



UDC 004.4

# APPLICATION TO AUTOMATE THE SALE OF TICKETS IN A LOCAL FILM THEATER

## ДОДАТОК ДЛЯ АВТОМАТИЗАЦІЇ ПРОДАЖУ КВИТКІВ У МІСЦЕВОМУ КІНОТЕАТРИ

Kolesnikov V.A.

Колесніков В.А.

Sumy State University, Sumy, Ukraine

ORCID: <https://orcid.org/0000-0002-1991-3614>

E-mail: [v.kolesnikov@cs.sumdu.edu.ua](mailto:v.kolesnikov@cs.sumdu.edu.ua)

Copyright © 2025 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: [10.15673/atbp.v17i3.3210](https://doi.org/10.15673/atbp.v17i3.3210)

**Abstract.** Digital automation is transforming ticket sales in local film theaters, enhancing efficiency and customer satisfaction. This paper presents a Java-based software application developed for a local cinema's cashier workstation to streamline ticket sales. Leveraging MySQL for robust data storage, AWT/Swing for an intuitive graphical interface, and iTextPDF for ticket generation, the system enables cashiers to view film schedules, check real-time seat availability, select seats, and produce PDF tickets containing essential details such as film name, date, time, seat row/number, price, and a detachable control section.

The application supports dynamic seat availability checks, multiple seat reservations, and database updates to prevent overbooking. Tickets are generated as PDFs and saved to user-specified directories. Rigorous testing validated functionality, addressing edge cases like missing database connections or full occupancy. The system reduces transaction time and eliminates allocation errors, offering a user-friendly interface for non-technical staff.

Compared to costly commercial systems, this offline, open-source solution is cost-effective for small theaters. Limitations include the absence of online payment integration and potential PDF truncation for lengthy film names. Future enhancements could incorporate networked operations and analytics. This project demonstrates affordable automation for local cinemas, improving operational efficiency and user experience.

**Анотація.** Цифрова автоматизація перетворює продаж квитків у місцевих кінотеатрах, підвищуючи ефективність та задоволеність клієнтів. Ця стаття представляє програмний додаток на основі мови програмування Java, розроблений для робочої станції касира в місцевому кінотеатрі, щоб спростити продаж квитків. Використовуючи MySQL для надійного зберігання даних, AWT/Swing для інтуїтивного графічного інтерфейсу та iTextPDF для генерації квитків, система дозволяє касирам переглядати розклади фільмів, перевіряти наявність місць у реальному часі, обирати місця та створювати PDF-квитки з ключовими деталями, такими як назва фільму, дата, час, ряд/номер місця, ціна та відривна частина для контролю.

Додаток підтримує динамічну перевірку наявності місць, бронювання кількох місць та оновлення бази даних для запобігання надмірному бронюванню. Квитки генеруються у форматі PDF та зберігаються в директоріях, обраних користувачем. Ретельне тестування підтвердило функціональність, враховуючи крайні випадки, такі як відсутність з'єднання з базою даних або повна заповненість. Система скорочує час транзакцій та усуває помилки розподілу, пропонуючи зручний інтерфейс для нетехнічного персоналу.

Порівняно з дорогими комерційними системами, це автономне, відкрите рішення є економічним для малих кінотеатрів. Обмеження включають відсутність інтеграції з онлайн-платежами та можливе скорочення довгих назв фільмів у PDF. Майбутні вдосконалення можуть включати мережеві операції та аналітику. Цей проєкт демонструє доступну автоматизацію для місцевих кінотеатрів, покращуючи операційну ефективність та користувацький досвід.

**Keywords:** cinema ticket automation, seat reservation system, Java, MySQL, Swing GUI, iTextPDF.

**Ключові слова:** автоматизація продажу квитків у кінотеатрі, система бронювання місць, Java, MySQL, Swing, iTextPDF.

### 1. Introduction

In the modern entertainment industry, cinemas play a pivotal role in providing cultural and recreational experiences to audiences. Digital automation is transforming ticket sales in local film theaters, enhancing efficiency and customer



satisfaction. This paper presents the development of a software application designed to automate ticket sales in a local cinema, focusing on a cashier's workstation.

