Федеральное агентство связи (Россвязь)

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра Прикладной математики

и кибернетики

Курсовая работа

по дисциплине

«Визуальное программирование и человеко-машинное взаимодействие»

По теме «Фильмотека»

Выполнил: студент группы ИП-111

Тетяков И.С.

Проверил: доцент кафедры ПМ и К

Мерзлякова Е.Ю.

**Содержание**

Введение 3

ЧАСТЬ 1 4

* Анализ задач и пользователей 4
* Выбор репрезентативных задач 5
* Заимствования 7
* Черновое описание дизайна 8

ЧАСТЬ 2 10

* Описание программы 10
* Скриншоты работы программы 12

ЧАСТЬ 3 15

* CWT-анализ 15
* GOMS-анализ 18

Соответствие требованиям курсового проекта и ссылка GitHub 21

Используемые источники 22

# Введение

Целью курсового проекта является закрепление навыков, полученных в ходе изучения дисциплины «Визуальное программирование и человеко-машинное взаимодействие» и разработка приложения «Фильмотека». Для достижения поставленной цели необходимо выполнение следующих задач:

* Провести первые 4 этапа проблемно-центрированного дизайна программного продукта.
* Провести CWT-анализ разработанного интерфейса.
* Провести GOMS-анализ разработанного интерфейса.
* По результатам CWT и GOMS анализов доработать интерфейс программы и выполнить создание макета или прототипа.

# ЧАСТЬ 1

## Анализ задач и пользователей

**Цель:**

Найти двух человек, которые могут быть заинтересованы в решении предложенной задачи. Дайте их краткое описание (возраст, образование, профессия, навыки в выбранной сфере, навыки владения компьютером). Люди должны быть заинтересованы в решении предложенной задачи.

**Решение:**

Программа каталога просмотренных фильмов может заинтересовать любого: взрослого человека, ребенка, пару, который просматривает фильмы и хотел бы их как-то каталогизировать.

В качестве заинтересованных лиц были выбраны 2 следующих человека:

* Мама: возраст – 49 лет, образование – три высших, профессия – директор школы, навыки в выбранной сфере – иногда смотрит фильмы, навыки владения компьютером – средние.
* Брат: возраст – 27 лет, образование – среднее специальное, профессия – не работает официально, занимается перепродажей автомобилей. Смотрит большее количество фильмов, сериалов, аниме, мультфильмов, навыки владения компьютером – высокие.

## Выбор репрезентативных задач

**Цель:**

Определение ключевых функциональных задач и второстепенных действий для обеспечения эффективного использования разрабатываемой программы.

**Решение:**

Репрезентативные задачи:

1. Просмотр записей:

* Разработка интуитивного интерфейса для удобного просмотра подробной информации о просмотренных фильмах.

1. Добавление записей:

* Реализация удобного интерфейса, позволяющего пользователям легко добавлять новые записи о фильмах в систему.

1. Редактирование записей:

* Предоставление пользователю удобных средств для редактирования существующих данных о фильмах.

1. Удаление записей:

* Создание интуитивного интерфейса для безопасного и простого удаления записей из фильмотеки.

1. Хранение данных:

* Разработка системы сохранения данных в базе данных, обеспечивая их сохранность после завершения работы с приложением.

Второстепенные задачи:

1. Выставление даты просмотра:

* Добавление функционала для установки и отображения даты просмотра фильма.

1. Добавление комментария:

* Реализация возможности добавления комментариев пользователями о просмотренных фильмах.

1. Выход из приложения:

* Обеспечение простого и безопасного завершения работы с приложением.

Такое четкое выделение репрезентативных и второстепенных задач поможет фокусироваться на ключевых аспектах функционала и обеспечит эффективное использование программы.

