Automation and Machine Learning (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/autml.2025.060102
ISSN 2516-5003 Vol. 6 Num. 1

Campus Dining Hall Ordering Software

Xin Yang [1*], Shengtao Bian [2], Rui Liu [1], Xiaoxiao Guo [1]

[1]Business School, Wuxi Taihu University, Wuxi, Jiangsu, 214125, China
[2]Intelligent Manufacturing School, Wuxi Taihu University, Wuxi, Jiangsu, 214125, China
1637858356@qq.com
[*]Corresponding author

Keywords: Campus canteen; Ordering software; Needs analysis; System design; Application effects; Digital construction

Abstract: This study developed a canteen ordering software for schools aimed at solving issues such as long waiting times, frequent ordering errors, and mismatches between dish supply and demand. Through surveys and interviews, we understood user needs and developed features including user management, dish management, and order processing. The softwares front-end uses Vue.js, the back-end uses Spring Boot, and the database uses MySQL. Pilot applications have shown that the software has improved canteen operational efficiency, shortened customer queuing and order processing times, optimized inventory management, and user satisfaction exceeds 80%. Despite progress in digital construction, the interface design and device compatibility need improvement. In the future, we plan to enhance intelligent recommendations, improve dish management, optimize the user experience, and consider integration with campus activities and health management systems.

1. Foreword

Under the impact of the digital wave, all sectors of society have been significantly affected. Among them, the operational model and service quality of school canteens directly influence the life experiences of teachers and students. Traditional canteen operation models often lead to overcrowding and long queues during peak hours, which not only affect students learning efficiency but also reduce their satisfaction.

The traditional manual ordering method is inefficient and prone to errors, which not only leads to inaccurate meal distribution and waste of resources, but also makes it difficult for the canteen to accurately predict the demand for dishes, thus increasing the operating cost.

Campus canteens adopted online ordering software for mobile device use, enabling students to order and pay easily. This smart system digitizes dining, improves service efficiency, and is becoming a trend in campus catering.

The introduction of ordering software helps improve the operation efficiency of the canteen: by optimizing the ordering process and realizing the automation of ordering, errors are reduced. The detailed order data provided by the software helps the canteen accurately predict the demand for dishes and reasonably arrange the procurement of food ingredients and the work plan of chefs.

The use of ordering software optimizes the dining experience of students: Students can order

food in advance, saving time and improving dining efficiency. Students can also evaluate dishes and services, and the canteen can adjust its service in time according to these feedbacks, enhancing interaction with students and thus improving the dining experience.

The promotion of food ordering apps helps advance the digitalization of campuses: As part of the digital ecosystem, these apps facilitate the digital transformation of schools. The integration and data sharing between the app and other campus systems enable schools to better understand students dietary preferences and optimize dining policies. The application of these apps drives the digitization of campus services, enhancing overall service quality and creating a smart and efficient campus environment.

2. Demand analysis of campus canteen ordering software [3]

2.1. User requirement analysis [9]

This study collected feedback from students, staff, and management to understand campus canteen ordering software needs through surveys and interviews.

Survey: Design a questionnaire and distribute it to all students and staff across the school, covering different grades, majors, and departments. The content of the questionnaire includes the main issues users face when ordering meals, their expectations for software functions, specific requirements for dish information display, and needs regarding the convenience of payment methods. For cafeteria management, the questionnaire particularly focuses on key areas they hope to optimize through software in their cafeteria operations.

Among them, 532 valid questionnaires were collected, including 426 students (21%, 24%,32% and 22% respectively from freshman to senior), and 106 faculty members (68% from teaching departments and 32% from administrative departments).