The application is built using Java for object-oriented programming, MySQL for data storage, and integrates graphical user interfaces (GUIs) via AWT and Swing, along with PDF generation for tickets using iTextPDF. This project addresses the need for a user-friendly system that allows cashiers to view film schedules, check seat availability, select seats, and generate printable tickets. By automating these tasks, the system reduces human error, speeds up transactions, and provides a seamless experience for both staff and patrons.

The relevance of this work stems from the growing demand for digital solutions in small-scale entertainment venues, where resource constraints often limit the adoption of enterprise-level systems. This study is an example of types of projects the author uses in teaching computer science courses, emphasizing practical implementation in a real-world scenario. The paper explores the project's design, development, testing, and outcomes, highlighting its potential for local cinemas.

## 2. Literature review and problem statement.

The literature on cinema management systems reveals a progression from manual ticketing to sophisticated digital platforms. Early studies, such as those in [1], emphasize the role of information systems in service industries, where automation improves data accuracy and operational speed. In the context of cinemas, research in [2, 3] highlights how online ticketing systems, like those used by major chains (e.g., AMC Theatres), integrate databases for real-time seat availability and payment processing. However, these systems are often cloud-based and expensive, making them unsuitable for local theaters with limited budgets.

Java-based applications have been widely discussed for their platform independence and robustness in GUI development. [4] details Swing and AWT frameworks for creating interactive interfaces, which are essential for user-centric applications like ticket sales. Database integration is another key area; [5] underscores MySQL's suitability for relational data management in small-to-medium enterprises due to its open-source nature and ease of use. PDF generation tools like iTextPDF are explored in [6], who founded the library, demonstrating its effectiveness for formatted document creation in Java environments.

Existing solutions, such as open-source projects on GitHub (e.g., Cinema Booking System by various contributors), often include features like seat mapping and reservations but lack customization for local needs, such as offline operation or simple PDF ticketing. Commercial software like Vista Cinema or RTS (Ready Theatre Systems) provides comprehensive features but requires significant investment and infrastructure.

The problem formulation arises from these gaps: Local film theaters face challenges in adopting affordable, easy-to-deploy systems that handle ticket sales without internet dependency. Manual processes lead to overbooking, lost revenue from unsold seats, and poor customer experiences. The core issue is the absence of a tailored, desktop-based application that integrates database management, GUI for seat selection, and printable tickets. This project addresses this by developing a standalone Java application, ensuring data persistence in MySQL and secure, offline functionality.

## 3. Purpose and objectives of the study.

The primary purpose of this study is to develop and evaluate a software application that automates ticket sales in a local film theater, specifically for the cashier's computer. This aims to streamline the process of viewing film information, selecting available seats, purchasing tickets, and generating printable receipts, thereby enhancing efficiency and user experience.

To achieve this purpose, the following objectives were set:

- Analyze requirements for a cinema ticketing system, including data storage, user interface, and output formats.
- Design the system's architecture, including use case diagrams, class diagrams and database schema (ER diagram).
- Implement the application using Java, integrating MySQL for data management, Swing/AWT for GUI, and iTextPDF for ticket generation.
- Develop features for viewing films, selecting halls and seats, reserving multiple seats, and printing tickets as PDFs with details like film name, date, time, seat row/number, price, and a tear-off control section.
- Conduct thorough testing to ensure functionality, error handling, and usability.
- Document the system, including code comments, user manual, and testing reports.
- Assess the system's performance against initial requirements.

These objectives ensure a comprehensive approach, from conceptualization to deployment, tailored to a local cinema's operational needs.

## 4. Methods and materials of research.

This research employed a methodological framework combining software engineering principles with experimental development. The study was conducted as a project-based inquiry, utilizing object-oriented programming paradigms in Java (version 22). The development process followed an iterative model: requirements gathering, design, implementation, testing, and documentation.

Materials included:

- Hardware: Standard desktop computers for development and testing.