## Заимствования

**Задача:**

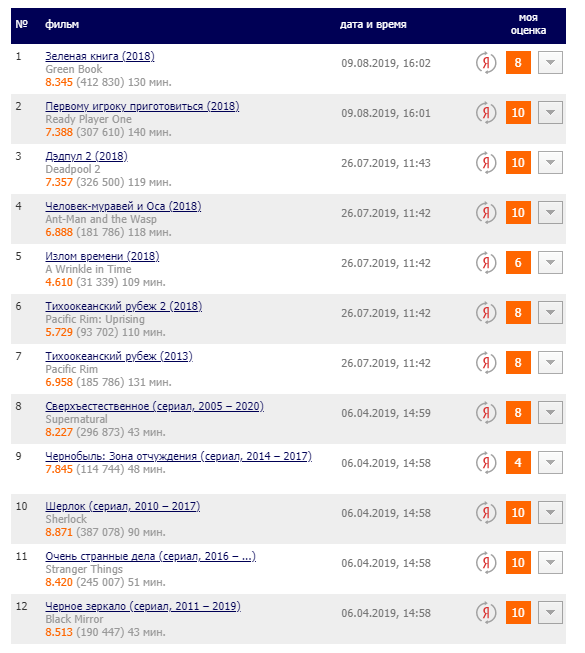
Найти приложения или сайты, с которых можно заимствовать какие-либо решения интерфейса, приведите ссылку на источник. Для обогащения пользовательского интерфейса разрабатываемого приложения, рассматривается возможность вдохновения и заимствования элементов дизайна из существующих приложений и веб-сайтов. Выберите и напишите, что именно Вы будете заимствовать из данных приложений и зачем.

**Решение:**

1. **КиноПоиск**

С данного сервиса была заимствована идея организации просмотренных фильмов в виде списка, а также дата просмотра и оценка.

Также с данного ресурса взято несколько описаний фильмов, которые будут заранее находиться в базе данных при первом запуске приложения.

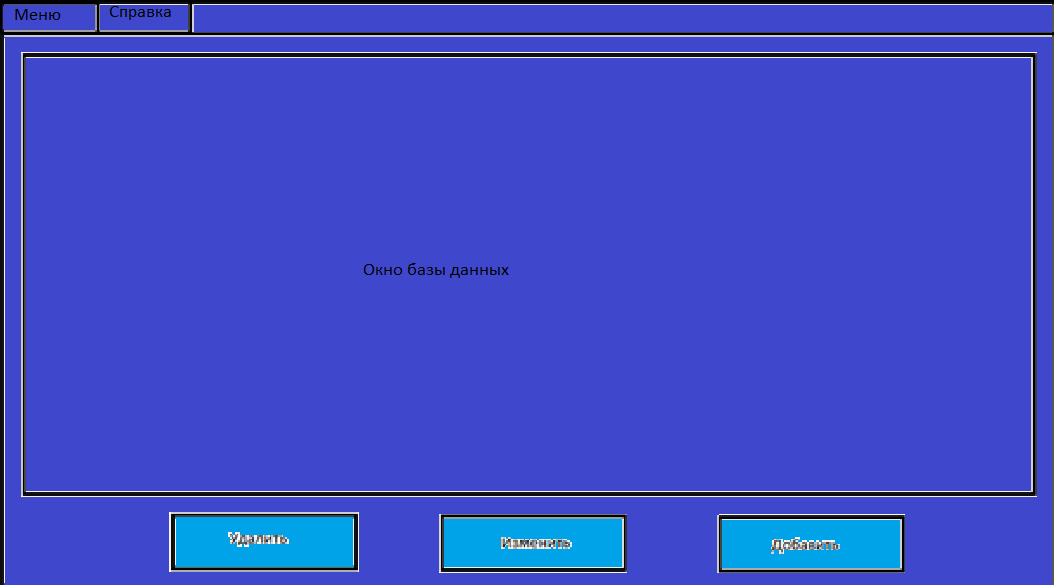


## Черновое описание дизайна

**Задача:**

Описать черновой вариант дизайна словами и графически (иллюстрации с пояснениями). Черновой вариант должен отражать все внешние элементы интерфейса и их назначение.

