой категории (жанры, актеры, страны, годы выпуска фильмов).
2. Если нет предпочтений: информационное сообщение: [“У вас пока нет предпочтений. Добавьте их с помощью команды */addPref*.”]

Если В:

### Добавить предпочтения (жанр, актёры, год выпуска, страна)

**Ожидаемое поведение пользователя:** вводит команду */addPref* или нажимает кнопку “Добавить предпочтения”.

**Реакция системы:**

1. Пользователю становятся доступными кнопки для добавления предпочтений по категориям и выводится информационное сообщение:  
[“Для добавления предпочтений нажмите одну из кнопок или воспользуйтесь одной из команд:  
    - a. Добавить жанр *(/addGenrePref)*
    - b. Добавить актёра *(/addActorPref)*
    - c. Добавить страну производства *(/addCountryPref)*
    - d. Добавить год выпуска *(/addYearPref)*”]
  2. Далее в зависимости от выбранной команды/нажатой кнопки:
-

Если а:

Ожидаемое поведение пользователя: вводит команду /addGenrePref или нажимает кнопку “Добавить жанр”.

**Реакция системы:**

1. Бот запрашивает у пользователя жанр/ы и выводит информационное сообщение о формате ввода.
  2. Пользователь вводит жанр/ы.
    - 2.1. Корректный ввод: жанр/ы, через запятую, которые есть в БД фильмов и которых нет в БД пользователей в таблице предпочтений.
    - 2.2. Некорректный ввод: жанр/ы, которых нет в БД фильмов или которые есть в БД пользователей в таблице предпочтений.
  3. Если 2.1: пользователю выводится информационное сообщение [“Введенные жанры: {жанры из 2.1} успешно добавлены в ваши предпочтения.”] и они добавляются в предпочтения пользователя.
  4. Если 2.2: пользователю выводится информационное сообщение: [“Введенные жанры: {жанры из 2.2} не удалось добавить в ваши предпочтения + {причина из 2.2: они уже есть в вашем списке предпочтений/таких жанров нет в БД фильмов}”].
  5. Возврат в состояние /addPref.
- 

Если b:

Ожидаемое поведение пользователя: вводит команду /addActorPref или нажимает кнопку “Добавить актера”.

**Реакция системы:**

1. Бот запрашивает у пользователя ФИ актеров и выводит информационное сообщение о формате ввода.
2. Пользователь вводит ФИ актеров.
  - 2.1. Корректный ввод: ФИ актеров, через запятую, которые есть в БД фильмов и которых нет в БД пользователей в таблице предпочтений.
  - 2.2. Некорректный ввод: ФИ актеров, которых нет в БД фильмов или которые есть в БД пользователей в таблице предпочтений.
3. Если 2.1: пользователю выводится информационное сообщение [“Введенные ФИ актеров: {ФИ актеров из 2.1} успешно добавлены в ваши предпочтения.”] и они добавляются в предпочтения пользователя.
4. Если 2.2: пользователю выводится информационное сообщение [“Введенные ФИ актеров: {ФИ актеров из 2.2} не удалось добавить в ваши предпочтения + {причина из 2.2: они уже есть в вашем списке предпочтений/таких ФИ актеров нет в БД фильмов}”].



## 5. Возврат в состояние */addPref*.

---

Если с:

**Ожидаемое поведение пользователя:** вводит команду */addCountryPref* или нажимает кнопку "Добавить страну производства фильма".

**Реакция системы:**

1. Бот запрашивает у пользователя страны и выводит информационное сообщение о формате ввода.
  2. Пользователь вводит страны.
    - 2.1. Корректный ввод: страны, через запятую, которые есть в БД фильмов и которых нет в БД пользователей в таблице предпочтений.
    - 2.2. Некорректный ввод: страны, которых нет в БД фильмов или которые есть в БД пользователей в таблице предпочтений.
  3. Если 2.1: пользователю выводится информационное сообщение: ["Введенные страны: {страны из 2.1} успешно добавлены в ваши предпочтения."] и они добавляются в предпочтения пользователя.
  4. Если 2.2: пользователю выводится информационное сообщение: ["Введенные страны: {страны из 2.2} не удалось добавить в ваши предпочтения + {причина из 2.2: они уже есть в вашем списке предпочтений/таких стран нет в БД фильмов}."].
  5. Возврат в состояние */addPref*.
- 