- Software Tools: Java Development Kit (JDK) for coding; MySQL 8.0.3 for the database; DBeaver for database administration; iTextPDF 8.0.3 for PDF generation; and Integrated Development Environment (IDE) like IntelliJ IDEA.  
 - Database: A MySQL database named "cinema" with tables for films, halls, positions (seats), sessions, and reservations. The schema used utf8mb4\_general\_ci encoding, with connections via localhost:3306, user "root" (no password for development).

Methods involved:

- Design Phase: Creation of UML diagrams (use case, class, ER) to model system interactions, entities, and relationships.
- Implementation: Coding in Java, with classes such as Application (entry point), MySQLConnector (database handling), MainWindow (GUI), Cinema (data holder), Film, Hall, Seat, Session, Ticket, and TicketPrinter.
- GUI Development: Using Swing for components like JComboBox (for film/date/time/hall selection), JTable (for seat display), and JButton (for printing).
- Database Operations: SQL queries for selecting halls, films, sessions, and reserved seats; inserting reservations.
- Testing: Manual functional testing, including edge cases like empty tables (all seats occupied), missing files, or invalid inputs. A testing report documented 17 test cases, covering startup, UI interactions, and error scenarios.
- Documentation: Javadoc-style comments in code; user manual.

The experimental part involved simulating cinema operations: loading data from MySQL, user interactions via GUI, and generating PDFs. Ethical considerations included ensuring no unauthorized data access.

## 5. Implementation of Application and Results.

### 5.1. Screen Design.

During the User Interface design phase, we developed the prototype of the main screen shown in Figure 1. The screen is divided into seat table and control panels. The control panel contains drop down fields for film selection and a button to print tickets. Ticket layout is shown in Figure 2.

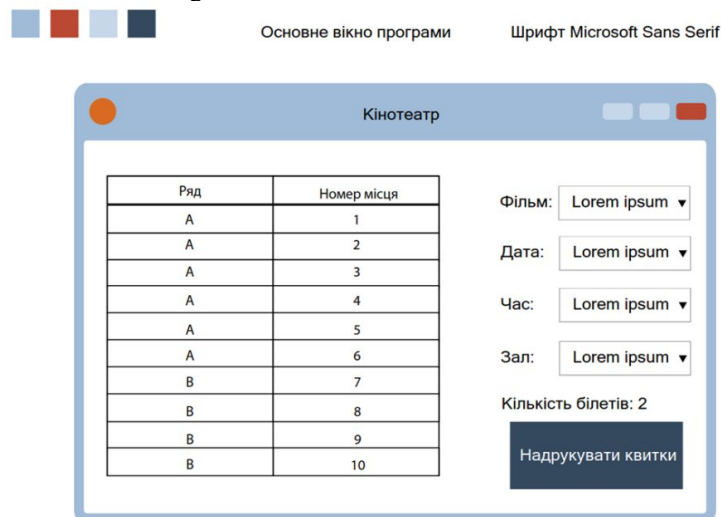


Fig. 1. Screen design

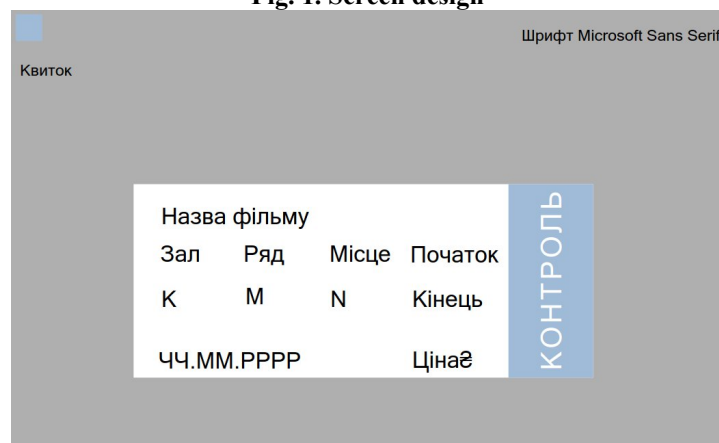


Fig. 2. Ticket layout

### 5.2. UML use-case diagram.

After the requirement gathering phase, the requirements have been analyzed and the use-case diagram was created for application possible use cases (see Figure 3). Main use cases include movie selection, seat selection and ticket printing.

