Design and Implementation of a Bill Printing System Based on the PyQt Framework and Artificial Intelligence

Heming Zhang, Hongli Li

Affiliated Hospital of Binzhou Medical University, Binzhou 256603, Shandong

Abstract: ***Aiming at the current situation of low efficiency and easy errors in the traditional manual invoicing process of the hospital's financial department, this research designs and implements an automated bill printing system based on the PyQt framework, Python language, and combined with artificial intelligence technology. The system integrates SQLite database management, dynamic data query, and PDF template printing technology to achieve the full-process automation of bill entry, printing, voiding, and multi-dimensional data export. The core functions include: rapid information filling based on database linkage, automatic conversion of amounts to Chinese capitalization, user-defined template adaptation, etc. Applications show that the system reduces the single invoicing time from 3 minutes to within 10 seconds, significantly reduces the error rate, and optimizes the interaction design through user feedback. This research provides an efficient and practical technical solution for financial management in medical scenarios and has good prospects for popularization and application.***

Keywords: Bill Printing System; PyQt; Artificial Intelligence.

1. SYSTEM DEVELOPMENT BACKGROUND

The hospital's financial department needs to handle a large number of payment and invoicing operations for affiliated units, individuals, and internal staff. The traditional manual invoicing method is inefficient, taking an average of 3 minutes per invoice, and is prone to errors due to manual entry. Information queries are also cumbersome. Especially during peak payment periods such as personnel examinations and professional title promotions, multiple staff are often required to handle payments simultaneously, resulting in long waiting times for payers and low efficiency. To address these issues, this study developed an automated invoice printing system based on the PyQt framework and the Python language.

PyQt is the Python binding of the Qt application framework, which has the advantages of cross-platform, rich controls, and database support, enabling the rapid construction of feature-rich and aesthetically pleasing applications. Considering the moderate volume of invoice data, this system selects the lightweight database SQLite provided by Python to simplify deployment and maintenance. Junxi et al. (2024) proposed a graph convolutional network based on matrix factorization (GCN-MF) for recommendation systems [1]. Hu et al. (2024) explored multiscale deep neural networks for text sentiment analysis [2]. Zheng et al. (2025) introduced Diffmesh, a motion-aware diffusion framework for human mesh recovery from videos [3]. Pinyoanuntapong et al. (2023) studied self-aligned domain adaptation for mmWave gait recognition with Gaitsada [4]. Zhao et al. (2023) developed smart warehouse track identification using Res2Net-YOLACT+HSV [5]. Shan et al. (2025) rethought wine tasting for Chinese consumers via a service design approach enhanced by multimodal personalization [6]. Yuan et al. (2026) designed TA-Mem, a tool-augmented autonomous memory retrieval for large language models in long-term conversational question answering [7]. Zhou (2026) diagnosed bottlenecks in international automotive sales funnels using gradient boosting trees, with evidence from cross-regional team efficiency evaluation [8]. Shen et al. (2025) applied the whale optimization algorithm to financial payment fraud detection [9]. Wang et al. (2026) developed QA-ReID, a quality-aware query-adaptive convolution leveraging fused global and structural cues for clothes-changing person re-identification [10]. Li et al. (2026) proposed MFT, a memory-aware fine-tuning of SAM2 for efficient long-sequence video object segmentation [11]. Sun (2025) examined adaptive interfaces for personalized user experience using a machine learning approach [12]. Miao (2026) investigated big data technologies for enhanced network security analysis [13]. Wang (2025) enhanced object detection in photovoltaic farm panoramic imagery with a multi-scale feature-enhanced YOLOv8 [14]. Liu (2025) researched GPU parallel computing for image processing [15]. Lu et al. (2025) employed a Faster R-CNN model for flame detection [16]. Liu (2025) systematically evaluated deep learning architectures for plant image classification [17]. Gao et al. (2026) conducted an experimental study and geomechanical modelling of hard and

soft rock masses including fault zones [18]. Zhang (2026) built a hybrid LSTM-GARCH model integrating volatility factors from the US financial markets [19]. Finally, Han et al. (2026) developed a dual-scale model with collaborative reasoning and multi-feature fusion for robust detection of AI-generated images [20].

2. SYSTEM FUNCTIONS

The system includes three basic functions: bill management, data query, and system management.