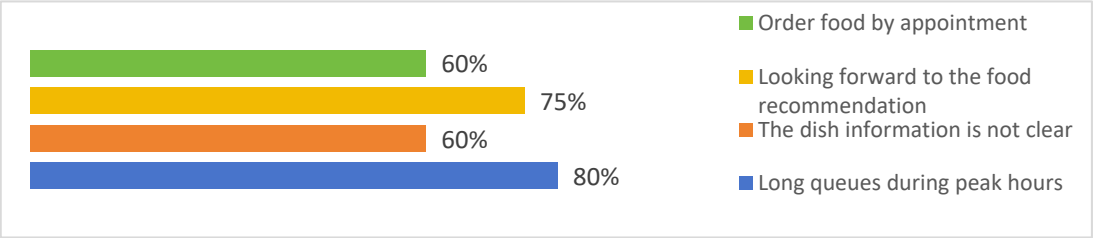

Figure 1: Survey Analysis

As shown in Figure 1, nearly 80% of students and staff reported long peak queue times, and 60% said the information about dishes was not clear. The canteen management reported that the shortage of manpower during peak orders led to slow meal delivery. 75% of users expected recommendations for dishes, and 60% wanted to make reservations.

And in the survey feedback, 85% of cafeteria managers hope that the software can achieve real-time inventory monitoring. 82% of users wish for dish displays to include information on ingredients and flavors, while 70% believe that high-definition images help with ordering. 85% of students and staff prefer WeChat Pay, 65% choose Alipay, and only 10% favor campus card payments. Cafeteria managers expect to integrate multiple payment methods.

Interview: Select representative students and staff members for one-on-one interviews to delve into their special needs during daily dining. For the cafeteria management, the core focus of the interview is on optimizing management processes, exploring how software technology can be used to achieve more efficient personnel allocation and cost control. The interview results are as follows in Table 1.

Table 1: Interview results

Interview results	
student	Some students want customized options, such as adjusting ingredients, and suggest adding social interaction features
teaching and administrative staff	Teachers and staff pay attention to healthy pairing, hoping that the software can provide healthy eating suggestions and quickly screen dishes
Canteen management	Emphasize the allocation of software optimization personnel, the allocation of window personnel according to the order volume, and the use of data analysis for accurate procurement to reduce food waste

All parties have clear requirements for the canteen ordering software. Users expect to solve issues such as queuing and lack of transparency in information, while managers hope to optimize operational processes. The software should feature intelligent recommendations, reservation ordering, detailed dish displays, and multiple payment methods, providing functions for inventory monitoring, personnel allocation, and data analysis for managers.

2.2. Functional requirements analysis [1]

As shown in Table 2 below.

Table 2: Key functional module

Ordering software needs to include key functional modules	
User management module	Implement student and faculty registration, use campus identity authentication, and ensure account security. Users can improve information, set preferences, view orders and reviews. The system supports user classification management and provides customized services.
Dish management module	The canteen management staff manage the dishes in the background, including adding, editing and deleting operations. Enter the detailed information of the dishes, update the inventory in real time, and flexibly set up the recommended special dishes.
Order management module	After the user places an order, the order information is transmitted in real-time to the kitchen. The system monitors the order status, allowing users to check the progress at any time. It supports various payment methods and provides electronic vouchers. The backend can query and analyze orders, reasonably arranging meal preparation and staffing.。
Raw material management module	Real-time access to raw material inventory details, including quantity, purchase time and shelf life. Inventory warning to replenish goods in time and avoid affecting food supply. Intelligent analysis of consumption trends to support procurement plans.
Complaint handling module	Users can easily submit complaints about food quality, service attitude, delivery delay and other issues. The management team responds quickly and handles them, and provides timely feedback on the results. Complaint data is recorded and analyzed, and service quality is continuously improved.

2.3. Performance requirements analysis

(1) The software should respond quickly, with feedback within 1 second after operation, and no more than 3 seconds in response during peak network times to ensure a smooth experience.

(2) The software needs to run stably, support high concurrency access, have load balancing, regular automatic data backup, resist hardware and software failures, and ensure the integrity and reliability of data.

(3) Pay attention to information security and payment security, use encryption technology to protect user data, follow payment security standards, implement strict permission management, and prevent security risks.

3. Design of campus canteen ordering software system [5]

The system architecture design organizational chart is shown below [4].As shown in Figure 2.