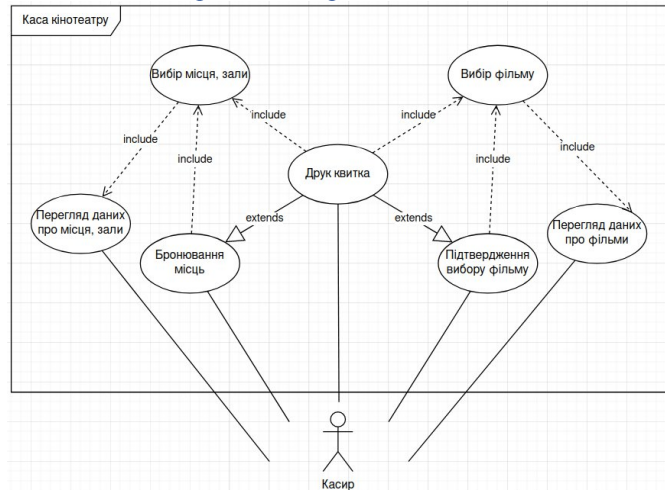


Fig. 3. Use-case diagram

5.3. UML class diagram.

After the requirements have been analyzed for possible classes, the user interface screen model and the use case diagram created, we identified the classes and the relationships among them shown in Figure 4.