2.1 Bill Management

1) Print bills: The core function simplifies the work of accounting personnel. The print date is automatically defaulted to the current system date and allows manual modification. The payer (Received today) supports directly reading the Excel sheet method and the payer database method. For centralized payment situations, the system supports batch importing of payer information from an Excel table, including mobile phone numbers, names, employee numbers, etc. The operator only needs to enter the first few digits of the mobile phone number or the employee number to quickly retrieve the payer's name. Select the "Auto Memory" checkbox behind the "Received today" edit box to automatically add the current item to the "Received today" database dictionary. Check the "Bring in the handler" checkbox

Table 1: Structure of the receipt detail table (receipt_detail)

Field Name	Data Type	Length	Decimal Places	Meaning
receipt_no	integer	10		Receipt Number, a number that uniquely identifies a receipt, automatically generated
todayget	varchar	20		Received Today, can be extracted from a dictionary or entered manually
handover	varchar	50		Handed In, extracted from a dictionary
charge_kind	varchar	10		Payment Method, including mobile payment, cash, etc.
charge_fee	decimal	10	2	Charge Amount
charge_date	datetime			Charge Date, automatically generated when payment is made
report_date	datetime			Settlement Date, automatically generated when settling accounts
handedby	varchar	10		Handler
op_code	varchar	10		Operator's Staff Number, automatically extracted from the logged-in user information
op_name	varchar	20		Cashier, retrieved from a dictionary based on the operator's staff number
invalid	varchar	10		Is Voided

Checkbox, which can automatically copy the "Received" information to the "Handler" column, reducing duplicate entry.

The functional layout of the ticket printing window is shown in Figure 2. The judgment process for entering tickets is shown in Figure 4. After clicking the "Print" button, the charging data is first saved to the database receipt detail table receipt_detail, and the table structure is shown in Table 1; then the relevant information is overprinted at the corresponding positions of the receipt, and the receipt format is shown in Figure 1.

2) Export charging details: Support exporting the details of all or the current cashier to an Excel file for convenient secondary processing.

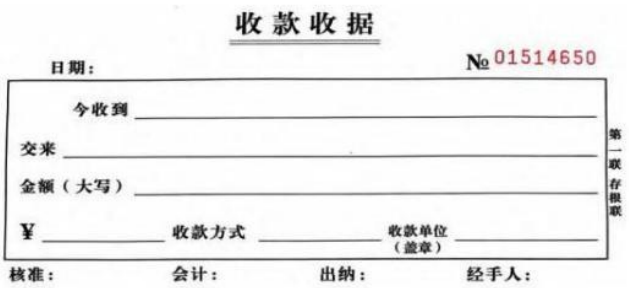

Figure 1: Sample of Receipt

Figure 2: Window for Printing Receipt

Figure 3: Window for Querying Details by Charging Time Period

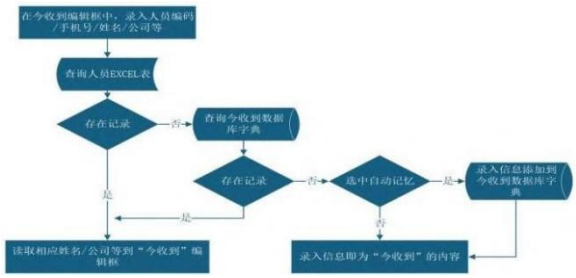

Figure 4: Flow Chart of Data Entry for "Received"

3) Cashier Settlement: Enter the current settlement date in the charging details of the current cashier for daily clearing.

4) Void Ticket: Used to handle tickets with input errors.

5) Re-print Ticket: Used to handle situations where the printer is abnormal or there are operation mistakes.

2.2 Data Query

1) Query of Charge Details: Support querying charge details by conditions such as cashier employee number, charge date, settlement date, settlement status, etc.

2) Query of Charge Summary: Support summarizing the number of charges and amounts by conditions such as charge/settlement date, cashier employee number, cashier, payment method, void status, etc.

3) Details Query by Charge Time Period: Support showing charge details by charge time period and exporting them to an Excel file.

2.3 System Management

1) User Maintenance: Manage system users.

2) Pay-in Name Maintenance: Maintain the "Pay-in" dictionary to improve user input speed.

3) Received Today Name Maintenance: Maintain the "Received Today" dictionary to improve user input speed.

4) Backup Charge Details: Backup worksheet data to the backup table.

