L.N. Gumilev Eurasian National University

Faculty of Information Technologies

"Information systems" department

**REPORT**

**«** **Object oriented programming»**

Performed by: Zhumabek A.R.

Group: IS-23

Verified by: Zhukabaeva T.K.

**Astana, 2024**

**Content**

[**1. Introduction** 3](#_Toc136840723)

[1.1. GitHub and Jira 4](#_Toc136840725)

[**2.** **Technical specification** 5](#_Toc136840727)

[2.1. Getting to know about Movie list management system 5](#_Toc136840728)

[**3.** **Realization** 6](#_Toc136840740)

[3.1. Postgresql database 6](#_Toc136840741)

[3.1. The Java programming language 9](#_Toc136840741)

[3.2. Diagram 10](#_Toc136840742)

[3.3. Overview of used libraries 12](#_Toc136840744)

[3.4. Code examples demonstrating key aspects of the implementation of Movie list management system of mechanics and functionality 13](#_Toc136840745)

[**Conclusion** 16](#_Toc136840758)

[**List of sources used** 17](#_Toc136840759)

# **Introduction**

In today's world, the film industry plays a significant role in the cultural life of society. With the emergence of a huge number of movies of various genres and formats, it is becoming increasingly difficult to organize and keep track of them. To manage the movie list and facilitate the viewing process, there is a need for software that can efficiently organize, categorize and provide information about movies.

The purpose of this report is to present a movie list management program developed in Java language and linked to Database. The program provides a user-friendly interface for adding, editing and viewing movie information. It also provides the ability to search for movies based on various criteria such as genre, year of release, and viewing status.

This report describes the program functionality, application architecture, technologies used, and plans for further improvement and development of the program.

1. Job requirements

## GitHub and Jira

The beginning of each project necessarily requires a well-crafted plan, which is why, on the recommendation of our teacher, I used Jira and also for project management, task tracking, I used GitHub.

Jira is a popular project management and bug tracking system developed by Atlassian. It is widely used for planning, task tracking, project management, and collaboration between development teams. The main functions of Jira include:

*Task Tracking*: Jira provides the ability to create tasks, describe them, assign responsibility and track the progress of work.

*Project management*: The system supports the creation of projects, defining their structure, setting deadlines and priorities.

*Reporting*: Jira provides tools for creating reports on project progress, work done, and other aspects of project management.

*Custom Workflo*ws: Jira is a flexible system that allows you to adapt task management processes to the specific needs of the team.

Jira is used in various fields, including software development, IT management, project management and other areas where effective task planning and management are needed.

Below in Figure 1.1 is a demonstration of the visual embodiment of my plans in Jira in a visual way.