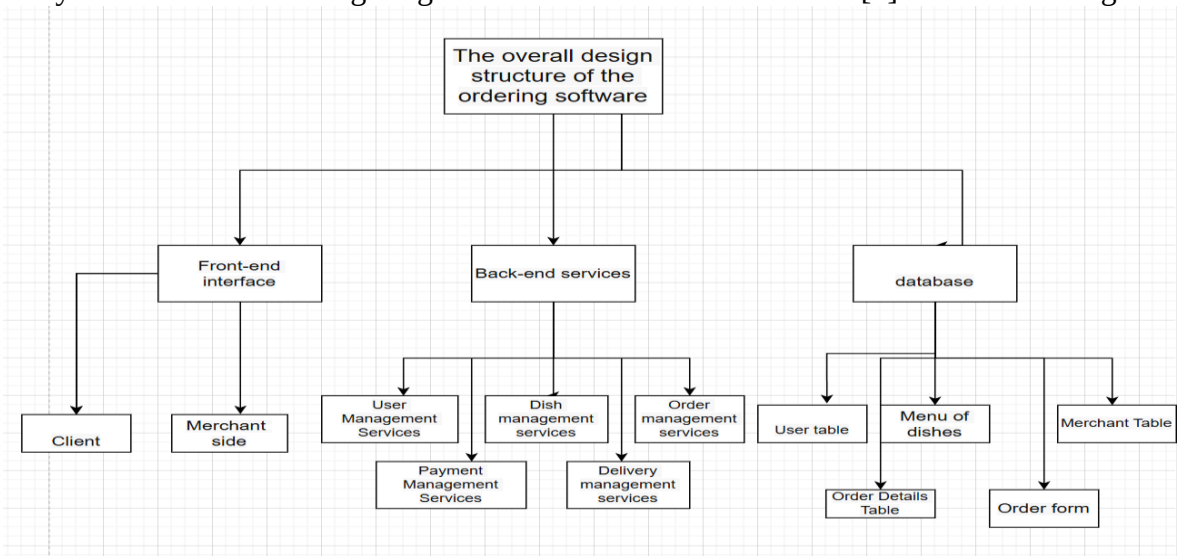

Figure 2: System Architecture Design Organizational Chart

3.1. Front-end interface design [6]

3.1.1. User side

As shown in Figure 3 below.