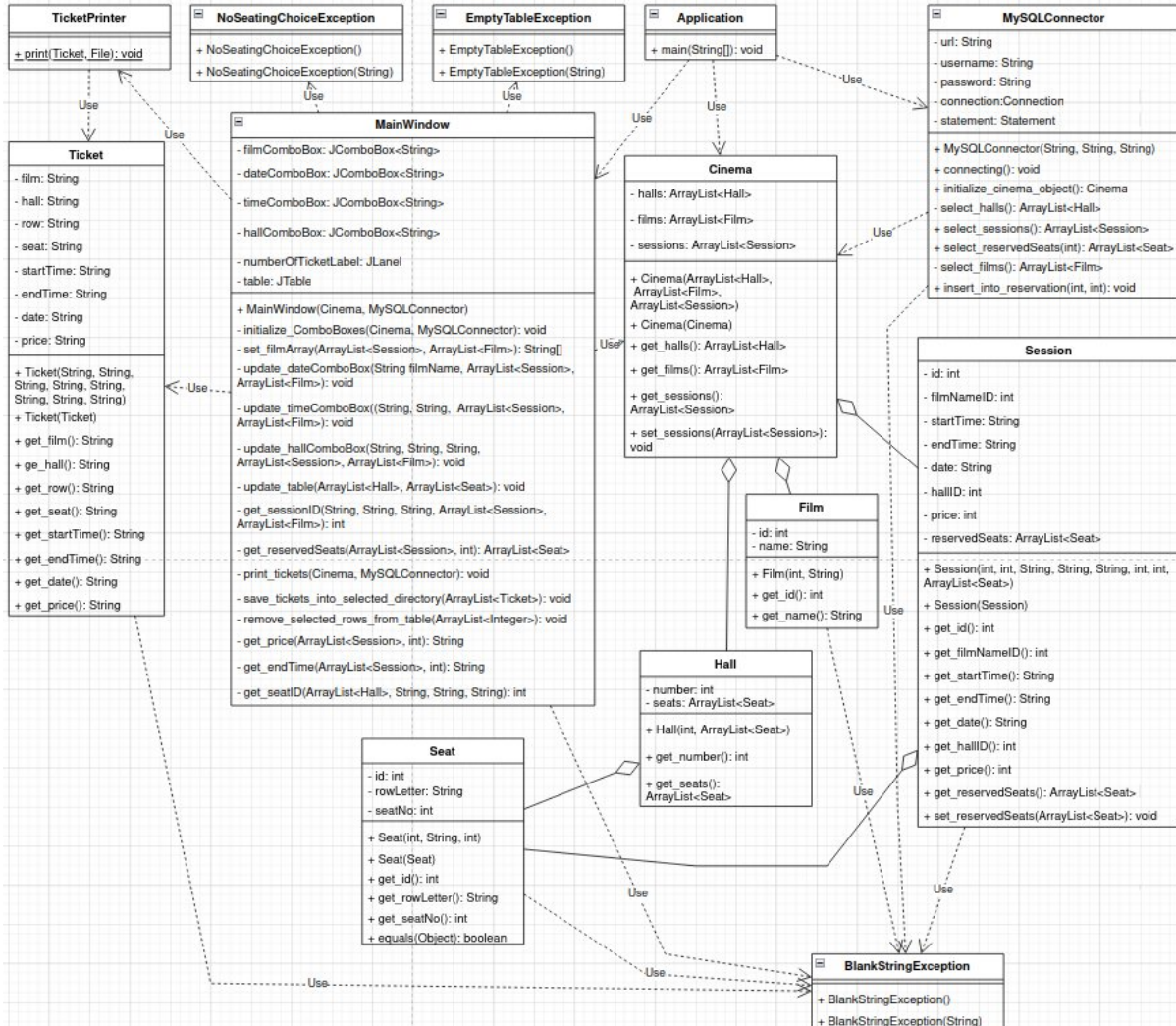


Fig. 4. UML class diagram

5.4. Screen shots of application at work.

Next, we show some screen shots of the application during its work. Figure 5 shows the main window after the application has been started. The screen is divided into seat table and control panels. The control panel contains drop down fields for film selection and a button to print tickets.



The initial screen of the 'Кінотеатр' application. It features a table on the left with columns 'Ряд' (Row) and 'Номер місця' (Seat Number). The table contains one row with 'Ряд' set to 'A' and 'Номер місця' set to '7'. To the right of the table are several input fields: 'Фільм:' (Movie) with a dropdown menu showing 'Метро 2033'; 'Дата:' (Date) with a dropdown menu showing '21.03.2024'; 'Час:' (Time) with a dropdown menu showing '17:30'; 'Зал:' (Hall) with a dropdown menu showing '1'; and 'Кількість квитків:' (Number of tickets) with a text input field containing '0'. At the bottom right, there is a dark blue button labeled 'Надрукувати квитки' (Print tickets).

**Fig. 5. Initial screen**

Figure 6 shows movie selection and Figure 7 shows the resulting screen.

The movie selection screen. The table on the left is populated with rows from 'A' to 'C' and seat numbers from 1 to 25. The 'Фільм:' dropdown menu is open, showing a list of movies: 'Дорога', 'Метро 2033', 'Пірати Карибського моря: Прокляття Чор', 'Володар пернів', and 'Хобіт'. The 'Дата:' dropdown menu shows '27.03.2024', 'Час:' shows '17:40', and 'Зал:' shows '1'. The 'Кількість квитків:' field is '0'. The 'Надрукувати квитки' button is visible at the bottom right.

**Fig. 6. Movie selection**

The screen after movie selection. The 'Фільм:' dropdown menu is now closed and shows 'Дорога'. The 'Дата:' dropdown menu shows '27.03.2024', 'Час:' shows '17:40', and 'Зал:' shows '1'. The 'Кількість квитків:' field is '0'. The 'Надрукувати квитки' button is visible at the bottom right.

**Fig. 7. Sample screen after movie selection**

Figure 8 shows seat selection and Figure 9 shows a ticket example.

The seat selection screen. The table on the left shows rows 'A' and 'B' with seat numbers 4, 7, 8, 9, 10, 16, 17, and 18. Rows 'B' and seat '17' are highlighted in blue. The 'Фільм:' dropdown menu shows 'Дорога', 'Дата:' shows '21.03.2024', 'Час:' shows '18:30', and 'Зал:' shows '2'. The 'Кількість квитків:' field is '2'. The 'Надрукувати квитки' button is visible at the bottom right.