**Решение:**

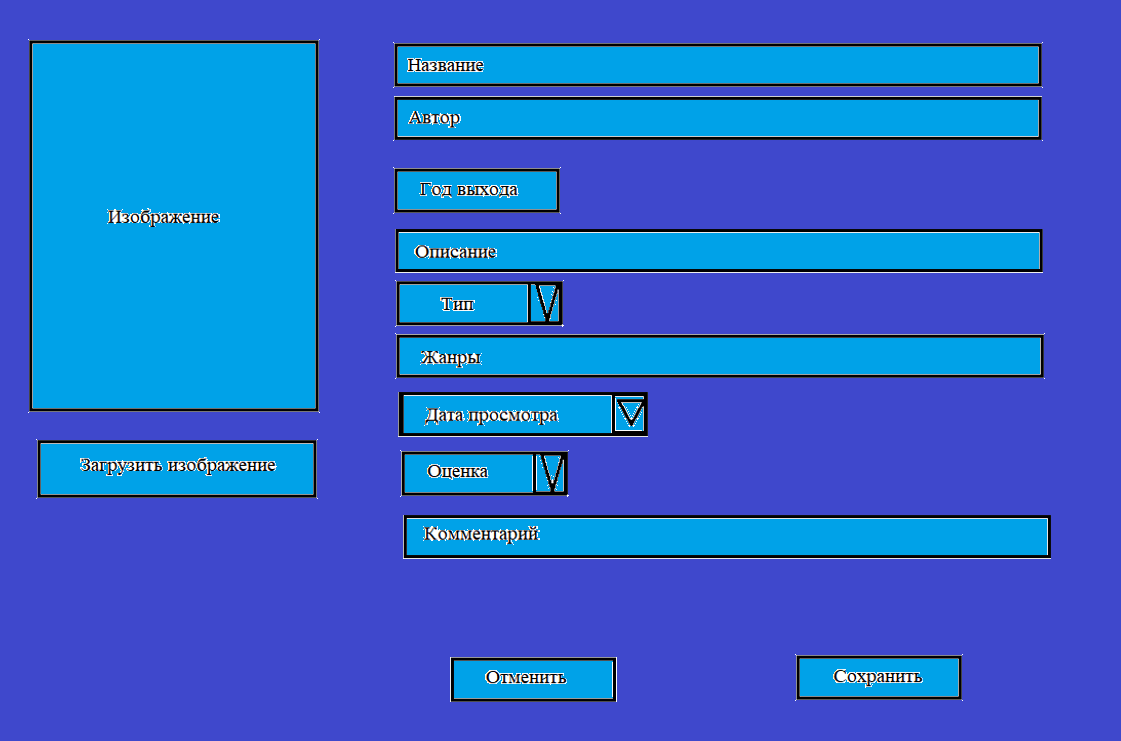


(рис. 1) Главное окно приложения

В данном приложении будет производиться отображение базы данных. При нажатии на кнопку «Изменить» должно открыться окно изменения записи. При нажатии на кнопку «Удалить» запись должна удаляться из базы. При нажатии на кнопку «Добавить» должно открываться окно создания новой записи.

Меню: содержит добавление, изменение, удаление, а также выход из программы.

Справка: содержит инструкцию пользования программой и сведения о программе



(рис. 2) Окно редактирования\создания рецепта

При создании нового рецепта пользователь должен обязательно загрузить изображение и заполнить все поля, кроме комментария. При нажатии на кнопку «Тип» должно появляться меню с доступными вариантами, одну из которых пользователь обязательно должен выбрать. При нажатии на кнопку «Оценка» должно появляться меню с доступными вариантами. При нажатии на кнопку «Дата просмотра» должен появляться календарь для выбора даты. При нажатии на кнопку «Загрузить изображение» должно открываться окно выбора изображения. При нажатии на кнопку сохранить при условии, что заполнены все поля, рецепт должен добавляться в базу данных. Если одно из полей не заполнено, должно появиться уведомление, которое не даст сохранить рецепт. При нажатии на кнопку «Отменить» окно создания рецепта закроется без сохранения написанных в нём данных.

Окно редактирования практически не отличается от окна создания, единственным отличием является то, что все данные заранее заполнены и кнопка «Сохранить» переименована в «Изменить».

