**Title: Designing an application Point of Sale for Restaurant**

Nhan Mac

GitHub: [macnhan79/cpsc-491 (github.com)](https://github.com/macnhan79/cpsc-491)

Contents

[2 Abstract: 4](#_Toc153481277)

[3 Project Overview 4](#_Toc153481278)

[3.1 Project Goals 4](#_Toc153481279)

[3.2 Target Users 5](#_Toc153481280)

[3.3 Introduction: 5](#_Toc153481281)

[3.4 Project Plan: 6](#_Toc153481282)

[3.4.1 Project Kickoff and Requirements Gathering 6](#_Toc153481283)

[3.4.2 User Interface Design 6](#_Toc153481284)

[3.4.3 Integration with SquareUp API (Squareup.com) 6](#_Toc153481285)

[3.4.4 Testing and Debugging 7](#_Toc153481286)

[3.4.5 Deployment and Maintenance 7](#_Toc153481287)

[3.5 Metrics: 8](#_Toc153481288)

[3.6 Completed Stories 10](#_Toc153481289)

[3.6.1 User Authentication 10](#_Toc153481290)

[3.6.2 Product Catalog Management 11](#_Toc153481291)

[3.6.3 Order Processing 11](#_Toc153481292)

[3.6.4 Transaction Management 11](#_Toc153481293)

[3.6.5 Inventory Tracking 11](#_Toc153481294)

[3.6.6 Reporting 11](#_Toc153481295)

[3.6.7 Accessibility Features 11](#_Toc153481296)

[3.6.8 Security Enhancements 12](#_Toc153481297)

[3.6.9 Cross-Platform Compatibility 12](#_Toc153481298)

[3.7 Task completed 12](#_Toc153481299)

[3.7.1 User Authentication Module: 12](#_Toc153481300)

[3.7.2 Product Catalog Management: 12](#_Toc153481301)

[3.7.3 Order Processing: 12](#_Toc153481302)

[3.7.4 Transaction Management: 12](#_Toc153481303)

[3.7.5 Inventory Tracking: 12](#_Toc153481304)

[3.7.6 Reporting System: 12](#_Toc153481305)

[3.7.7 User Management Module: 13](#_Toc153481306)

[3.7.8 Accessibility Features: 13](#_Toc153481307)

[3.7.9 Security Enhancements: 13](#_Toc153481308)

[3.7.10 Cross-Platform Compatibility: 13](#_Toc153481309)

[3.8 Requirements Engineering: 13](#_Toc153481310)

[3.9 Architecture and Design: 15](#_Toc153481311)

[3.10 UX Design: 19](#_Toc153481312)

[3.10.1 Sign In: 19](#_Toc153481313)

[3.10.2 Mapping Table List: 19](#_Toc153481314)

[3.10.3 Taking Order: 19](#_Toc153481315)

[3.10.4 Order Management: 20](#_Toc153481316)

[3.10.5 Ticket Management: 20](#_Toc153481317)

[3.10.6 Transaction Management: 20](#_Toc153481318)

[3.10.7 Product Management: 21](#_Toc153481319)

[3.10.8 Payment Processing: 21](#_Toc153481320)

[3.10.9 Report: 21](#_Toc153481321)

[3.11 Use case Diagram: 22](#_Toc153481322)

[3.11.1 Use Case Diagram 1: User Authentication 22](#_Toc153481323)

[3.11.2 Use Case Diagram 2: Order Processing 23](#_Toc153481324)

[3.11.3 Use Case Diagram 3: Reporting System 23](#_Toc153481325)

[3.11.4 Use Case Diagram 4: Inventory Tracking 23](#_Toc153481326)

[3.12 State Diagram 24](#_Toc153481327)

[3.13 Prototyping: 25](#_Toc153481328)

[3.14 Conclusion: 25](#_Toc153481329)

[4 Installation 26](#_Toc153481330)

[4.1 Installing SQL Server Express: 26](#_Toc153481331)

[4.2 Installing Internet Information Services (IIS): 27](#_Toc153481332)