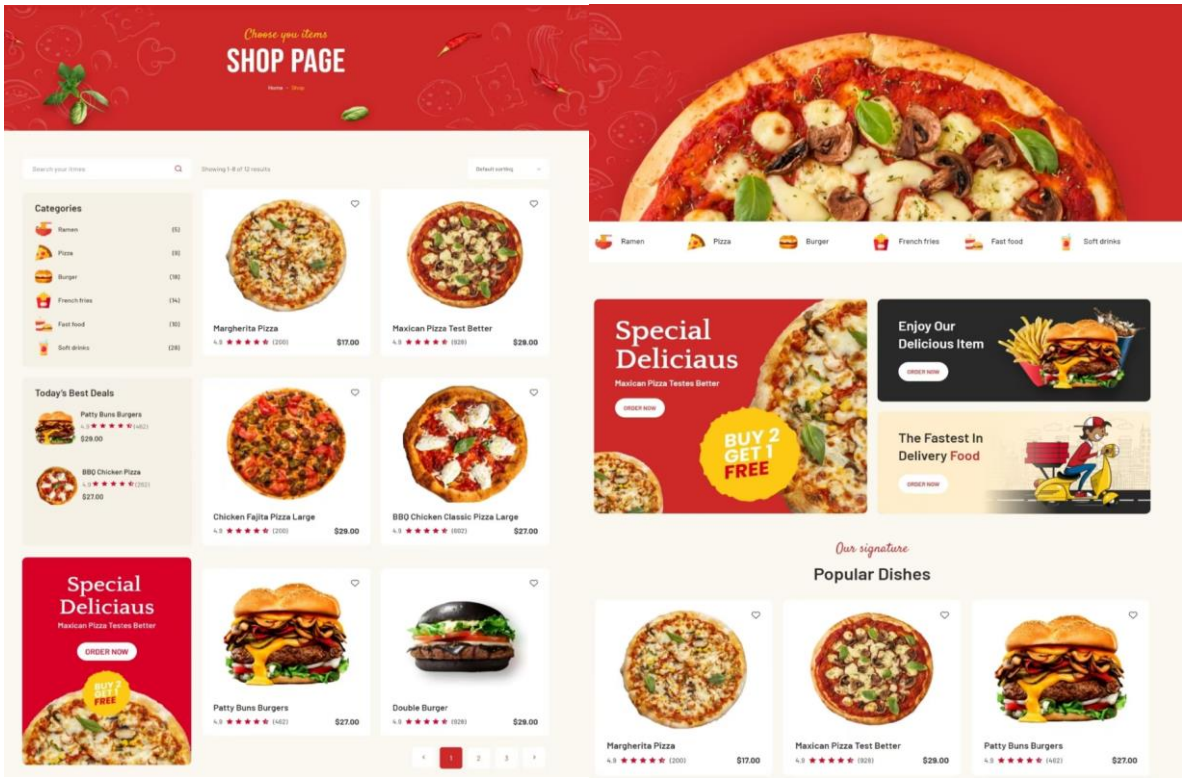

Figure 3: User interface illustration

1) Home page: presents popular dish recommendations and promotional activities, equipped with a search bar and dish category navigation, so that users can quickly locate the dishes they need.

2) Dish list page: Display dishes according to categories, each dish is accompanied by pictures, names and prices, and supports price and sales sorting screening function.

3) Dish detail page: display detailed pictures, descriptions, ingredients and other information of dishes. Users can choose the specifications and quantities of dishes, and can perform "add to shopping cart" or "order immediately".

4) Shopping cart page: summarize the dishes selected by users, display the dish information, total price, support quantity modification, dish deletion, provide settlement function.

5) Order confirmation page: used to check the delivery address, selected dishes and total price. After users select the payment method, they can complete the order submission.

6) Order tracking page: display the order status, including order placed, delivery in progress, delivered and so on.

7) Personal Center: Users can view personal information, order history, manage personal delivery address, and check available coupons.

3.1.2 Merchant side [2]

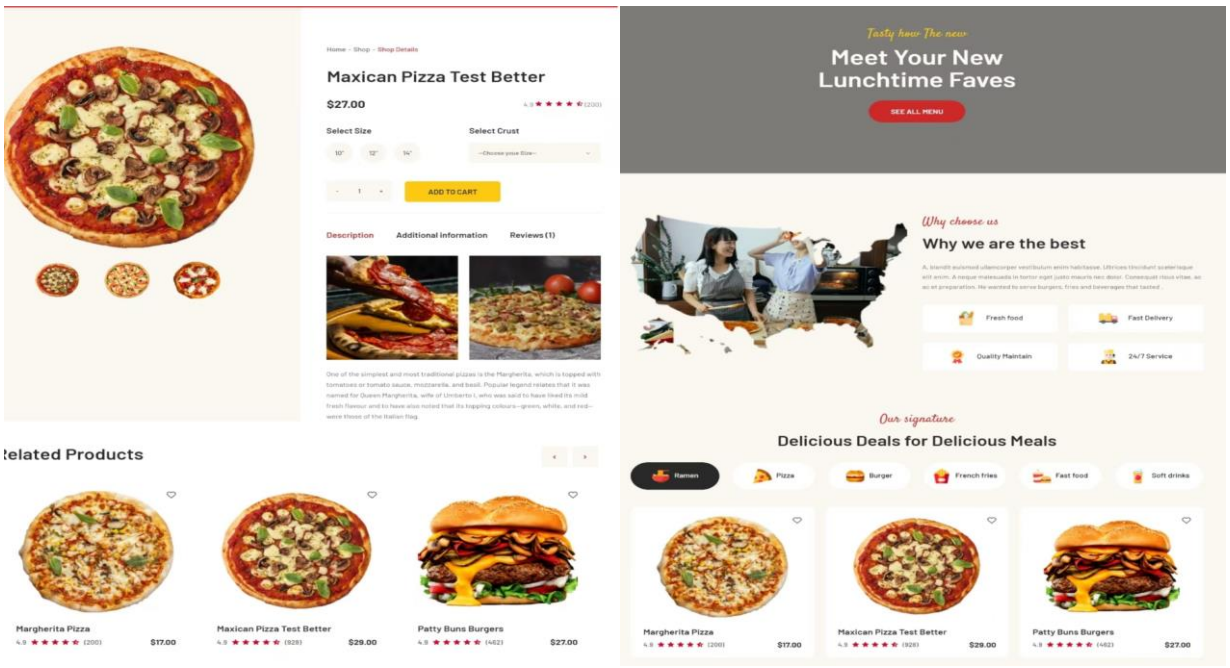

Figure 4: Merchant side illustration

As shown in the figure above, Figure 4.