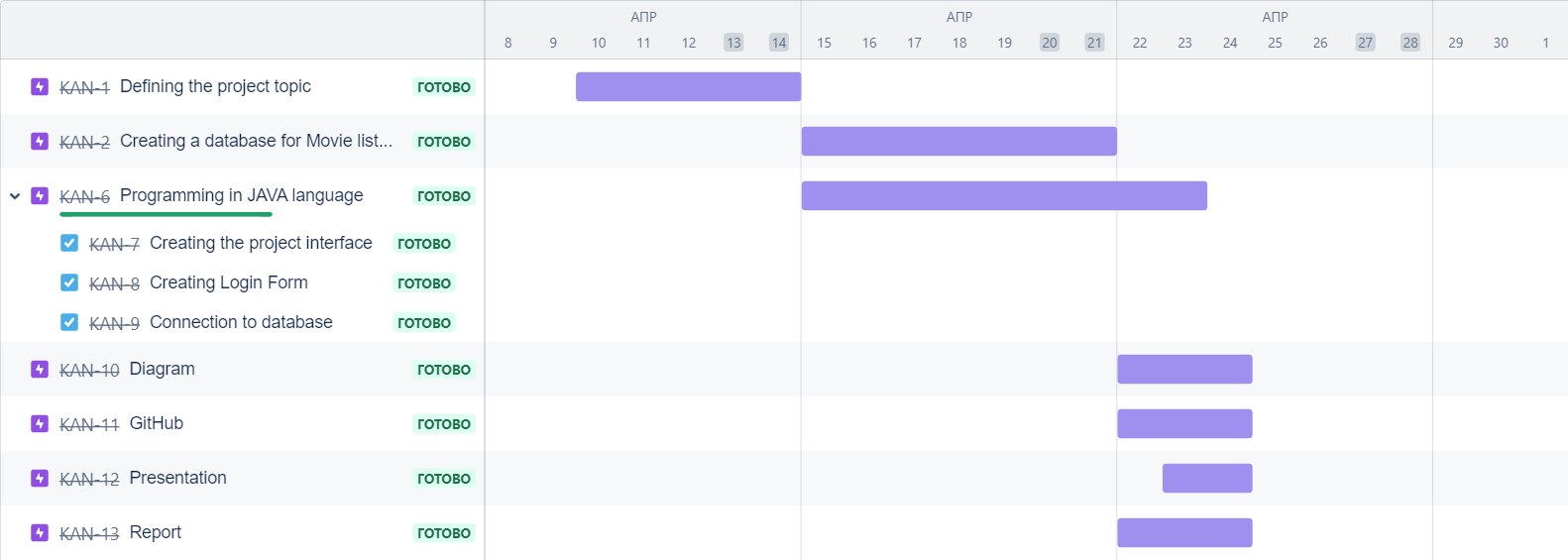


Figure 1.1. Project plan

GitHub is a web-based hosting and collaborative software development platform using the Git version control system. It provides tools for version control, project management, task tracking, and programmer collaboration. The main aspects of GitHub include:

*Version Control System (Git):* GitHub uses Git to track changes in the project code. This allows developers to work on different versions of the code, roll back to previous states, and merge their changes.

*Collaboration:* GitHub provides a means for team collaboration. Developers can make changes to the code, offer bug fixes (pull requests), discuss changes and review the code.

*Task Tracking*: Using integrated task tracking systems such as Issues, GitHub allows teams to create, discuss, and close tasks within a project.

GitHub is one of the most widely used tools for collaborative software development, and it plays a key role in the development of open and closed source software.

Below in Figure 1.2, the files that were created and saved in GitHub are provided.