[4.3 Steps to Restore a Database: 28](#_Toc153481333)

[4.4 Steps to Deploy a Web Application to IIS – server side 29](#_Toc153481334)

[4.5 Steps for the Client (KItchen display screen User): 30](#_Toc153481335)

[5 User Manual 32](#_Toc153481336)

[5.1 Instructions for Using the Web Application for the First Time: 32](#_Toc153481337)

[5.2 Kitchen Display screen – client side 53](#_Toc153481338)

# Abstract:

This development document outlines the process and strategies for creating a Point of Sale (POS) application. The proposed POS application aims to streamline and enhance the efficiency of retail transactions by providing a user-friendly and robust system for managing sales, inventory, and customer interactions.

# Project Overview

## Project Goals

The primary goal of this Point of Sale application development project is to design and implement a reliable and feature-rich software solution that empowers businesses to manage their sales operations efficiently. Specific objectives include:

1. Efficient Sales Management: Create a system that allows businesses to process transactions quickly and accurately.
2. Inventory Management: Develop features for tracking and managing inventory levels in real-time.
3. User-Friendly Interface: Design an intuitive user interface that minimizes training requirements for users.
4. Reporting and Analytics: Provide tools for generating sales reports and analyzing business performance.
5. Security: Implement robust security measures to protect sensitive customer and transaction data.

## Target Users

The target users for this POS application include small and medium-sized businesses across various industries, such as retail stores, restaurants, and service providers.

## Introduction:

Incorporating accessibility features into the Point of Sale (POS) application is crucial to ensure that visually impaired users can efficiently interact with the system. This module aims to develop accessibility enhancements that enable blind users to navigate the POS application and perform transactions seamlessly. The primary focus will be on voice commands and touch inputs to provide an accessible and user-friendly experience.

## Project Plan:

The development of the POS application will follow a structured approach to ensure its successful and timely completion. The key activities and responsibilities throughout the project include:

### Project Kickoff and Requirements Gathering

**Objective**: Define the project scope, objectives, and requirements.

**Tasks**:

* Define project scope and objectives.
* Identify stakeholders.
* Gather requirements from stakeholders and end-users.
* Finalize the project plan and timeline.

### User Interface Design

**Objective**: Create a place where a customer executes the payment for goods or services and where sales taxes may become payable.

**Tasks**:

* Design a user interface that accommodates transaction may occur in person or online.
* Incorporate high-contrast themes for better visibility.
* Develop wireframes and prototypes for user feedback.

### Integration with SquareUp API (Squareup.com)

**Objective**: Enable users to authenticate, access their **SquareUp** API, management Order and Transaction.

**SquareUp**:

* Integrate the SquareUp API.
* Implement Order/Transaction management.
* Create functionalities for viewing the Order/Transaction, updates Order/ Transaction.

### Testing and Debugging

**Objective**: Thoroughly test the application for functionality, usability, accessibility, and performance.

**Tasks**:

* Conduct extensive testing with visually impaired users.
* Identify and address bugs and issues promptly.
* Ensure compliance with project requirements and specifications.

### Deployment and Maintenance

**Objective**: Deploy the application to Web Server, monitor performance, and provide ongoing support.

**Tasks**:

* Deploy the application to the Web Server and Hosting.
* Monitor user feedback and application performance.
* Provide continuous maintenance, support, and feature updates as needed.

Timeline:

**Phase 1**: Planning and Requirements Gathering

* Week 1: Define project scope and objectives, identify stakeholders, and create a project plan.
* Week 2-3: Conduct market research, user analysis, and requirements gathering. Develop functional and non-functional requirements, use cases, and user stories. Create a project requirements document.

**Phase 2**: Design and Prototyping

* Week 4-5: Design the user interface and user experience for the app. Develop wireframes, storyboards, and prototypes.
* Week 6-7: Review and refine the app design based on user feedback. Develop a high-fidelity prototype.