(1) Registration function: Support registration through mobile phone number, and submit relevant qualification information for review when registering.

(2) Dish management: realize the operation of adding, editing and deleting dishes, setting dish inventory, price and other information.

(3) Order management module: allows merchants to view new orders, execute order acceptance or rejection operations, and track the delivery status of orders in real time.

(4) Data statistics: provides the function of viewing basic data such as todays order volume and turnover.

(5) Store information management: set up basic information such as store name, address and business time.

3.2. Back-end service architecture [10]

3.2.1. User management module

1) Registration and login: Support mobile phone registration and login, verify the information is correct, store user data, and generate login credentials.

2) Information management: allows users to modify personal information and encrypt the information for storage.

3) Points management: Users can get points when they place an order, which can be used to redeem coupons or deduct amounts.

3.2.2. Dish management module

1) Dish operation: receive the request of adding, modifying and deleting dishes from merchants, and update the database data.

2) Dish query: Provide a dish query interface for the front end, and return dish information according to the category, keyword and other conditions.

3.2.3. Order management module

1) Order processing: receive user orders, verify the information and store it in the database, notify merchants to receive orders, and update the order status in real time.

2) Order query function: allows users and merchants to easily query the specific details of orders with order number and other conditions.

3.2.4. Payment management module [11]

1) Payment integration: With the development of the e-commerce industry, the access of mainstream payment channels such as Alipay and WeChat payment has become the key to improve consumers payment experience.

2) Payment processing: process payment request, receive payment receipt, update order payment status.

3.2.5. Distribution management module

1) Delivery information: manage the delivery scope, delivery method and other information of merchants.

2) Delivery tracking function: Through seamless connection with the delivery platform, order delivery location information can be obtained and displayed in real time, so that users can check at any time.

3.3. Database architecture design

3.3.1. User Information Table [13]

As shown in Table 3 below.

Table 3: User information table

user ID	As the primary key, use the self-incrementing integer form
cell-phone number	Set as a unique index for user registration and login verification

password	Encryption algorithm is used for storage
nickname	Users can define fields
integration	Record the users accumulated points
address message	Store the users delivery address

3.3.2. Table of dish information

As shown in Table 4 below.

Table 4: Dish Information Table

variety of dishes ID	As the primary key, use a self-incrementing integer
Name of dish	Set it as a unique index under the same merchant
Dish classification	Set it as a normal index to improve query efficiency
price	Use the decimal data type
stock	Record the current inventory of dishes
seller ID	As an external key, it is associated with the merchant information table

3.3.3. Order Information Sheet [14]

As shown in Table 5 below.

Table 5: Order Information Table

order for goods ID	As the primary key, use a self-incrementing integer
user ID	As an external key, it is associated with the user information table
seller ID	As an external key, it is associated with the merchant information table
order status	It is represented by enumeration type
Order time	Record in timestamp format
order amount	Use decimal data type
mode of payment	Record the type of order payment
shipping address	Store order delivery address information
courier ID	As an external key, establish an association with the deliveryman information table (such as the deliveryman module included in the system)

3.3.4. Order details information table

As shown in Table 6 below.

Table 6: Order Details Table

Order details ID	As the primary key, use a self-incrementing integer
order for goods ID	As an external key, it is associated with the order information table
variety of dishes ID	As an external key, it is associated with the dish information table
Number of dishes	Record the quantity of each dish in the order

4. Implementation of campus canteen ordering software

4.1. Development environment and technology selection

Front-end development [12]

Development tools: Select Visual Studio Code and use Live Server plug-ins to enable real-time preview of code.

Technical framework: Use Vue.js framework to build user interface, combine Vue Router for page routing management, use Vuex for application state management, and use Element UI library to enrich interface components.

Programming language: JavaScript is the main programming language, combined with HTML5 and CSS3 technology.