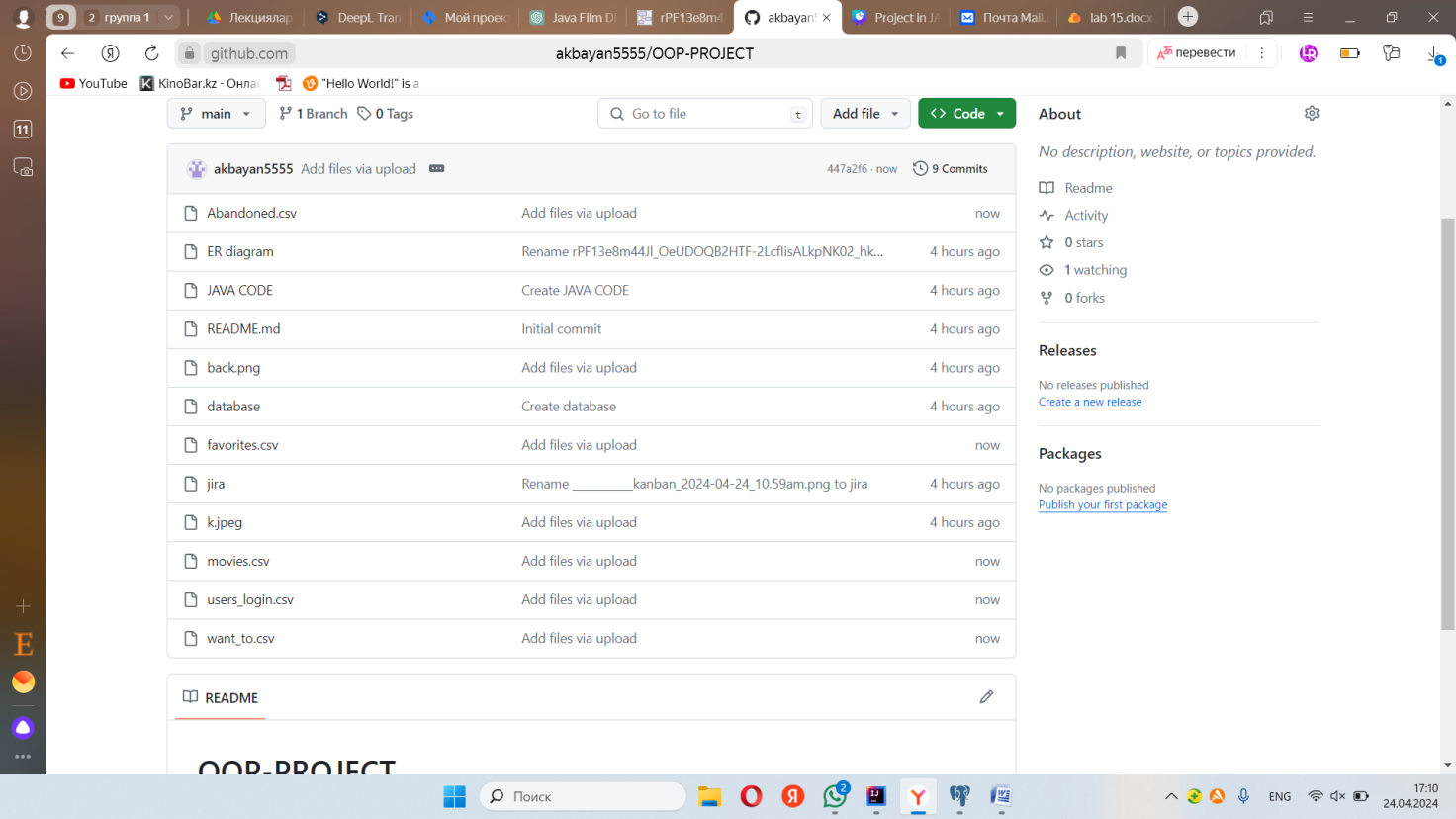


Figure 1.2. Files in GitHub.

Jira and GitHub provide reliable and convenient tools for effective project management, ensuring synchronization of the team's work and facilitating the process of joint development. Jira, an innovative project management solution, and GitHub, a popular collaborative software development platform, turned out to be the perfect combination for my needs. They have provided effective means of planning, tracking and managing the project.

# **Technical specification**

## Familiarization with the Filmograf Movie list management system

## Filmograf is a movie list management program developed in Java. It provides a user-friendly interface to organize, classify and manage your movie collection. With Filmograf, you can add new movies as well as browse your movie list by different categories and also search by different categories.

Filmograf features:

Adding movies: Filmograf allows you to easily add new movies to your collection. Just enter the movie information such as title, director, year, genre and language and the program will automatically add it to the list.

User-friendly interface: The program has an intuitive and easily customizable interface that makes managing your movie collection as easy and convenient as possible.

Filmograf will help you organize your movie collection and make your movie watching more structured and convenient.

Create a system that allows users to create a list of movies they would like to watch. Users can add new movies, mark movies they have watched, and drop them into specific categories. Use a database to store information about the movies.

# **Realization**

## Postgresql database

## PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. The origins of PostgreSQL date back to 1986 as part of the POSTGRES project at the University of California at Berkeley and has more than 35 years of active development on the core platform.PostgreSQL has earned a strong reputation for its proven architecture, reliability, data integrity, robust feature set, extensibility, and the dedication of the open source community behind the software to consistently deliver performant and innovative solutions. PostgreSQL runs on all major operating systems, has been ACID-compliant since 2001, and has powerful add-ons such as the popular PostGIS geospatial database extender.

## PostgreSQL comes with many features aimed to help developers build applications, administrators to protect data integrity and build fault-tolerant environments, and help you manage your data no matter how big or small the dataset. In addition to being free and open source, PostgreSQL is highly extensible. For example, you can define your own data types, build out custom functions, even write code from different programming languages without recompiling your database!