**Phase 3**: Development and Testing

* Week 8-12: Develop the app features based on the project requirements and design. Integrate the SquareUp API into the app.
* Week 13-14: Test the app for functionality, usability, accessibility, and performance. Fix bugs and issues as they are discovered.

**Phase 4**: Deployment and Maintenance

* Week 15-16: Deploy the app to Web Server.
* Week 17-20: Provide user support and maintenance. Gather user feedback, and improve the app based on the feedback. Add new features as needed.

## Metrics:

To measure the success and effectiveness of the Point of Sale (POS) application development project, several key indicators and metrics will be used. These indicators will help assess the usability, accessibility, user satisfaction, and overall impact of the application. Additionally, the efficiency of the development process will be evaluated based on implementation strategies and coding standards. The following metrics will be employed:

1. Usability: Assess how easy it is for users to interact with the POS application efficiently.

Measurement:

* 1. User Testing: Conduct usability testing with actual users to evaluate the application's ease of use and effectiveness in facilitating transactions.
  2. User Feedback: Collect feedback from users, including store employees and customers, to identify areas of improvement.

1. Accessibility: Evaluate the extent to which the POS application meets the accessibility needs of all users.

Measurement:

* 1. Compatibility: Ensure compatibility with assistive technologies such as screen readers, keyboard navigation, and alternative input methods.
  2. Feedback Mechanisms: Implement audio and visual cues to provide feedback to users with different abilities.
  3. Testing with Diverse Users: Conduct testing with a diverse user group to validate accessibility features.

1. User Satisfaction: Measure user happiness and satisfaction with the POS application.

Measurement:

1. Surveys and Interviews: Administer surveys and conduct in-person interviews with store employees and customers to gather their opinions and feedback.
2. Feedback Analysis: Analyze user feedback to identify areas of improvement and prioritize enhancements.
3. Impact on Retail Experience: Assess how the POS application enhances the overall retail experience for businesses and customers.

Measurement:

1. User Feedback: Collect feedback from businesses and customers regarding their experience with the POS application.
2. Efficiency Metrics: Evaluate the efficiency of retail operations, such as transaction processing speed and inventory management, to gauge the application's impact.
3. Implementation efficiency: Evaluate the effectiveness of the development process, taking into account implementation techniques and coding standards.

Measurement:

1. Code Reviews: Conduct code reviews to assess adherence to coding standards, code quality, and maintainability.
2. Testing: Evaluate the application's quality, effectiveness, and scalability through comprehensive testing, including unit testing, integration testing, and user acceptance testing.

## Completed Stories

### User Authentication

As a user, I want to be able to register an account.

As a registered user, I want to log in securely to access my account.

As a logged-in user, I want the system to remember my session for a convenient experience.

### Product Catalog Management

As a business owner, I want to add new products to the catalog.

As a business owner, I want to edit/update product details.

As a business owner, I want to deactivate or remove products from the catalog.

### Order Processing

As a cashier, I want to create a new order for a customer.

As a cashier, I want to add products to the order with quantity.

As a cashier, I want to remove items from the order.

As a cashier, I want to view a summary of the order before finalizing.

### Transaction Management

As a cashier, I want to process payments securely.

As a cashier, I want to apply discounts or promotions to transactions.

As a cashier, I want to generate and print receipts for customers.

### Inventory Tracking

As a business owner, I want to receive notifications when inventory is low.

As a business owner, I want to track the quantity of each product in stock.

As a business owner, I want to view historical inventory data.

### Reporting

As a business owner, I want to generate daily, weekly, and monthly sales reports.

As a business owner, I want to analyze sales trends over time.

As a business owner, I want to export reports in different formats (e.g., CSV, PDF).

User Management

As an admin, I want to manage user roles and permissions.

As an admin, I want to deactivate or remove user accounts.

As an admin, I want to view a log of user activities.

### Accessibility Features

As a user, I want the application to have high-contrast themes for better visibility.

As a user, I want the application to be compatible with screen readers.

As a user, I want intuitive navigation with keyboard shortcuts.