Если d:

**Ожидаемое поведение пользователя:** вводит команду */addYearPref* или нажимает кнопку "Добавить год выпуска".

**Реакция системы:**

1. Бот запрашивает у пользователя годы и выводит информационное сообщение о формате ввода.
2. Пользователь вводит годы.
  - 2.1. Корректный ввод: диапазон лет в формате ГГГГ-ГГГГ (например 2002-2004) или годы через запятую (2002, 2003, 2004), которых нет в БД пользователей в таблице предпочтений. (годы от 1888 до текущего)
  - 2.2. Некорректный ввод: которые есть в БД пользователей в таблице предпочтений и которые не в диапазоне 1888-текущий год.
3. Если 2.1: пользователю выводится информационное сообщение: ["Введенные годы/диапазон: {годы/диапазон из 2.1} успешно добавлены в ваши предпочтения."] и они добавляются в предпочтения пользователя.

4. Если 2.2: пользователю выводится информационное сообщение ["Введенные страны: {годы/диапазон из 2.2} не удалось добавить в ваши предпочтения + {причина из 2.2: они уже есть в вашем списке предпочтений/таких годов нет в БД фильмов}."].
5. Возврат в состояние */addPref*.

Если С:

Удалить предпочтения (жанр, актёры, год выпуска, страна)

**Ожидаемое поведение пользователя:** вводит команду */deletePref*.

**Реакция системы:**

1. Пользователю становятся доступными кнопки для добавления предпочтений по категориям и выводится информационное сообщение ["Для удаления предпочтений нажмите одну из кнопок или воспользуйтесь одной из команд:
  - a. Удалить жанр (*/deleteGenrePref*)
  - b. Удалить актёра (*/deleteActorPref*)
  - c. Удалить страну производства (*/deleteCountryPref*)
  - d. Удалить год выпуска (*/deleteYearPref*)"]
2. Далее в зависимости от выбранной команды/нажатой кнопки:

---

Если а:

**Ожидаемое поведение пользователя:** вводит команду */deleteGenrePref* или нажимает кнопку "Удалить жанр".

**Реакция системы:**

1. Пользователю выводится список его предпочтений по жанрам.
2. Бот запрашивает у пользователя жанр/ы и выводит информационное сообщение о формате ввода.
3. Пользователь вводит жанр/ы.
  - 3.1. Корректный ввод: жанр/ы, через запятую, которые есть в предпочтениях этого пользователя.
  - 3.2. Некорректный ввод: жанр/ы, которых нет в предпочтениях этого пользователя.
4. Если 2.1: пользователю выводится информационное сообщение ["Введенные жанры: {жанры из 2.1} успешно удалены из ваших предпочтений.] и они удаляются из предпочтений пользователя.
5. Если 2.2: пользователю выводится информационное сообщение: ["Введенные жанры: {жанры из 2.2} не удалось удалить из ваших предпочтений + {причина из 2.2: нет в вашем списке предпочтений}"].
6. Возврат в состояние */deletePref*.

---

Если b:

**Ожидаемое поведение пользователя:** вводит команду */deleteActorPref* или нажимает кнопку “Удалить актера”.

**Реакция системы:**

1. Пользователю выводится список его предпочтений по актерам.
  2. Бот запрашивает у пользователя ФИ актеров и выводит информационное сообщение о формате ввода.
  3. Пользователь вводит ФИ актеров.
    - 3.1. Корректный ввод: ФИ актеров, через запятую, которые есть в предпочтениях этого пользователя.
    - 3.2. Некорректный ввод: ФИ актеров, которых нет в предпочтениях этого пользователя.
  4. Если 2.1: пользователю выводится информационное сообщение [“Введенные ФИ актеров: {ФИ актеров из 2.1} успешно удалены из ваших предпочтений.”] и они удаляются из предпочтений пользователя.
  5. Если 2.2: пользователю выводится информационное сообщение [“Введенные ФИ актеров: {ФИ актеров из 2.2} не удалось добавить в ваши предпочтения + {причина из 2.2: нет в вашем списке предпочтений}.”].
  6. Возврат в состояние */deletePref*.
- 

Если с:

**Ожидаемое поведение пользователя:** вводит команду */deleteCountryPref* или нажимает кнопку “Удалить страну производства фильма”.