# ЧАСТЬ 2

## Описание программы

Программа "Фильмотека", разработанная на платформе Qt 5.12.0, предоставляет пользователю обширный функционал:

1. Взаимодействие с фильмами:
   1. Просмотр подробной информации о фильмах.
   2. Добавление новых фильмов с удобным интерфейсом.
   3. Удаление выбранных записей из коллекции.
   4. Редактирование данных о фильмах для актуализации информации.
2. Хранение фильмов в базе данных:
3. Навигатор помощи:
   1. Подробное описание функционала приложения в удобном навигаторе.
   2. Обеспечение пользователей информацией о каждой функции для легкого освоения приложения.
4. Всплывающие уведомления-подсказки:
   1. Предоставление интуитивных и информативных всплывающих уведомлений для улучшения опыта пользователя.
   2. Поддержка обратной связи через наглядные подсказки для удобного взаимодействия.

Программа структурирована с использованием пяти ключевых классов, каждый из которых отвечает за определенный функционал:

1. **Главный класс (MainWindow):**

- Основной контроллер, обеспечивающий взаимодействие и координацию работы остальных классов.

- Управление главным окном приложения и основными действиями пользователя.

2. **Класс создания записи (AddRecord):**

- Реализация окна для добавления новых записей в фильмотеку.

- Предоставление пользователю интерфейса для ввода данных о фильме.

3. **Класс редактирования записи (EditRecord):**

- Разработка окна редактирования существующих записей в фильмотеке.

- Предоставление возможности изменения информации о фильме.

4. **Класс навигатора помощи (HelpInformation):**

- Создание окна с навигатором для ознакомления пользователя с функционалом приложения.

- Интеграция с QTextBrowser для отображения HTML-страниц с подробным описанием.

5. **Класс обработки базы данных (DataBase):**

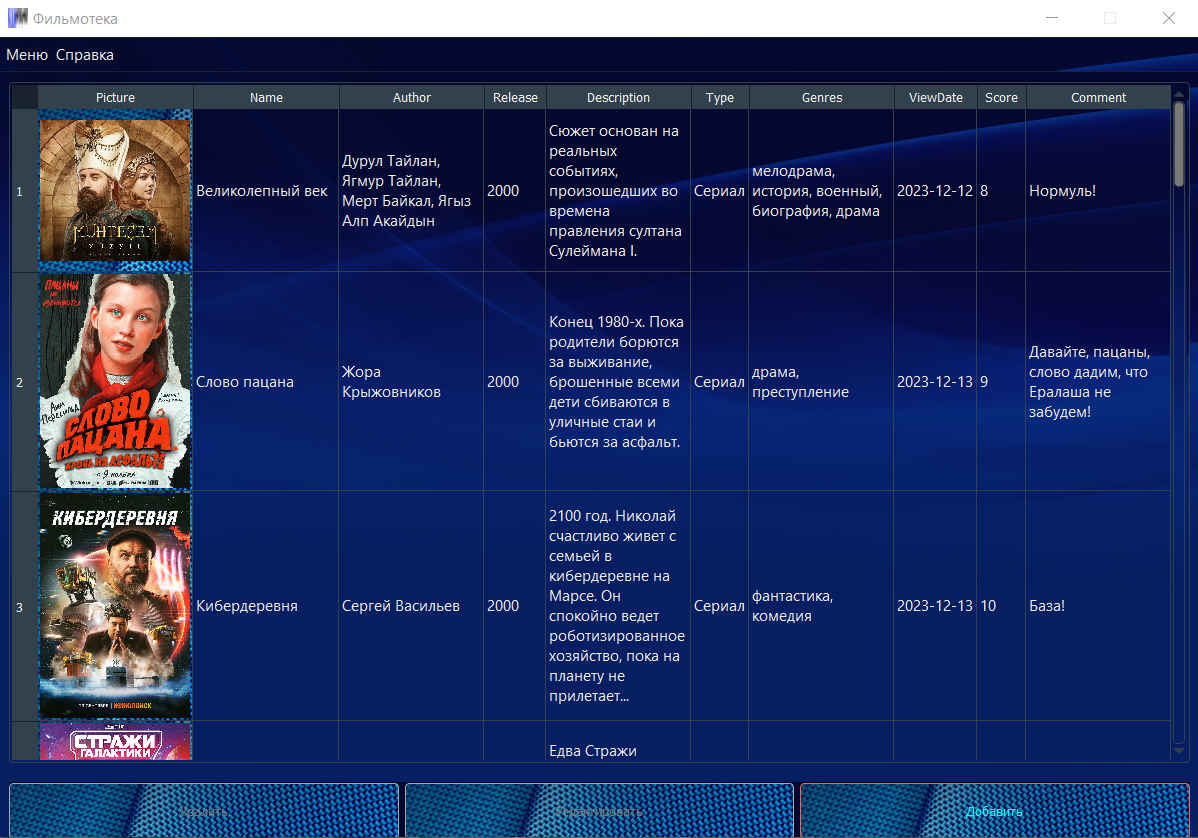
- Взаимодействие с базой данных SQLite.