5) Delete Charge Details: Delete worksheet data to slim down the worksheet and improve access speed.

6) Adjust Invoice Printing Position: Set the position of each printing item on the receipt to improve the adaptability to invoice and printer changes.

3. DEEPSEEK IMPLEMENTATION OF THE "CHECK DETAILS BY CHARGING TIME PERIOD" FUNCTION

This function allows users to customize the start date and end date. The system will retrieve and display all charging detail records within this time period and support exporting them as an Excel file. The layout window of the function to check details by charging time period is shown in Figure 3.

3.1 Command Prompt Words

Implemented with PyQt6: The user enters the start date and end date of the charge_date field in the receipt_detail table in the SQLite database to query the details that meet the conditions. The details are displayed in a QTableWidget. Set an "Excel Export" button. Click this button to export to an Excel file, and the default file name is "Charge Details" + "Start Date" + "-" + "End Date" + ".xlsx". The receipt_detail table contains the fields: receipt_no, todayget, handover, charge_kind, charge_fee, charge_date, report_date, handedby, op_name, invalid. The corresponding Chinese names are receipt number, received today, handed in, payment method, charge amount, charge date, settlement date, person handling, cashier, and whether void. It is hoped that these data will be displayed in Chinese names in the table. Add a delete button to delete the queried data in the receipt_detail table. There will be a prompt before deletion, and you can choose to cancel or confirm. The exported Excel file uses the corresponding Chinese field names as column names.

3.2 Main Generated Code

```
def initUI(self):
#Table
self.table = QTableWidget()
self.headers = ['Receipt Number', 'Received From', 'Paid By', 'Payment Method', 'Amount Charged', 'Charge Date', 'Settlement Date', 'Handler', 'Teller', 'Void?']
.....
def query_data(self):
start_date = self.start_date_input.text()
end_date = self.end_date_input.text()
try:
datetime.strptime(start_date, '%Y-%m-%d')
datetime.strptime(end_date, '%Y-%m-%d')
except ValueError:
QMessageBox.warning(self, 'Date Format Error', 'Please enter the correct date format (YYYY-MM-DD)')
return
.....
def export_to_excel(self):
start_date = self.start_date_input.text() end_date = self.end_date_input.text()
filename = f"Charge Details{start_date} - {end_date}.xlsx"
.....
df = pd.DataFrame(data, columns=self.headers)
df.to_excel(filename, index=False)
QMessageBox.information(self, 'Export Successful', f'Data has been successfully exported to {filename}')
```

4. EXPERIENCE

4.1 Advantages and Disadvantages of the Python Language

The Python language has a rich collection of third-party modules and fast development speed, but it has limitations in generating quick reports. Initially, Excel was used to print bills. Designing the report format was convenient and fast. It was even possible to use Excel's own functions to convert numbers to Chinese capital forms, and users also had a relatively high degree of freedom in design. The following are the Excel function commands for converting numbers to Chinese capital forms.

= " ⊗ "&IF(B7<0, " Negative ", "")&IF(TRUNC(B7)=B7, TEXT(IF(B7<0, -B7, B7), "[DBNum2]")&" Yuan Whole ", IF (TRUNC(B7*10) =B7*10, TEXT(TRUNC(IF(B7<0, -B7, B7)), "[DBNum2]")&" Yuan "& TEXT(RIGHT(B7), "[DB-Num2]")&"Jiao Whole", TEXT(TRUNC(B7),"[DBNum2]")&"Yuan "& IF(ISNUMBER(FIND(". 0", B7)), " Zero ", TEXT(LEFT (RIGHT(B7,2)),"[DBNum2]") & "Jiao") & TEXT(RIGHT(B7), "[DBNum2]")&"Fen"))

The lowercase amount in Chinese currency is saved in cell "B7". The "TRUNC" function truncates the decimal part of a number and returns an integer. The "FIND" function is used to search for a text string within another text string and returns the starting position. The "ISNUMBER" function is used to determine whether a value is a number. The "TEXT" function can convert a numeric value into text according to a specified format. The "LEFT" function is used to extract a specified number of characters from the left side of a text string. The "RIGHT" function is used to extract a specified number of characters from the right side of a text string. The "DBNum2" function can convert numbers to Chinese capital numbers. For example, "=DBNum2(123)" returns "One Hundred and Twenty-Three". However, when using the openpyxl module to write data to an Excel file, there is a delay of 5 to 10 seconds, resulting in a long waiting time for users and affecting efficiency.