**Реакция системы:**

1. Пользователю выводится список его предпочтений по странам.
  2. Бот запрашивает у пользователя страны и выводит информационное сообщение о формате ввода.
  3. Пользователь вводит страны.
    - 3.1. Корректный ввод: страны, через запятую, которые есть в предпочтениях этого пользователя.
    - 3.2. Некорректный ввод: страны, которых нет в предпочтениях этого пользователя.
  4. Если 2.1: пользователю выводится информационное сообщение [“Введенные страны: {страны из 2.1} успешно удалены из ваших предпочтений.”] и они удаляются из предпочтений пользователя.
  5. Если 2.2: пользователю выводится информационное сообщение [“Введенные страны: {страны из 2.2} не удалось добавить в ваши предпочтения + {причина из 2.2: нет в вашем списке предпочтений}.”].
  6. Возврат в состояние */deletePref*.
- 

Если d:

**Ожидаемое поведение пользователя:** вводит команду */addYearPref* или нажимает кнопку “Удалить год выпуска”.

**Реакция системы:**

1. Пользователю выводится список его предпочтений по годам.
2. Бот запрашивает у пользователя годы и выводит информационное сообщение о формате ввода.
3. Пользователь вводит годы.
  - 3.1. Корректный ввод: диапазон лет в формате ГГГГ-ГГГГ (например 2002-2004) или годы через запятую (2002, 2003, 2004), которые есть в предпочтениях этого пользователя.
  - 3.2. Некорректный ввод: годы, которых нет в предпочтениях этого пользователя.
4. Если 2.1: пользователю выводится информационное сообщение [“Введенные годы/диапазон: {годы/диапазон из 2.1} успешно удалены из ваших предпочтений.”]] и они удаляются из предпочтений пользователя.
5. Если 2.2: пользователю выводится информационное сообщение [“Введенные страны: {годы/диапазон из 2.2} не удалось добавить в ваши предпочтения + {причина из 2.2: нет в вашем списке предпочтений}.”].
6. Возврат в состояние */deletePref*.

## Поиск фильмов

**Ожидаемое поведение пользователя:** вводит команду */searchFilm*.

**Реакция системы:**

1. Пользователю становятся доступными кнопки для выбора варианта поиска фильмов и выводится информационное сообщение: [“Для поиска фильмов нажмите одну из кнопок или воспользуйтесь одной из команд:
  - A. Использовать фильтры (*/searchByFilters*)
  - B. Использовать предпочтения (*/searchByPref*)
  - C. Случайный фильм (*/searchRandom*)”].

Если A:

Найти фильм с фильтрами (жанры, актёры, год выпуска, рейтинг, длительность фильма).

**Ожидаемое поведение пользователя:** вводит команду */searchByFilters* или нажимает кнопку “Использовать фильтры”.

**Реакция системы:**

1. Пользователю становятся доступными кнопки для добавления фильтров по категориям и выводится информационное сообщение ["Для добавления фильтров нажмите одну из кнопок или воспользуйтесь одной из команд:
    - a. Выбрать жанр (/chooseGenres)
    - b. Выбрать актера (/chooseActors)
    - c. Выбрать рейтинг (/chooseRate)
    - d. Выбрать длительность (/chooseDuration)
    - e. Выбрать год выпуска (/chooseYears)
    - f. Выбрать страну производства (/chooseCountry)
    - g. Применить фильтры (/applyFilters)При повторном выборе параметра предыдущий выбор будет сброшен."]
  2. Далее в зависимости от выбранной команды/нажатой кнопки:
- 

Если a:

**Ожидаемое поведение пользователя:** вводит команду /chooseGenres или нажимает кнопку "Выбрать жанр".

**Реакция системы:**

1. Сброс предыдущего выбора.
  2. Пользователю выводится список доступных жанров в БД фильмов.
  3. Бот запрашивает у пользователя жанр/ы и выводит информационное сообщение о формате ввода.
  4. Пользователь вводит жанр/ы.
    - 4.1. Корректный ввод: жанр/ы, через запятую, которые есть в БД фильмов,
    - 4.2. Некорректный ввод: жанр/ы, которых нет в БД фильмов.
  5. Если 2.1: пользователю выводится информационное сообщение ["Введенные жанры: {жанры из 2.1} успешно добавлены в фильтры для поиска."].
  6. Если 2.2: пользователю выводится информационное сообщение ["Введенные жанры: {жанры из 2.2} не удалось добавить в фильтры для поиска, т.к. таких жанров нет в БД фильмов"].
  7. Возврат в состояние /searchByFilters.
- 