База данных реализована на языке SQLite с использованием библиотеки QSql для эффективного хранения и обработки данных. Навигатор помощи предоставляет подробные инструкции в удобном формате HTML через QTextBrowser. Все элементы программы спроектированы для обеспечения легкости использования и понимания пользователем.

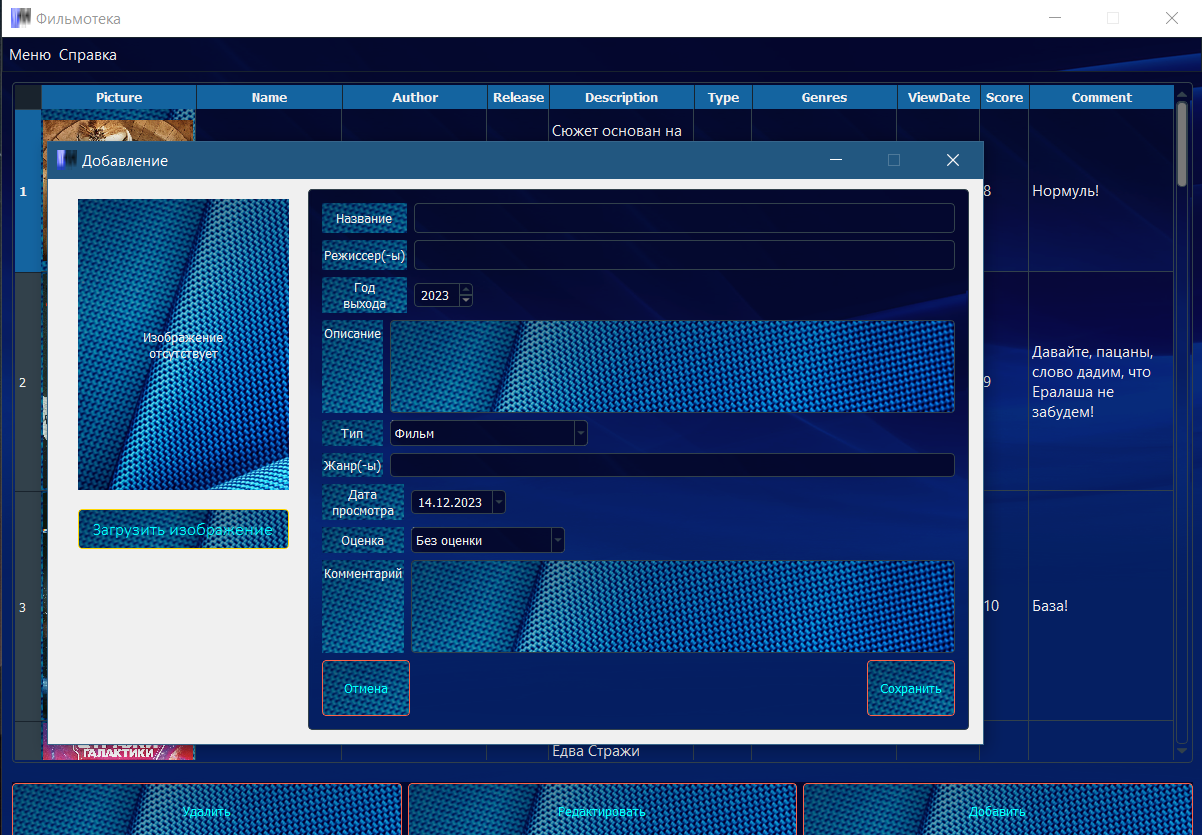