## PostgreSQL tries to conform with the SQL standard where such conformance does not contradict traditional features or could lead to poor architectural decisions. Many of the features required by the SQL standard are supported, though sometimes with slightly differing syntax or function. Further moves towards conformance can be expected over time. As of the version 16 release in September 2023, PostgreSQL conforms to at least 170 of the 179 mandatory features for SQL:2023 Core conformance. As of this writing, no relational database meets full conformance with this standard.

## PostgreSQL has been proven to be highly scalable both in the sheer quantity of data it can manage and in the number of concurrent users it can accommodate. There are active PostgreSQL clusters in production environments that manage many terabytes of data, and specialized systems that manage petabytes.

## Creating database for Movie list management system:

## *CREATE TABLE movies(*

## *title varchar(255),*

## *producer varchar(255),*

## *year int,*

## *genre varchar(255),*

## *language varchar(255),*

## *plot text*

## *);*

## *INSERT INTO movies (title, producer, year, genre, language, plot)*

## *VALUES* (…)

*CREATE TABLE want\_to(*

*title varchar(255),*

*producer varchar(255),*

*year int,*

*genre varchar(255),*

*language varchar(255),*

*plot text*

*);*

*INSERT INTO want\_to (title, producer, year, genre, language, plot)*

*VALUES(…)*

*CREATE TABLE favorites(*

*title varchar(255),*

*producer varchar(255),*

*year int,*

*genre varchar(255),*

*language varchar(255),*

*plot text*

*);*

*INSERT INTO want\_to (title, producer, year, genre, language, plot)*

*VALUES(…)*

*CREATE TABLE Abandoned (*

*title varchar(255),*

*producer varchar(255),*

*year int,*

*genre varchar(255),*

*language varchar(255),*

*plot text*

*);*

*INSERT INTO want\_to (title, producer, year, genre, language, plot)*

*VALUES(…)*

## The Java programming language

This part of the report uses the Java programming language using the standard Swing library to create a graphical user interface (GUI). It also used database libraries such as java.sql to connect to a PostgreSQL database, run SQL queries, and retrieve results. This code allows you to create an application to manage a list of movies, including adding new movies, editing information about existing movies, and viewing a list of movies by different categories.

## 3.3. Diagram

The entity-relationship (ER) model has been most successful as a tool for communication between the designer and the end user during the requirements analysis and conceptual design phases because of its ease of understanding and its convenience in representation [1].

ER Model Diagram: A diagram is created to display the entities, their attributes, relationships and keys. This helps to visualize the data structure and its relationships.

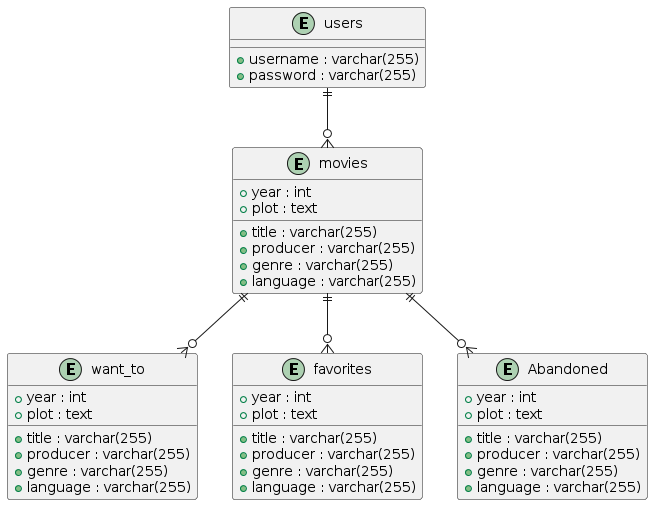


Figure 3.2. ER Model for Movie list management system

Each part of the diagram for Movie list management system:

* users: This is an entity representing the users of your system. It has two attributes: username and password.
* movies: This is an entity representing the movies on your system. It has several attributes such as title, producer, year, genre, language, and plot.
* want\_to: This is an entity representing a list of movies that the user wants to watch. It has the same attributes as the movies entity.
* favorites: This is an entity representing a list of movies that the user has added to favorites. It also has the same attributes as the movies entity.
* Abandoned: This is an entity representing a list of movies that the user has marked as abandoned. It also has the same attributes as the movies entity.