By using the PyPDF2 module and changing the way to write bill information to PDF, the time delay problem was solved. However, new problems such as printer control and special symbol printing were introduced. The Ghostscript module was used to implement PDF printer control, and the canvas module was used to draw special symbols. Although additional processing functions were added, the system speed was still much faster than the Excel-based implementation.

4.2 Importance of User Requirements

During the system development process, continuous communication with users is crucial. Paying attention to every requirement put forward by users, even if it seems trivial, can provide clues and inspiration for developers to improve the program. The addition of checkboxes such as "Auto Memory", "Do not clear the amount", and "Bring in the handler" all stem from user requirements, and the addition of these functions has significantly improved the invoicing speed.

4.3 Application of Artificial Intelligence Code Generation Tools

With the breakthrough development of Large Language Models (LLMs) technology, intelligent code generation tools based on natural language interaction are gradually reconstructing the traditional software development paradigm. This case explores the application of artificial intelligence code generation tools in improving software development efficiency. Generative artificial intelligence tools represented by DeepSeek, Doubao of ByteDance, and ERNIE CodeLens of Baidu can significantly improve work efficiency in specific development scenarios through their advanced natural language processing (NLP) capabilities and code semantic understanding technologies.

With the help of NLP, the system can understand the intention behind the user's query, thus providing more accurate and contextually relevant search results. Search engines no longer rely solely on keyword matching but analyze the meaning of words and phrases, making it easier to find relevant information even for vague or complex queries, which helps improve the user experience in web search, document retrieval, and enterprise data systems.

With the help of code semantic understanding technology, it is possible to deeply analyze the logical structure and semantic information of the code, identify potential logical errors, syntax errors, and possible risks in the code, and help developers prevent problems in advance.

From a technical implementation perspective, these tools generally adopt multi-modal pre-trained models based on the Transformer architecture. Through deep learning of massive open-source code libraries (such as GitHub) and professional technical documents, a dual representation space of code syntax and semantic features is constructed. Taking the core module of the bill system developed in this project as an example, the traditional development of the "Chinese capital conversion of the amount" function takes about 3-4 hours for boundary condition testing and exception handling. With the help of the code generation function of DeepSeek and optimization through progressive prompt engineering, the development cycle can be compressed to within ten minutes. Specifically, the initial prompt "Generate a function for converting the amount to Chinese capital" can only output basic function code. By continuously adding constraint conditions (such as "Handling trillion-level numerical values", "Supporting positive and negative number identification", "Adding the word 'whole' when there is only one digit after the decimal point"), after 5 rounds of iterative optimization, the Python function finally obtained has a one-time pass rate of over 90% in unit tests.

In terms of implementing complex business logic, the development process of the "Query details by charging time period" function was investigated. The traditional development mode requires a linear process of business requirement analysis - flowchart design - code implementation. However, the human-machine collaboration paradigm based on the Trea tool shows significant advantages: First, describe the business rules in natural language, and the system automatically generates a UML state diagram; then, after interactive correction based on the visual chart, a SQL query statement with transaction processing capabilities is synchronously output. This "requirement - design - code" closed-loop iterative mechanism greatly improves development efficiency and reduces the risk of rework caused by misunderstandings of requirements.

5. SUMMARY

The automated bill printing system developed in this study effectively solves the problems of low efficiency and high error rate in the traditional manual invoicing process of hospital financial departments. This system has complete functions and convenient operations, significantly improving the invoicing efficiency and accuracy, and has high practical value and promotion potential.

FUND PROJECT:

Research on Hospital Informatization Based on Big Data (Shandong Health Economics Association); Research on the Construction of Evaluation Index System for Hospital Performance Management (SDWJYJ2023LM01019)

REFERENCES

[1] Junxi, Y., Wang, Z., & Chen, C. (2024). GCN-MF: A graph convolutional network based on matrix factorization for recommendation. Innovation & Technology Advances, 2(1), 14–26. https://doi.org/10.61187/ita.v2i1.30

[2] Hu, H., Zhang, J., & Sun, Y. (2024). The Multiscale Deep Neural Networks: Unveiling New Directions in Text Sentiment Analysis. Innovation & Technology Advances, 2(2), 34–45. https://doi.org/10.61187/ita.v2i2.65

[3] Zheng, Ce, et al. "Diffmesh: A motion-aware diffusion framework for human mesh recovery from videos." 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). IEEE, 2025.