## Скриншоты работы программы



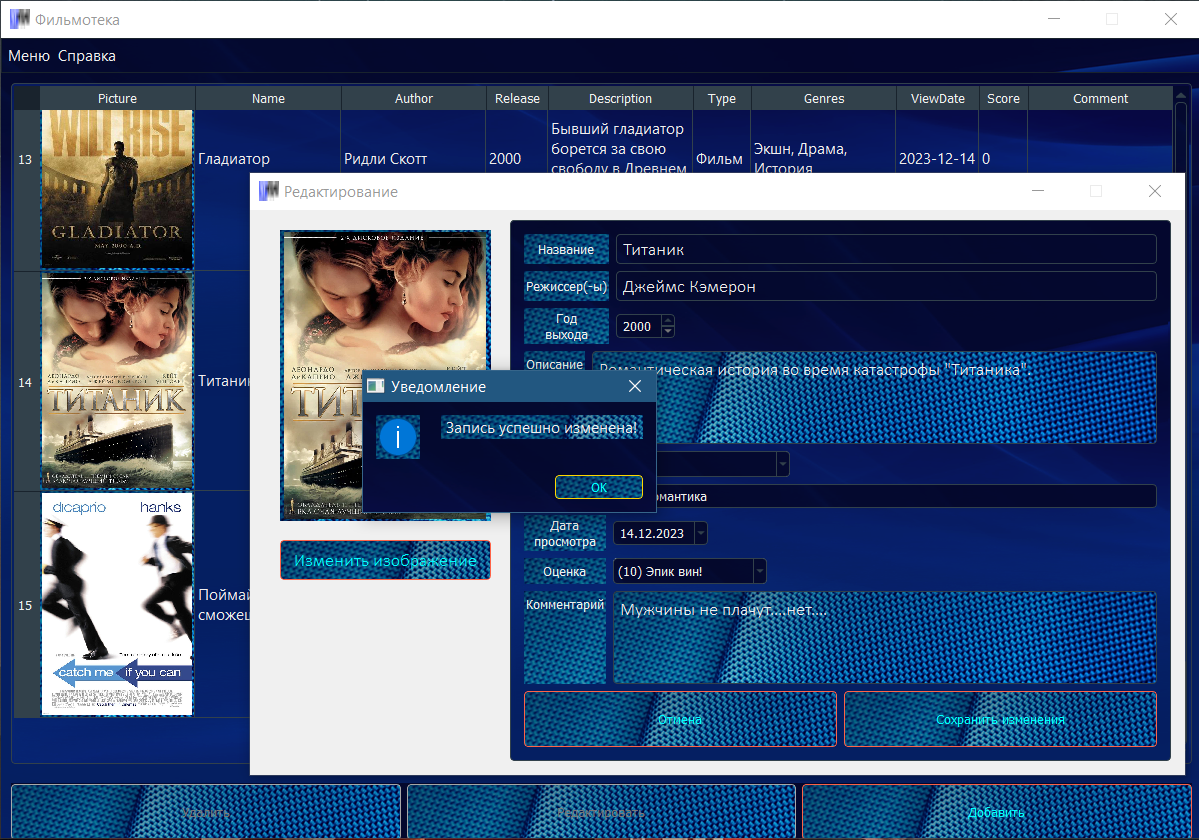
(рис. 1) Окно заставки



(рис. 2) Основное окно



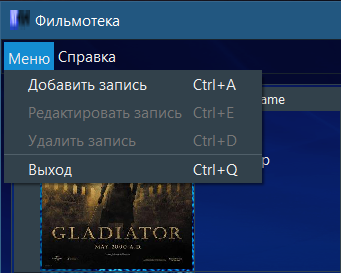
(рис. 3) Добавление записи



(рис. 4) Редактирование записи

(рис. 5) Навигатор помощи



(рис. 6) Реализация Меню и Справки

# ЧАСТЬ 3

## CWT-анализ

**Задача №1:**

Создание записи, но отмена ее добавления.