**Fig. 8. Seat selection**



Метро 2033				КОНТРОЛЬ
Зал	Ряд	Місце	17:30	
1	A	5	18:25	
21.03.2024		100₴		

**Fig. 9. Ticket example**

The developed application successfully automates ticket sales, meeting all specified requirements. Key results include:

- System Architecture: The ER diagram models entities like films (id, name), halls (id), positions (id, row\_letter, seat\_no, id\_hall), sessions (id, id\_film\_name, start\_time, end\_time, date, id\_hall, price), and reservations (id\_session, id\_seat). The class diagram reflects Java classes with appropriate relationships (e.g., Cinema aggregates Halls, Films, Sessions).
- GUI Implementation: The main window features dropdowns for film, date, time, and hall selection. A table displays available seats (row letter and number). Selecting rows updates a ticket count label. The "Print Tickets" button prompts for a save directory and generates PDFs.
- Functionality:
  - Data is fetched from MySQL upon startup, initializing a Cinema object.
  - Dropdowns dynamically update based on selections (e.g., dates filter by film).
  - Seats are marked as reserved in the database upon purchase, removing them from the table.
  - Tickets are PDFs with fields for film name, hall, row/seat, start/end time, date, and price (in UAH). Long film names are truncated if necessary.
- Code Highlights: The MySQLConnector class handles connections and queries (e.g., select\_halls(), insert\_into\_reservation()). MainWindow manages UI events, such as updating tables and printing. TicketPrinter uses iTextPDF to fill form fields in a template (empty\_ticket.pdf).
- Testing Outcomes: All tests passed, confirming robustness. For instance, launching without MySQL yields a connection error; selecting seats and printing creates valid PDFs; empty tables trigger "All seats occupied" messages.

The application occupies minimal resources, runs offline, and produces formatted tickets.

#### The discussion of the results.

The results demonstrate a functional, efficient system for local cinemas, aligning with literature on automation benefits (e.g., reduced errors per [2]). Strengths include offline capability, simple GUI, and PDF integration, making it accessible for small theaters unlike cloud-based alternatives [2]. The use of MySQL ensures scalable data management, while Java's OOP enhances maintainability.

Limitations exist: The system lacks online payments or mobile integration, focusing solely on cashier use. Long film names cause truncation in PDFs, suggesting dynamic resizing. Testing was manual and limited to developers; broader user trials could reveal usability issues. Compared to commercial systems like RTS, this app is cost-free but less feature-rich (no analytics or multi-user support).

Future enhancements could include web integration or advanced queries for analytics. The project validates the feasibility of open-source tools for niche applications, contributing to educational and practical software development in cinema management.

#### Conclusions

This study successfully developed an application to automate ticket sales in a local film theater, achieving the purpose of enhancing efficiency through digital means. Objectives were met: a robust design, implementation in Java/MySQL, comprehensive testing, and detailed documentation. The system provides a GUI for seat selection, real-time availability, and PDF tickets, addressing key problems in manual processes.

Key achievements include seamless database integration, user-friendly interface, and error handling. While limitations like offline-only operation persist, the application offers a viable, affordable solution for local cinemas. Future work could expand to networked environments.

#### References

- [1].Laudon, K. C., & Laudon, J. P. (2019). Management Information Systems: Managing the Digital Firm. Pearson.
- [2].Mohammed, M., et al. (2024). "Digital Transformation in the Customer Experience." DOI:10.1201/9781003560449. ISBN: 9781003560449.
- [3]. Solidoro, A., & Viscusi, G. (2020). "Digital Transformation and Business Model Innovation in the Film Industry: The Case of Movieday.it." Technology and Creativity (pp.239-265). DOI:10.1007/978-3-030-17566-5\_10.
- [4].Horstmann, C. S., & Cornell, G. (2019). Core Java Volume I--Fundamentals. Oracle Press.
- [5].Date, C. J. (2019). An Introduction to Database Systems. Pearson.
- [6].Lowagie, B. (2010). iText in Action. Manning Publications. ISBN 9781935182610.

Отримана в редакції 12.06.2025. Прийнята до друку 18.06.2025. Received 12 June 2025. Approved 18 June 2025. Available in Internet 30 June 2025