Если b:

**Ожидаемое поведение пользователя:** вводит команду /chooseActors или нажимает кнопку "Выбрать актера".

**Реакция системы:**

1. Сброс предыдущего выбора.
2. Бот запрашивает у пользователя ФИ актеров и выводит информационное сообщение о формате ввода.
3. Пользователь вводит ФИ актеров

- 3.1. Корректный ввод: ФИ актеров, которые есть в БД фильмов, через запятую.
  - 3.2. Некорректный ввод: ФИ актеров, которых нет в БД фильмов.
  4. Если 2.1: пользователю выводится информационное сообщение ["Введенные ФИ актеров: {ФИ актеров из 2.1} успешно добавлены в фильтры для поиска."].
  5. Если 2.2: пользователю выводится информационное сообщение ["Введенные ФИ актеров: {ФИ актеров из 2.2} не удалось добавить в фильтры для поиска, т.к. таких ФИ актеров нет в БД фильмов"].
  6. Возврат в состояние */searchByFilters*.
- 

Если с:

**Ожидаемое поведение пользователя:** вводит команду */chooseRate* или нажимает кнопку "Выбрать рейтинг".

**Реакция системы:**

1. Сброс предыдущего выбора.
  2. Бот запрашивает у пользователя рейтинг и выводит информационное сообщение о формате ввода.
  3. Пользователь вводит рейтинг
    - 3.1. Корректный ввод: рейтинги через запятую или диапазон в формате Ч-Ч (где Ч - вещественное/целое число от 0 до 10).
    - 3.2. Некорректный ввод: рейтинг, который не является вещественным/целым числом от 0 до 10.
  4. Если 2.1: пользователю выводится информационное сообщение ["Введенный рейтинг: {рейтинг из 2.1} успешно добавлен в фильтры для поиска."].
  5. Если 2.2: пользователю выводится информационное сообщение ["Введенный рейтинг: {рейтинг из 2.2} не удалось добавить в фильтры для поиска, некорректный ввод"].
  6. Возврат в состояние */searchByFilters*.
- 

Если d:

**Ожидаемое поведение пользователя:** вводит команду */chooseDuration* или нажимает кнопку "Выбрать длительность".

**Реакция системы:**

1. Сброс предыдущего выбора.
2. Бот запрашивает у пользователя длительность и выводит информационное сообщение о формате ввода.
3. Пользователь вводит длительность
  - 3.1. Корректный ввод: длительность в минутах - диапазон в формате Ч-Ч (где Ч -целое число от 0 до 51420).

- 3.2. Некорректный ввод: длительность, которая не является вещественным/целым числом от 0 до 51420.
  4. Если 2.1: пользователю выводится информационное сообщение ["Введенная длительность: {длительность из 2.1} успешно добавлена в фильтры для поиска."].
  5. Если 2.2: пользователю выводится информационное сообщение ["Введенная длительность: {длительность из 2.2} не удалось добавить в фильтры для поиска, некорректный ввод"].
  6. Возврат в состояние */searchByFilters*.
- 

Если e:

**Ожидаемое поведение пользователя:** вводит команду */chooseYear* или нажимает кнопку "Выбрать год выпуска".

**Реакция системы:**

1. Сброс предыдущего выбора.
  2. Бот запрашивает у пользователя годы и выводит информационное сообщение о формате ввода.
  3. Пользователь вводит годы
    - 3.1. Корректный ввод: диапазон лет в формате ГГГГ-ГГГГ (например 2002-2004) или годы через запятую (2002, 2003, 2004).
    - 3.2. Некорректный ввод: любой другой формат.
  4. Если 2.1: пользователю выводится информационное сообщение ["Введенные годы: {годы из 2.1} успешно добавлена в фильтры для поиска."].
  5. Если 2.2: пользователю выводится информационное сообщение ["Введенные годы: {годы из 2.2} не удалось добавить в фильтры для поиска, некорректный ввод"].
  6. Возврат в состояние */searchByFilters*.
- 