*CWT-анализ первой задачи:*

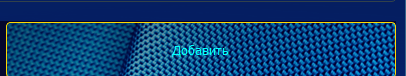
***История действий*** ***пользователя***, впервые столкнувшегося с данной задачей:

1. Запуск приложения

Пользователь двойным кликом запускает приложение и попадает в основное окно программы

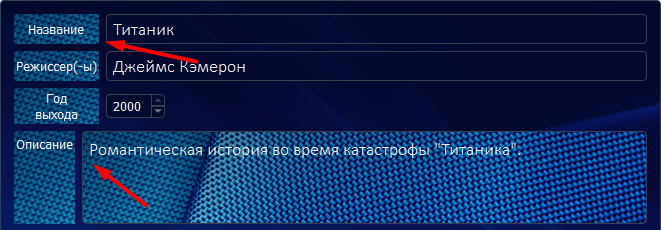
1. Нажатие кнопки «Добавить»

Пользователь нажимает на кнопку «Добавить», создается диалоговое окно формы добавления записи



1. Заполнение полей

Пользователь видит перед собой форму добавления, он заполняет поля «Название» и «Описание»



1. Нажатие кнопки «Отмена»;

Пользователь нажимает на кнопку «Отмена», диалоговое окно формы добавления закрывается



***Найденные проблемы***:

1. Отсутствует какое-либо уведомление пользователя о том, что данные будут стерты;

***Возможные способы решения:***

1. Добавление диалогового окно с информацией о том, что при нажатии «Отмена», введенные данные будут стерты;
2. Не удалять данные при закрытии несохраненной формы;

**Задача №2:**

Удаление записи.

*CWT-анализ первой задачи:*

***История действий*** ***пользователя***, впервые столкнувшегося с данной задачей:

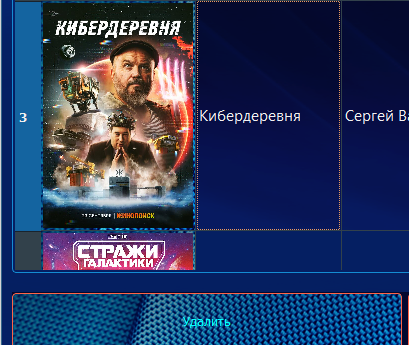
1. Запуск приложения

Пользователь двойным кликом запускает приложение и попадает в основное окно программы



1. Выделение записи

Пользователь в таблице с записями кликает на строку №1, строка выделяется, кнопка «Удалить» становится активной



1. Нажатие кнопки «Удалить»

Пользователь нажимает на кнопку «Удалить», происходит вывод диалогового информационного окна об успешном удалении записи

***Найденные проблемы***:

1. Возможно случайное удаление записи;

***Возможные способы решения:***

1. Добавление диалогового окна с подтверждением удаления записи;

## GOMS-анализ

**Описание анализа**

Практически все интерфейсные взаимодействия можно описать следующими операциями:

K – нажатие клавиши;

B – клик кнопкой мыши;

P – наведение указателя мыши;

R – ожидание ответной реакции компьютера;

H – перенос руки с клавиатуры на мышь или наоборот

D – проведение с помощью мыши прямой линии (например, выделение или прокрутка текста);

M – мыслительная подготовка (к осуществлению одной из перечисленных операций).

Разные пользователи выполняют указанные операции за разное время. Однако, GOMS исследует работу опытного пользователя. Многочисленные исследования выявили средние значения времени операций, выполняемых опытными пользователями.

* K 0.2 с
* B 0.2 с
* P 1.1 с
* H 0.4 с
* M 1.35 с

**Цель №1:** изменение записи с последующим ее сохранением

***Для выполнения цели сформулируем подцели:***

1. Выбрать интересующую запись из таблицы
2. Нажатие кнопки «Редактировать»
3. Установить указатель на поле «Название»
4. Удаление последнего символа в поле «Название»
5. Сохранение изменений

***Теперь опишем методы для каждой подцели и распишем каждый метод с точностью до операции:***

1. Выбрать интересующую запись из таблицы

**MPB (1.35 + 1.1 + 0.2 =** **2.65)**

1. Нажатие кнопки «Редактировать»

**MPB (1.35 + 1.1 + 0.2 = 2.65)**

1. Установить указатель на поле «Название»

**MPB (1.35 + 1.1 + 0.2 = 2.65)**

1. Удаление последнего символа в поле «Название»

**HK (0.4 + 0.2 = 0.6)**

1. Сохранение изменений

**MHPB (1.35 + 0.4 + 1.1 + 0.2 = 3.05)**

Итог: **(3 \* 2.65 + 0.6 +3.05) = 11,6 с**

*В качестве возможных решений*: добавить горячую клавишу для выполнения пункта «Сохранение изменений», в качестве комбинации сохранения можно использовать общепринятую «Ctrl + S».