Back-end development

Development tools: Use IntelliJ IDEA for back-end development.

Technical framework: Use Spring Boot framework to quickly build the project architecture, use Spring Data JPA to operate the database, and ensure system security through Spring Security.

Programming language: Java is used to write the backend logic.

Database: MySQL database is selected to store data.

Auxiliary tools: use Git for version control management, and use Postman tools for interface testing.

4.2. System implementation process [7]

User management module

Code implementation: Backend defines user entity class and implements service and control layer code for registration, login, and information modification. Frontend develops registration and login pages using Vue components and Axios library to interact with backend for data.

Interface Design: Registration interface receives and verifies user information, stores it in the database, and returns the result. Login interface verifies phone number and password; if successful, returns a JWT token. Information modification interface updates user info based on ID and returns the result.

Database operation: Through the Repository interface of Spring Data JPA, the save method is used to save user information during registration. During login, the query and verification are performed through the mobile phone number and password. When modifying the information, the query operation is performed first and then the update and save are performed.

Dish management module

Code implementation: The backend defines the entity class of dishes and writes the service layer and control layer code for adding, deleting, modifying and checking. The frontend implements the dishes management page.

Interface Design: Add an interface to receive dish information, store it in the database, and return the dish ID. Edit an interface to update dish information based on the dish ID. Delete an interface to remove dishes according to the ID. Query an interface to search for dishes by category or keyword and return the results.

Database operation: Use save method to save when adding dishes, query and update before editing dishes, use delete ById method to delete dishes, and implement the query of dishes through custom methods or JPQL.

Order management module

Code implementation: The backend defines the order entity class and writes the service layer and control layer code for order creation, status update, query, etc. The frontend implements the order submission and status tracking page.

Interface design: Create an interface to receive order information and store it in the database, and return the order ID. The status update interface updates the order status according to the order ID. The query interface queries the order according to the user or merchant ID and the order status, and returns it.

Database operation: Use save method to save when creating an order, query and locate the order before modifying and saving when updating the status, and implement query operation through custom methods combined with specific conditions.

Payment Management Module

Code Implementation: Integrate the payment SDK into the backend, write the control layer and service layer code for payment and callback. The frontend sets up a payment button on the order confirmation page and calls the backend interface.

Interface design: The payment interface receives order amount and other information, calls the SDK to generate a payment link or QR code, and returns it to the front end. The callback interface receives notifications from the payment channel, verifies them, and then updates the order payment status.

Database Operations: After payment is successful, query the order entity, modify and save the payment status.

4.3. System testing and optimization [15]

The system testing is shown in Table 7 below, and the system optimization is shown in Table 8 below.

Table 7: System testing

test	
function testing	In the process of executing backend interface logic testing, this study adopted JUnit and MockMvc frameworks to ensure the correctness of functions. To evaluate system performance, this study used JMeter tools to simulate multi-user concurrent access, identifying potential performance bottlenecks by testing response time and throughput.
safety test	The OWASP ZAP tool is used to detect SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF) and other vulnerabilities, and authentication and authorization mechanisms are tested.

Table 8: System optimization

optimize	
Performance optimization measures are covered	Introduce Spring Cache cache mechanism for high-frequency queries, optimize database indexes to improve query efficiency, and simplify code logic to reduce unnecessary computing burden.
Security optimization	Implement user input filtering and transfer strategy, strengthen authentication and authorization mechanism.
code optimization	Perform code reviews, follow programming standards and design patterns, and extract duplicate code segments.

5. Application effect of campus canteen ordering software

5.1. The efficiency of canteen operation has been improved

Campus cafeteria ordering software has greatly boosted operational efficiency via data analysis, cutting queue times to 5-10 minutes and order processing to 2-3 minutes. Inventory management has been optimized, preventing food waste and shortages, further enhancing efficiency.

5.2. User satisfaction improvement [8]

Survey and feedback show over 80% of users are satisfied with the dining app's overall experience, 85% with its ease of use. 75% find the dish information rich and accurate, and 82% are satisfied with the payment options. Users feel the app has improved daily convenience, and future improvements will be based on this feedback.