Если f:

**Ожидаемое поведение пользователя:** вводит команду */chooseCountry* или нажимает кнопку "Выбрать страну производства".

**Реакция системы:**

1. Бот запрашивает у пользователя страны и выводит информационное сообщение о формате ввода.
2. Пользователь вводит страны
  - 2.1. Корректный ввод: страны, через запятую, которые есть в БД фильмов.
  - 2.2. Некорректный ввод: любой другой формат.
3. Если 2.1: пользователю выводится информационное сообщение ["Введенные страны: {страна из 2.1} успешно добавлены в фильтры для поиска."].

4. Если 2.2: пользователю выводится информационное сообщение ["Введенные страны: {страны из 2.2} не удалось добавить в фильтры для поиска, таких стран нет в БД фильмов."].
  5. Возврат в состояние */searchByFilters*.
- 

Если g:

**Ожидаемое поведение пользователя:** вводит команду */applyFilters* или нажимает кнопку "Применить фильтры".

**Реакция системы:**

1. Все ранее введенные фильтры (a-f) применяются и выполняется поиск фильма.
  - 1.1. Если фильтры не введены, выводится информационное сообщение ["Вы не ввели ни одного фильтра, поиск невозможен."].
2. Если не 1.1, пользователю выводится список из 5 фильмов с подробным описанием (аналогичным */infoAboutFilm*, но без похожих к каждому), наиболее подходящих под фильтры или информационное сообщение о том, что фильмы не найдены: ["Не найдено ни одного фильма подходящего под ваши фильтры поиска."].
3. Возврат в состояние */searchFilm*.

Если B:

Найти фильм по предпочтениям пользователя.

**Ожидаемое поведение пользователя:** вводит команду */searchByPref* или нажимает кнопку "Использовать предпочтения".

**Реакция системы:**

1. Все ранее введенные предпочтения пользователя (a-d) применяются и выполняется поиск фильма.
  - 1.1. Если предпочтений не добавлены, выводится информационное сообщение ["У вас нет добавленных предпочтений, поиск невозможен."].
2. Если не 1.1, пользователю выводится список из 5 фильмов с подробным описанием (аналогичным */infoAboutFilm*, но без похожих к каждому), наиболее подходящих под предпочтений или информационное сообщение о том, что фильмы не найдены: ["Не найдено ни одного фильма подходящего под ваши предпочтения."].
3. Возврат в состояние */searchFilm*.

Если C:



Найти случайный фильм (фильтры и предпочтения игнорируются).

**Ожидаемое поведение пользователя:** вводит команду */searchRandom* или нажимает кнопку “Случайный фильм”.

**Реакция системы:**

1. Выполняется поиск случайного фильма без учета предпочтений пользователя и фильтров.
2. Пользователю выводится подробное описание фильма (аналогичным */infoAboutFilm*, но без похожих к каждому).
3. Возврат в состояние */searchFilm*.

## Списки и оценки

### Список просмотренных фильмов

**Ожидаемое поведение пользователя:** вводит команду */watchedFilmsList* или нажимает на кнопку “Список просмотренных фильмов”.

**Реакция системы:**

1. Пользователю становятся доступными кнопки для выбора периода, за который он хочет получить список просмотренных фильмов, и выводится информационное сообщение:  
[“Для выбора периода, за который Вы хотите получить список просмотренных фильмов, воспользуйтесь одной из кнопок или введите одну из команд):
  - a. Три месяца (*/threeMonths*)
  - b. Месяц (*/lastMonth*)
  - c. Год (*/lastYear*)
  - d. Все время (*/allPeriod*)
  - e. Конкретный период (*/exactPeriod*)”]
2. Далее в зависимости от выбранной команды/нажатой кнопки:

---

Если а:

**Ожидаемое поведение пользователя:** вводит команду */threeMonths* или нажимает кнопку “Три месяца”.

**Реакция системы:**

1. Бот выводит список просмотренных фильмов за последние три месяца, а также оценку и отзыв на фильм(если пользователь их добавлял). Если у пользователя нет добавленных фильмов/нет фильмов за последние три месяца, то бот выводит сообщение: [“У вас нет просмотренных фильмов за последние три месяца.”].
2. Возврат в состояние */watchedFilmsList*.