[4] Pinyoanuntapong, Ekkasit, et al. "Gaitsada: Self-aligned domain adaptation for mmwave gait recognition." 2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS). IEEE, 2023.

[5] Zhao, X., Zhang, L., & Hu, Z. (2023). Smart warehouse track identification based on Res2Net-YOLACT+HSV. Innovation & Technology Advances, 1(1), 7–11. https://doi.org/10.61187/ita.v1i1.2

[6] Shan, X., Xu, Y., Xia, T., & Lin, Y. S. (2025, October). Rethinking Wine Tasting for Chinese Consumers: A Service Design Approach Enhanced by Multimodal Personalization. In 2025 International Conference on Content-Based Multimedia Indexing (CBMI) (pp. 1-5). IEEE.

[7] YUAN, Mengwei, et al. TA-Mem: Tool-Augmented Autonomous Memory Retrieval for LLM in Long-Term Conversational QA. In: 2026 9th International Conference on Advanced Algorithms and Control Engineering (ICAACE). IEEE, 2026. p. 2684-2688.

[8] Zhou, Z. (2026). Bottleneck Diagnosis in International Automotive Sales Funnels Using Gradient Boosting Trees: Evidence from Cross-Regional Team Efficiency Evaluation. Journal of Computer Technology and Applied Mathematics, 3(1), 11-18.
[9] Shen, Zepeng, et al. "Research on Application of Whale Optimization Algorithm in Financial Payment Fraud Detection." 2025 4th International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID). IEEE, 2025.
[10] Wang, Y., Jiang, K., Zhang, T., Tian, K., & Jiang, G. (2026). QA-ReID: Quality-Aware Query-Adaptive Convolution Leveraging Fused Global and Structural Cues for Clothes-Changing ReID. arXiv preprint arXiv:2601.19133.
[11] Li, G., Yuan, H., Chen, S., Hu, Q., Wang, J., & Jiang, K. (2026). MFT: Memory-Aware Fine-Tuning of SAM2 for Efficient Long-Sequence Video Object Segmentation. IEEE Signal Processing Letters.
[12] Sun, L. (2025, November). Adaptive Interfaces for Personalized User Experience: A Machine Learning Approach. In Proceedings of the 2025 International Conference on Artificial Intelligence and Sustainable Development (pp. 457-462).
[13] Miao, J. (2026). Big Data Technologies for Enhanced Network Security Analysis: Applications and Approaches. International Journal of Advance in Applied Science Research, 5(4), 16-20.
[14] Wang, J. (2025). Multi-Scale Feature-Enhanced YOLOv8 for Object Detection in Photovoltaic Farm Panoramic Imagery. International Journal of Advance in Applied Science Research, 4(10), 7-11.
[15] Liu, X. (2025). Research on the Application of GPU Parallel Computing in Image Processing. International Journal of Advance in Applied Science Research, 4(2), 1-7.
[16] Lu, J., Chen, J., Chen, Z., Zhang, L., & Fang, J. (2025). Flame Detection Based on Faster R-CNN Model. International Journal of Advance in Applied Science Research, 4(7), 41-45.
[17] Liu, Y. (2025). The Deep Learning Paradigm for Plant Image Classification: A Systematic Evaluation of Architectural Efficacy. International Journal of Advance in Applied Science Research, 4(8), 73-79.
[18] Gao, H., Zhao, W., Chen, W., Liu, J., Peng, W., & Zhou, S. (2026). Experimental study and geomechanical modelling of hard and soft rock masses including fault zones. Environmental Earth Sciences, 85(1), 7.
[19] Zhang, X. (2026, March). A Hybrid LSTM-GARCH Model Integrating Volatility Factors from the US Financial Markets. In Proceedings of the 2026 International Conference on AI Decision-Making and Management (pp. 183-189).
[20] Han, J., Zhong, P. & Sun, L. Dual-scale model collaborative reasoning with multi-feature fusion for robust AI-generated image detection. Multimedia Systems 32, 363 (2026). https://doi.org/10.1007/s00530-026-02425-4

Author Profile

Heming Zhang (1971-), male, senior engineer, master, research direction: hospital informatization, hospital operation management;

Hongli Li (1973-), male, senior accountant, bachelor, research direction: hospital financial management.