5.3. Contribution to campus digitalization

Campus meal ordering software has improved informatization, enabling real-time data collection, transmission, and analysis for canteen operations, enhancing campus information flow and sharing. It has advanced smart campus construction, offering personalized services to teachers and students, optimizing menus and service strategies, and boosting satisfaction. The software's use has built experience in digital development, fostered digital talent, and supported other campus digital initiatives.

This study focuses on the campus cafeteria ordering software, its design, implementation, and impact. The software offers diverse ordering and payment options, such as WeChat and Alipay, with real-time updates, order tracking, and a review feature for feedback. It ensures performance, stability, scalability, and cost-effectiveness using suitable frameworks and tools.

In front-end development, a user interface with good interactivity and quick response was built using Visual Studio Code and the Live Server plugin. Back-end development, based on IntelliJ IDEA, implemented business logic, user authentication and authorization, and database interactions, ensuring the stability and reliability of the back-end services. MySQL was chosen for database management, and by optimizing database indexes and query statements, the efficient and stable operation of the software was ensured.

The software has undergone extensive testing for functionality, performance, and compatibility. It's built on a Python+Django food delivery system, ensuring high-quality output. Initial applications demonstrate it enhances catering efficiency, optimizes resources, reduces food waste, increases user satisfaction, and promotes digital transformation and literacy.

Acknowledgements

This work was supported by Jiangsu Province College Student Innovation Training Program Project “Campus Food Gathering-Development of Campus Ordering Software”. The base number is 202413571067T. This work was also supported by Research Project on the Construction of Teachers' Morality and Style in the New Era in 2024 of the Wuxi Taihu University “Research on the Construction of Teacher Ethics and Conduct in Private Undergraduate Colleges in the New Era”. The base number is NDJY2403.

References

[1] Wang Xin, Li Ming. WeChat Mini Program Campus Canteen Order System [J]. Computer Application Research, 2020, 37(05):145-148.

[2] Liu Gang, Gao Feng. Design of SSM Framework Canteen Management System [J]. Software Guide, 2019, 18(08):95-98.

[3] Chen Yang and Zhou Na. Design of a Mini Program for Campus Catering Management System [J]. Information and Computer (Theoretical Edition), 2021, 33(11):74-76.

[4] Anonymous. Campus canteen ordering system: integration of convenience, personalization and intelligence [J]. Forum on Educational Informatization, 2022, 6(04):112-114.

[5] Anonymous. Digital Transformation Plan for University Canteens [J]. China Education Informatization, 2021, (13):88-91.

[6] Anonymous. Student information management system based on Java [J]. Computer Knowledge and Technology, 2018, 14(27):81-83.

[7] Wang Shuai, Chen Li. Design and implementation of campus canteen ordering system based on Spring Boot and Vue [J]. Modern Computer (Professional Edition), 2020, (22):79-82.

[8] Li Hua and Zhang Yue. Research on user experience optimization strategy of campus ordering system [J]. Software Engineering, 2021, 24(07):33-35.

[9] Zhao Gang and Sun Xiao. Exploration of the Application of Big Data in the Campus Canteen Ordering System [J]. Data Analysis and Knowledge Discovery, 2022, 6(06):135-142.

[10] Liu Lin and Wu Di. Practice and Thinking on Mobile Application Interface Design in Campus Order System [J]. Design Art Research, 2021, 11(03):115-118.
[11] He Ming, Zhou Wei. Research on the security and privacy protection mechanism of campus canteen ordering system [J]. Network Security Technology and Application, 2022, (05):101-103.
[12] Chen Jie and Wang Fang. Research on dish recommendation algorithm of campus ordering system based on artificial intelligence [J]. Journal of Intelligent Systems, 2021, 16(04):741-747.
[13] Zhang Feng, Li Xia. Analysis of integration strategy of campus canteen ordering system and campus one-card [J]. Modern Educational Technology, 2020, 30(09):112-117.
[14] Yang Jun, Liu Chang. Research on the Operation mode innovation of campus food ordering system in Chinese universities [J]. Science and Technology in Chinese Universities, 2021, (07):90-92.
[15] Guo Qiang and Zhao Lin. Selection and Practice of Cross-platform Development Technology for Campus Canteen Ordering System [J]. Computer Application and Software, 2022, 39(08):176-180.