The relationship between these entities is indicated by lines. For example:

* The users entity has a one-to-many relationship with the movies entity, which means that each user can have multiple movies.
* The movies entity also has one-to-many relationships with the want\_to, favorites, and Abandoned entities, which means that each movie can belong to each of these categories.

This is an ER diagram that helps you visualize the database structure and relationships between entities in your movie list management system.

## Overview of used libraries

As part of the development of the movie management system in Java, a number of external libraries and tools were used to greatly simplify the development process and provide more efficient interaction with the database, creation of a graphical user interface, and other functionality. In this section, we will look at the key libraries, their roles in the project, and examples of their use.

The following libraries were used to develop the software:

AWT (Abstract Window Toolkit) and Swing: These Java libraries are used to create a graphical user interface (GUI). AWT provides basic components for creating the interface such as buttons, text boxes and panels, while Swing provides more advanced components and a more flexible approach to creating a GUI.

JDBC (Java Database Connectivity): It is a Java API standard for interacting with databases. It provides the ability to connect to databases, execute SQL queries and process query results.

These libraries enabled the development of software with user interface and database interaction, providing functionality for data manipulation and management. The result was a movie list management system that provides the ability to add, search, and view movie information.

Let's break them down:

Libraries:

java.awt.\*: This is a library for creating a graphical user interface (GUI) in Java.

javax.swing.\*: This is an extension of the AWT library that provides more advanced GUI components such as buttons, text boxes, and panels.

java.sql.\*: This is a library for interacting with databases in Java.

Built-in packages and classes:

java.awt.event.ActionEvent: This is a class representing an action event, such as a button click.

java.awt.event.ActionListener: This is an interface that is implemented to handle action events, such as a button press.

java.sql.PreparedStatement: This is an interface that represents a pre-compiled SQL query that can be executed multiple times with different parameters.

java.sql.ResultSet: This is an interface representing the result of a database query. It allows you to loop through the result rows and extract values from them.

java.sql.SQLException: This is a class representing an exception related to database handling.

java.sql.Connection: This is an interface representing a database connection.

java.sql.DriverManager: This is a class that provides methods for managing JDBC drivers and database connections.

These libraries and classes allow you to create user interfaces and interact with a database in Java.

## Code samples demonstrating key implementation aspects of the movie management system

This section presents code examples that demonstrate key aspects of implementing a movie management system in the Java programming language. These examples provide a better understanding of how to implement basic application functionality such as adding, searching, and displaying movie information, interacting with a database, and creating a graphical user interface.

The code examples include creating and using classes for working with a database, developing methods for performing basic CRUD (Create, Read, Update, Delete) operations with movies, and building a graphical user interface using the Swing library. Each example is a small piece of code that can be easily integrated into the overall application structure.

The following code samples will be presented to explain their operation and purpose of use within a movie management system.In Figure 3.3, Figure 3.4 below, I showed how the library's book accounting system implemented the Python code interfaces:

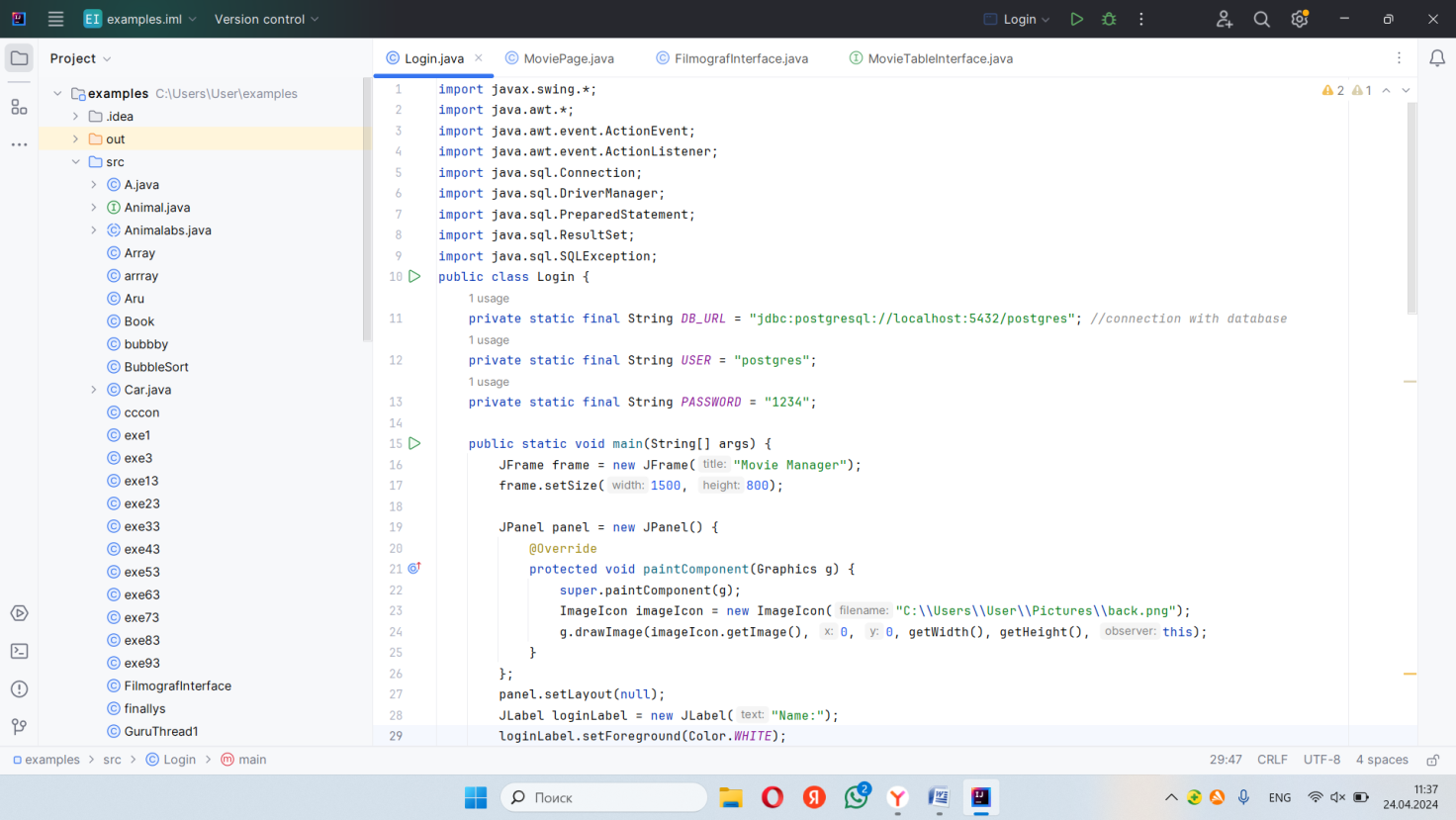


Figure 3.3 The connection with database PostgreSQL

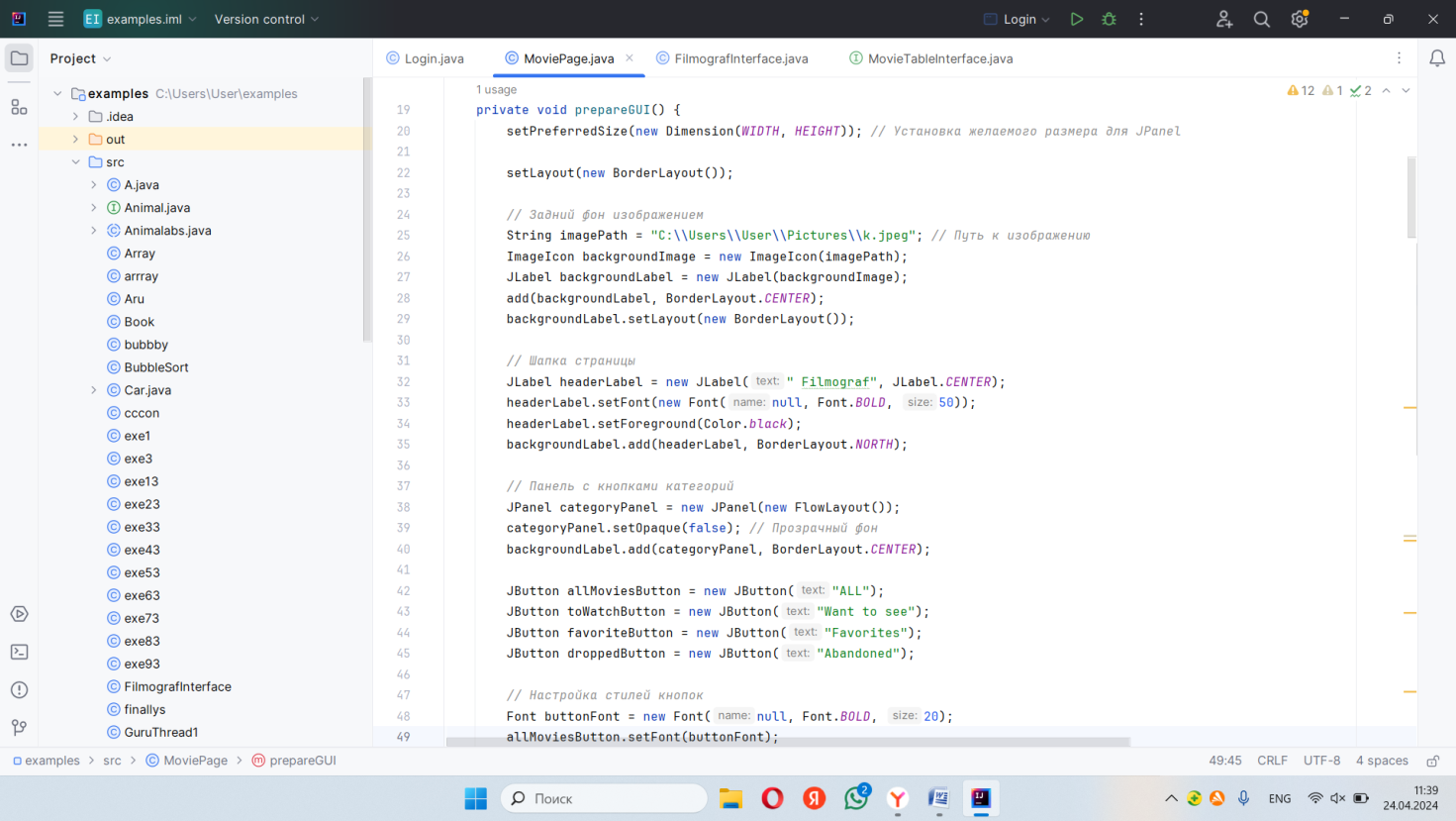


Figure 3.4 Buttons and interface

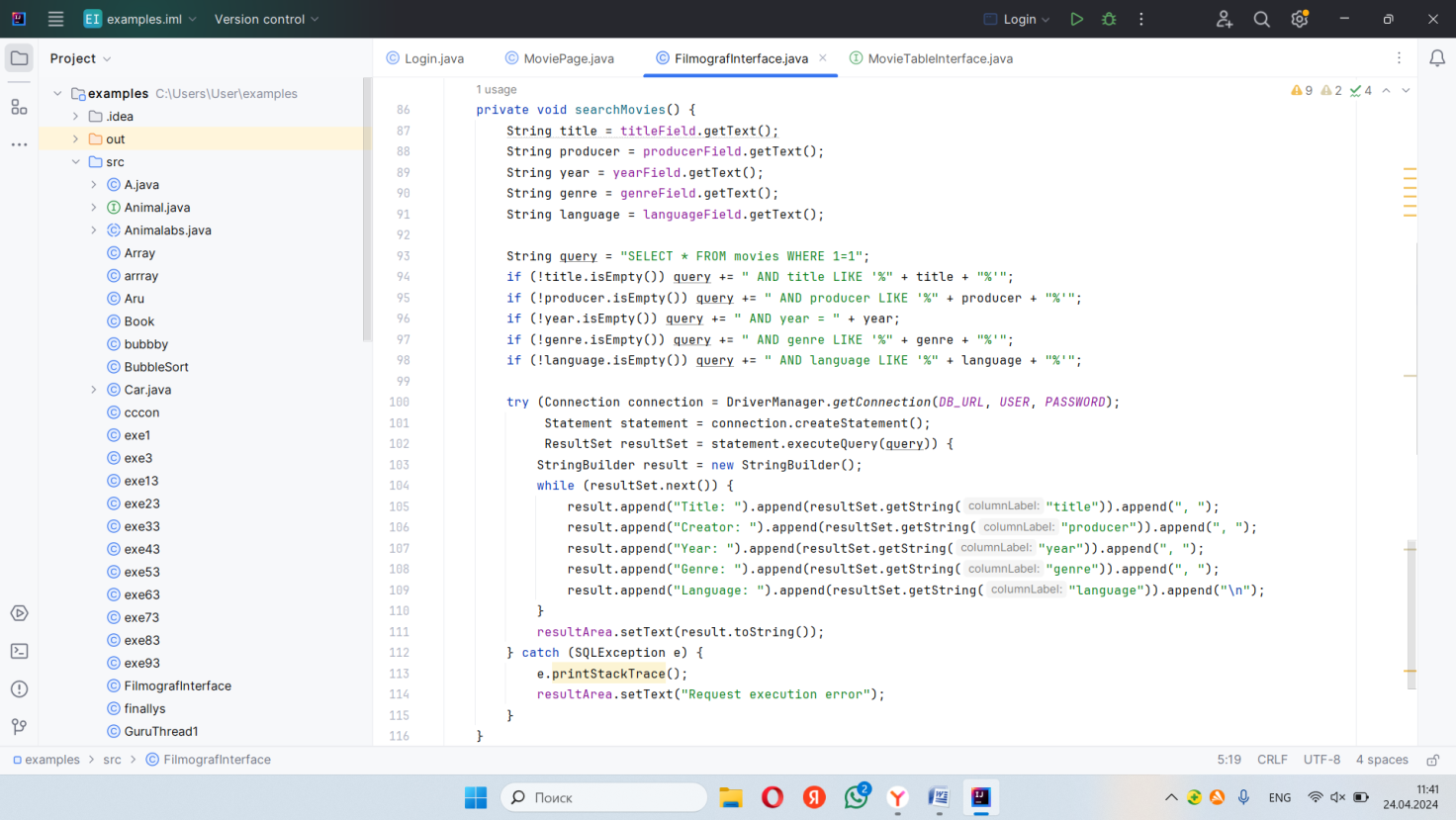


Figure 3.5 Creating a search by specific categories

**Conclusion**

In the process of developing a movie management system, key aspects of creating a system in the Java programming language using a database and graphical user interface were mastered. Within the framework of this project the functions of adding, searching and displaying information about movies, as well as interaction with the database for data storage and processing were considered and implemented.