### Security Enhancements

As a developer, I want to implement secure password storage.

As a developer, I want to secure communication between the client and server.

As a developer, I want to conduct regular security audits.

### Cross-Platform Compatibility

As a user, I want to access the application from different web browsers.

As a user, I want to run the desktop application on various Windows versions.

## Task completed

### User Authentication Module:

* Implemented user registration functionality.
* Developed a secure login system.
* Integrated password encryption for user security.
* Implemented session management for user sessions.

### Product Catalog Management:

* Created features for adding new products to the catalog.
* Implemented product editing and updating capabilities.
* Enabled product deactivation or removal from the catalog.
* Integrated image uploading for product thumbnails.

### Order Processing:

* Implemented the creation of new orders with a user-friendly interface.
* Developed functionalities for adding, removing, and modifying items in an order.
* Provided a comprehensive order summary for confirmation.

### Transaction Management:

* Integrated secure payment processing gateways.
* Implemented discount and promotion application during transactions.
* Enabled the generation and printing of detailed receipts.

### Inventory Tracking:

* Implemented low inventory notifications for proactive management.
* Enabled real-time tracking of product quantities in stock.
* Provided historical data on inventory changes.

### Reporting System:

* Developed features for generating daily, weekly, and monthly sales reports.
* Enabled trend analysis for informed business decisions.
* Provided export options for reports in CSV and PDF formats.

### User Management Module:

* Implemented admin features for role and permission management.
* Enabled deactivation and removal of user accounts.
* Developed a comprehensive user activity log.

### Accessibility Features:

* Designed high-contrast themes for improved visibility.
* Ensured compatibility with screen readers and keyboard navigation.
* Implemented intuitive navigation using keyboard shortcuts.

### Security Enhancements:

* Implemented secure password storage techniques.
* Ensured secure communication between the client and server.
* Conducted regular security audits to identify and address vulnerabilities.

### Cross-Platform Compatibility:

* Developed the web application to be compatible with major browsers.
* Designed and tested the desktop application for various Windows versions.
* Ensured consistent functionality across both web and desktop platforms.

## Requirements Engineering:

In the process of developing the Point of Sale (POS) application, requirements engineering plays a pivotal role in defining the project's scope and ensuring that the application meets the needs and expectations of its users. In this context, we will outline the essential aspects of requirements engineering for our POS application:

**User-Centered Requirements:** To create a successful POS application, it is essential to gather, evaluate, and document user needs and expectations. These requirements will be developed with a focus on accommodating a wide range of users, including store employees and customers. Specifically, we will consider the needs of users who rely on screen readers and keyboard navigation for accessibility.

**Modeling and Specification Tools:** To ensure that the requirements are well-defined and implementable, we will employ modeling tools and specification processes. These tools will assist in creating a clear and comprehensive set of requirements that guide the development of the POS application.

**Transaction Processing and Order Management:** The core functionality of the POS application revolves around transaction processing and order management. The application will utilize the SquareUP.com API to facilitate these processes. Here are the key technical considerations related to transaction processing and order management:

**Implementation Strategies:** SquareUP.com API Integration will serve as the foundation for processing transactions and managing orders within the POS application. Here are the key steps and strategies for integrating this API:

1. **API Registration**: Register an account with SquareUP.com and obtain the necessary API key or access token to enable communication between the application and the SquareUP.com platform.
2. **API SDK Setup**: Utilize the software development kit (SDK) provided by SquareUP.com to seamlessly integrate the API into the application. Ensure that the SDK is configured to work harmoniously with the chosen programming language and platform.
3. **Transaction Processing**: Implement functionality within the application to record and transmit transaction data to SquareUP.com for processing. Ensure compatibility with various transaction types and payment methods.
4. **Order Management**: Develop features for managing customer orders, including order creation, modification, and tracking. Utilize SquareUP.com's order management capabilities to enhance the user experience.
5. **Testing and Feedback**: Rigorously test the integration of the SquareUP.com API to verify its functionality, security, and reliability. Gather user feedback to identify any issues or areas for improvement.
6. **User Feedback**: Collect feedback from store employees and customers to assess the effectiveness and user-friendliness of the transaction and order management processes enabled by the SquareUP.com API.