В данном случае удалось бы добиться следующего результата:

1. Выбрать интересующую запись из таблицы

**MPB (1.35 + 1.1 + 0.2 = 2.65)**

1. Нажатие кнопки «Редактировать»

**MPB (1.1 + 0.2 = 2.65)**

1. Установить указатель на поле «Название»

**MPB (1.35 + 1.1 + 0.2 = 2.65)**

1. Удаление последнего символа в поле «Название»

**HK (0.4 + 0.2 = 0.6)**

1. Сохранение изменений

**MKK (1.35 + 0.2 + 0.2 = 1.75)**

Итог: **(2.65 + 1.3 + 0.2 + 2.65 + 0.6 +1.75) = 10.3 с**

\*(Аналогично для формы добавления)

**Цель №2:** выход из приложения

***Для выполнения цели сформулируем подцели:***

1. Навестись на кнопку «Меню»
2. Закрыть программу выбором из контекстного меню пункта «Выход»

***Теперь опишем методы для каждой подцели и распишем каждый метод с точностью до операции:***

1. Навестись на кнопку «Меню»

**MPB (1.35 + 1.1 + 0.2 = 2.65)**

1. Закрыть программу выбором из контекстного меню пункта «Выход»

**PB (1.1 + 0.2 = 1.3)**

Итог: **(2.65 + 1.3) = 3.95 с**

Время также удалось бы сократилось путем добавления общепринятой горячей клавиши «Ctrl + Q», которая бы отвечала за моментальное закрытие программы после нажатия.

Тогда было бы:

1. Закрытие программы

**MKK (1.35 + 0.2 + 0.2 = 1.75)**

Итог: **1.75 с**

## Соответствие требованиям курсового проекта и ссылка GitHub

|  |  |  |
| --- | --- | --- |
| Требования: | Выполнено: | Комментарий: |
| реализация основных задач пользователя по выбранной теме | + | Все основные задачи по теме Фильмотеки реализованы! |
| работа с файлами | + | В проекте пользователь работает с файлами, пример: изображения |
| использование SQLite | + | База данных реализована на языке SQLite, взаимодействующим с функционалом библиотеки QSql. |
| проверки на ввод некорректных значений в программе | + | Имеются. К примеру, уведомления о том, что поля остались пустыми, недопущение введения букв там, где требуются только цифры и т.д. |
| CWT, GOMS | + | Подробно описываются в отчете |
| использование таймера | ­+ | При загрузке приложения открывается заставка с иммитацией загрузки от 0 до 100 (прописывается циклом) |
| информационная наполненность | + | Добавлена подробная информация по 15+ фильмам и сериалам. |
| сохранение результатов предыдущей работы пользователя | + | Вся информация успешно сохраняется. |
| использование менеджеров компоновки на всех формах | + | Интерфейс был проработан и скомпонован с учетом всех условий, рекомендаций и эстетических соображений. |
| использование QSettings | + | QSettings используется для чтения значения из настроек при создании главного окна, как пример. |
| реализация навигатора справки | + | Реализован, имеет гиперссылки, по которым открывается HTML документ. |
| Использование отдельного файла css для всего проекта | + | style.qss позволяет прописать стили для каждого элемента в объекте, также допуская объявление его настроек при каких-то действиях (нажатие, ожидание и т.д.) |

## Исходный код проекта со всеми материалами опубликован в GitHub по ссылке: https://github.com/Diablox777/QT-Filmoteka-CursWork/tree/main

## Используемые источники

1. SQLite Viewer– использовался для просмотра базы данных в процессе разработки проекта;

URL:  [https://inloop.github.io/sqlite-viewer/](https://sqlitebrowser.org/)

1. Qt Documentation – официальная документация по разработке в среде Qt;

URL: <https://doc.qt.io/>