One of the main outcomes of the project is the experience gained in developing scalable and functional applications in Java. Through this project, the understanding of application architecture, database principles, and user interface design was improved.

In addition, this system can be further developed and extended in the future. For example, it is possible to add functionality for user authorization, the ability to rate and comment on movies, as well as to implement support for different data storage formats, such as JSON or XML.

In general, the development of the movie management system allowed us to immerse ourselves in the world of creating applications in the Java language, to acquire new skills and experience, as well as to demonstrate the ability to creatively approach to problem solving.

# **List of sources used**

1. Evgeny Morgunov. Making PostgreSQL Basics of the SQL Language : tutorial.

2. Shamkant B. N. Ramez E. Fundamentals of Database System. Textbook. Seventh edition, 2016. - 1178 p.

3 Toby J. Teorey, Guangping Wei, Deborah L. Bolton, and John A. Koenig. ER Model Clustering as an Aid for User Communication and Documentation in Database Design, 1989. – 13 p.

4. https://youtu.be/-MDPYDi01dM?si=\_txUVy-o7-n6ZW7M

5. Cosentino V., Luis J., Cabot J. Findings from GitHub: methods, datasets and limitations //Proceedings of the 13th International Conference on Mining Software Repositories. – 2016.

6. Martin Fowler. UML Basics. A quick guide to the standard Object Modeling Language.