## Architecture and Design:

The architecture and design of the Point of Sale (POS) application play a crucial role in ensuring the efficiency, scalability, and usability of the system. This document outlines the architectural considerations and design principles for the POS application, which will be developed as a web application using ASP.NET and a Windows Forms (WinForms) application for desktop use.

The POS application is intended to provide a comprehensive solution for businesses to manage their sales, inventory, and customer transactions efficiently. It will consist of two main components:

* Web Application (ASP.NET): This component will serve as the primary interface for businesses to manage their operations, including product catalog management, order processing, and reporting. It will be accessible via web browsers.
* WinForms Application: This desktop application will cater to businesses that prefer a standalone, offline solution for point of sale operations. It will offer the same functionality as the web application but in a native Windows desktop environment.

**Architecture**

The POS application will follow a multi-tier architecture to ensure separation of concerns and scalability. The architecture consists of the following tiers:

1. Presentation Tier

Web Application (ASP.NET): The web application will serve as the user interface for businesses to interact with the POS system. It will be developed using ASP.NET, HTML, CSS, and JavaScript to provide a responsive and user-friendly interface accessible via web browsers.

WinForms User Interface: The WinForms application will provide a desktop-based user interface for businesses. It will be developed using WinForms, offering a native Windows experience.

1. Application Logic Tier

Business Logic Layer: This layer will contain the core business logic of the POS application. It will handle functions such as order processing, inventory management, and payment processing. The business logic will be shared between the web and WinForms applications to ensure consistency.

1. Data Access Tier

Data Access Layer: The data access layer will be responsible for interacting with the application's database. It will use Entity Framework for database access in both the web and WinForms applications. Database management systems like SQL Server will be used to store and manage data.

1. Database Tier

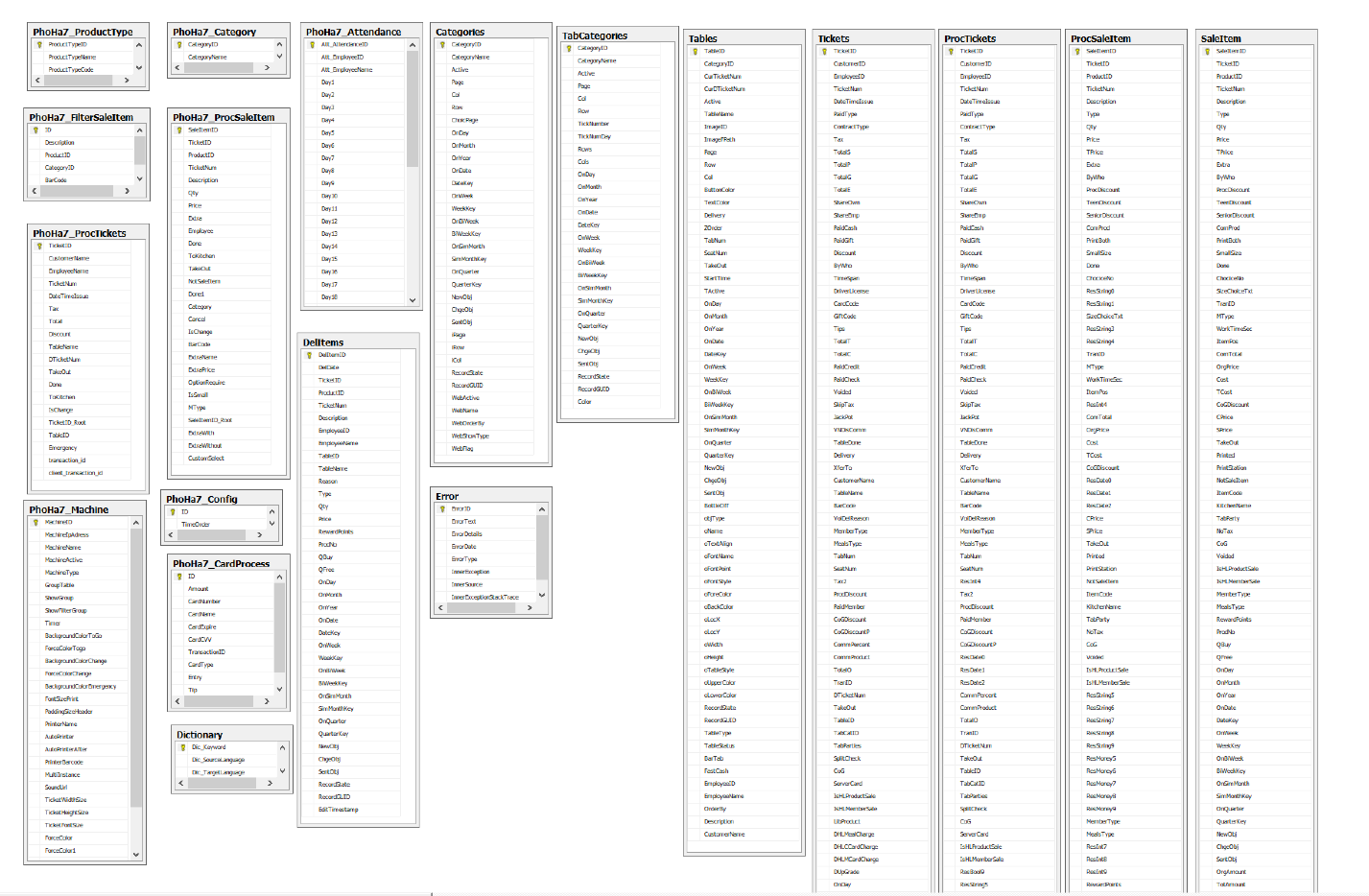
Database Server: The database tier will consist of a centralized database server where all application data will be stored. It will be accessed by both the web and WinForms applications.