---

Если b:

**Ожидаемое поведение пользователя:** вводит команду */lastMonth* или нажимает кнопку “Месяц”.

**Реакция системы:**

1. Бот выводит список просмотренных фильмов за последний месяц, а также оценку и отзыв на фильм(если пользователь их добавлял). Если у пользователя нет добавленных фильмов/нет просмотренных фильмов за последний месяц, то бот выводит сообщение: [“У вас нет просмотренных фильмов за последний месяц.”].
  2. Возврат в состояние */watchedFilmsList*.
- 

Если c:

**Ожидаемое поведение пользователя:** вводит команду */lastYear* или нажимает кнопку “Год”.

**Реакция системы:**

1. Бот выводит список просмотренных фильмов за последний год, а также оценку и отзыв на фильм(если пользователь их добавлял). Если у пользователя нет добавленных фильмов/нет просмотренных фильмов за последний год, то бот выводит сообщение: [“У вас нет просмотренных фильмов за последний год.”].
  2. Возврат в состояние */watchedFilmsList*.
- 

Если d:

**Ожидаемое поведение пользователя:** вводит команду */allPeriod* или нажимает кнопку “Все время”.

**Реакция системы:**

1. Бот выводит список просмотренных фильмов за все время использования бота, а также оценку и отзыв на фильм(если пользователь их добавлял). Если у пользователя нет добавленных фильмов/нет просмотренных фильмов, то бот выводит сообщение: [“У вас нет просмотренных фильмов.”].
  2. Возврат в состояние */watchedFilmsList*.
- 

Если e:

**Ожидаемое поведение пользователя:** вводит команду */exactPeriod* или нажимает кнопку “Конкретный период”.

**Реакция системы:**

1. Бот запрашивает у пользователя период и выводит информационное сообщение.
2. Пользователь вводит диапазон дат
  - 2.1. Корректный ввод: диапазон дат в формате "н-в", где "н" - нижняя граница, "в" - верхняя граница; границы в формате "дд.мм.гг"(например, 20.11.2024 - 12.12.2024).
  - 2.2. Некорректный ввод: любой другой формат.
3. Если 2.1: пользователю выводится список фильмов за введенный период. Если у пользователя нет добавленных фильмов/нет просмотренных фильмов за конкретный период, то бот выводит сообщение: ["У вас нет просмотренных фильмов за период {период}."].
4. Если 2.2: пользователю выводится информационное сообщение ["Период введен некорректно. Вывод фильмов за этот период невозможен."].
5. Возврат в состояние */watchedFilmsList*.

#### Добавить фильм в список просмотренных

**Ожидаемое поведение пользователя:** вводит команду */markFilm* или нажимает кнопку "Добавить фильм в список просмотренных".

**Реакция системы:**

1. Бот запрашивает название фильма и выводит информационное сообщение.
2. Пользователь вводит название фильма.
  - 2.1. Корректный ввод: фильм, который есть в БД фильмов.
  - 2.2. Некорректный ввод: фильм, которого нет в БД фильмов.
3.
  - а. Если 2.1: выводится информационное сообщение ["Фильм {Название фильма} добавлен в список просмотренных фильмов."].
  - б. Фильм добавляется в список просмотренных фильмов пользователя. Добавляем без оценки и отзыва.
4. Если 2.2: выводится информационное сообщение ["Не удалось найти фильм с таким названием. В список просмотренных фильмов ничего не добавлено."].

#### Добавить оценку фильму

**Ожидаемое поведение пользователя:** вводит команду */addScoreToWatchedFilm* или нажимает кнопку "Добавить оценку фильму".

**Реакция системы:**

1. Бот запрашивает название фильма и выводит информационное сообщение о формате ввода.

2. Пользователь вводит название фильма.

2.1. Корректный ввод: фильм, который есть в БД фильмов.

2.2 Некорректный ввод: фильм, которого нет в БД фильмов.

Если 2.1: выводится информационное сообщение ["Фильм {Название фильма} найден. Добавьте оценку фильму в следующем сообщении. Оценка должна быть целым числом от 0 до 10."]

2.1.1 Пользователь вводит оценку фильма

2.1.1.1 Корректный ввод: целое число от 0 до 10.

2.1.1.2 Некорректный ввод: иное число.

Если 2.1.1.1: бот выводит сообщение: ["Вы поставили фильму {Название фильма} оценку {Оценка}. Фильм {Название фильма}."]. Фильм добавляется в список просмотренных фильмов пользователя, если его там еще нет.

Если 2.1.1.2: бот выводит сообщение: ["Вы ввели некорректную оценку. Повторите ввод."]

Если 2.2: выводится информационное сообщение ["Фильм {Название фильма} не найден. Повторите ввод."].

3. Возврат в начальное меню.

## Добавить текстовый отзыв фильму

**Ожидаемое поведение пользователя:** вводит команду

`/addReviewToWatchedFilm` или нажимает кнопку "Добавить отзыв на фильм".

**Реакция системы:**

1. Бот запрашивает название фильма и выводит сообщение о формате ввода.

2. Пользователь вводит название фильма

2.1. Корректный ввод: фильм, который есть в БД фильмов.

2.2 Некорректный ввод: фильм, которого нет в БД фильмов.

Если 2.1: выводится информационное сообщение: ["Фильм {Название фильма} найден. Напишите отзыв фильму в следующем сообщении."]

2.1.1. Пользователь отправляет текстовый отзыв.

2.1.2. Бот выводит сообщение ["Вы добавили отзыв фильму {Название фильма}."]. Фильм добавляется в список просмотренных фильмов пользователя, если его там еще нет.

Если 2.2.: бот выводит информационное сообщение ["Фильм {Название фильма} не найден. Повторите ввод."].

3. Возврат к командам меню.

## Список "Буду смотреть"

**Ожидаемое поведение пользователя:** вводит команду `/watchlist` или нажимает кнопку "Список "Буду смотреть"".

### Реакция системы:

1. Пользователю становятся доступными кнопки для работы со списком "Буду смотреть" и выводится информационное сообщение: ["Для работы со списком "Буду смотреть" нажмите одну из кнопок или воспользуйтесь одной из команд:
  - a. Показать список "Буду смотреть" (`/showWatchList`)
  - b. Добавить фильм в список "Буду смотреть" (`/addToWatchList`)
  - c. Удалить фильмы из списка "Буду смотреть" (`/deleteFromWatchList`)"]
2. Далее в зависимости от выбранной команды/нажатой кнопки:

---

Если a:

## Показать список "Буду смотреть"

**Ожидаемое поведение пользователя:** вводит команду `/showWatchList` или нажимает кнопку "Показать список "Буду смотреть"".

### Реакция системы:

1. Бот выводит список "Буду смотреть", если есть добавленные фильмы. Иначе, выводится информационное сообщение: ["Список "Буду смотреть" пуст."]
2. Возврат в состояние `/watchlist`.

---

Если b:

## Добавить фильм в список "Буду смотреть"

**Ожидаемое поведение пользователя:** вводит команду `/addToWatchList` или нажимает кнопку "Добавить фильм в список "Буду смотреть"".

### Реакция системы:

1. Бот запрашивает название фильма и выводит информационное сообщение с форматом ввода.
2. Пользователь вводит название фильма.
  - 2.1. Корректный ввод: фильм, который есть в БД фильмов.
  - 2.2. Некорректный ввод: фильма, которого нет в БД фильмов.
3. Если 2.1: бот выводит информационное сообщение ["Фильм {Название фильма} успешно добавлен в список "Буду смотреть"].
4. Если 2.2: бот выводит информационное сообщение ["Фильм {Название фильма} не найден. В список "Буду смотреть" ничего не добавлено."].
5. Возврат в состояние `/watchlist`.

---

### Удалить фильм из списка “Буду смотреть”

Если с:

**Ожидаемое поведение от пользователя:** вводит команду `/deleteFromWatchList` или нажимает кнопку “Удалить фильм из списка “Буду смотреть””.

**Реакция системы:**

1. Бот запрашивает название фильма и выводит информационное сообщение с форматом ввода.
2. Пользователь вводит название фильма
  - 2.1. Корректный ввод: фильм, который есть в списке “Буду смотреть”.
  - 2.2. Некорректный ввод: фильм, которого нет в списке “Буду смотреть”.
3. Если 2.1: бот выводит информационное сообщение [“Фильм {Название фильма} удален из списка “Буду смотреть”.”]
4. Если 2.2: бот выводит информационное сообщение [“Фильм {Название фильма} не найден. Из списка “Буду смотреть” ничего не удалено.”]
5. Возврат в состояние `/watchlist`.

# Глоссарий

## 1. Общие термины

- 1.1. Movie Recommendation Bot – Telegram-бот для рекомендации фильмов на основе предпочтений пользователя, случайного выбора или фильтрации по параметрам.
- 1.2. Telegram API – API мессенджера Telegram, используемое для взаимодействия с ботом.
- 1.3. Java 23 – версия языка программирования Java, используемая в проекте.
- 1.4. Spring 6.2.3 – фреймворк для разработки приложений на Java (включает WebFlux, JOOQ, Spring Rest Docs).
- 1.5. PostgreSQL – реляционная база данных, используемая для хранения информации о пользователях, фильмах и предпочтениях.
- 1.6. Docker – платформа для контейнеризации приложения.
- 1.7. Docker Compose – инструмент для определения и запуска многоконтейнерных приложений.
- 1.8. Fat JAR – исполняемый JAR-файл, содержащий все зависимости приложения.
- 1.9. GitHub – платформа для хранения кода и контроля версий.
- 1.10. README.md – файл с инструкциями по настройке, запуску и использованию проекта.
- 1.11. Google Java Style Guide – руководство по стилю кода, используемое в проекте.
- 1.12. Модульные тесты – автоматические тесты для проверки отдельных компонентов кода.
- 1.13. Валидация ввода – проверка корректности данных, введенных пользователем.
- 1.14. Обработка ошибок – механизмы реагирования на исключительные ситуации.
- 1.15. Инкапсуляция – принцип ООП, ограничивающий доступ к внутренней реализации компонентов.
- 1.16. Принципы чистого кода – правила написания читаемого и поддерживаемого кода.
- 1.17. БД пользователей - база данных с пользователями бота и их метаданными, хранящаяся на сервере бота. Метаданные включают в себя список "Буду смотреть", список просмотренных

фильмов: название фильма, оценка фильма(если есть), отзыв на фильм(если есть).

- 1.18. БД фильмов - внешняя база данных, к которой происходит обращение через API. Эта база данных используется исключительно в качестве источника данных.

## 2. Роли

- 2.1. Пользователь – стандартный участник, имеющий доступ к основным функциям бота (поиск фильмов, управление списками, отзывы).
- 2.2. Администратор – пользователь с расширенными правами (доступ к служебным командам).

## 3. Авторизация и управление доступом

- 3.1. telegramId – уникальный идентификатор пользователя в Telegram, используемый для авторизации.
- 3.2. /healthcheck – команда для проверки состояния сервера (доступна без авторизации).

## 4. Поиск фильмов

- 4.1. Случайный фильм – фильм, выбранный случайным образом без учета предпочтений и фильтров.
- 4.2. Фильтры поиска – критерии для отбора фильмов (жанр, актеры, год выпуска, рейтинг, длительность, страна).
- 4.3. Жанры – категории фильмов (например, комедия, драма, фантастика).
- 4.4. Актеры – список главных актеров фильма.
- 4.5. Год выпуска – диапазон годов, в которые были выпущены фильмы.
- 4.6. Страна – страна/ы производства фильма.
- 4.7. Вещественное число рейтинга - число в формате Ц.Ц, где Ц-цифра.

## 5. Списки и оценки

- 5.1. Предпочтения - список предпочтений пользователя по категориям (жанры, актеры, страны, годы выпуска)
- 5.2. Список "Буду смотреть" – коллекция фильмов, отложенных пользователем для просмотра.
- 5.3. Список просмотренных – фильмы, отмеченные пользователем как просмотренные.
- 5.4. Отзыв – текстовый комментарий пользователя к просмотренному фильму.



- 5.5. Оценка – пользовательский рейтинг/пользователь выставляет просмотренному фильму число от 0 до 10, где 0 - очень плохое кино, 10 - отличное.