1. Payment Processing:

Currently, our Point of Sale (POS) web application exclusively supports iOS devices for processing credit and debit card payments. This decision was made to ensure a seamless and secure payment experience for our users on the iOS platform. By focusing on iOS, I can optimize our application to take full advantage of the platform's capabilities and provide a consistent and reliable payment processing solution. However, I am actively exploring opportunities to expand our platform compatibility in the future to accommodate a wider range of devices and operating systems, enhancing accessibility and convenience for all users. Our commitment is to continually improve our POS application to meet the evolving needs of our customers and provide a versatile payment solution.

* ER diagram

A screenshot of a computer

Description automatically generated

## UX Design:

The User Experience (UX) design of the Point of Sale (POS) application for assistive use by blind Facebook users is crucial for ensuring a user-friendly and accessible interface. This document outlines the design details for each step of the application's user interface to enhance usability, accessibility, and overall user satisfaction.

### Sign In:

Objective: Enable users to securely sign in to their accounts.

Design Details:

Simplified Authentication: Implement a straightforward sign-in process with minimal steps.

Voice Guidance: Provide voice-guided instructions for entering login credentials.

High-Contrast Themes: Offer high-contrast color schemes for better visibility.

Screen Reader Compatibility: Ensure compatibility with screen readers for reading login fields and errors.

### Mapping Table List:

Objective: Facilitate navigation and item selection within the application.

Design Details:

Intuitive Navigation: Organize table listings in a logical and easy-to-navigate manner.

Text-to-Speech: Utilize text-to-speech technology to announce table names and availability.

Voice Commands: Enable voice commands to select tables and access additional details.

Haptic Feedback: Implement haptic feedback to confirm table selection.

### Taking Order:

Objective: Allow users to efficiently take customer orders.

Design Details:

Voice-Enabled Ordering: Enable voice commands to add items to orders.

Menu Accessibility: Provide an easily accessible menu with clear item descriptions.

Order Summary: Continuously update and read aloud the order summary.

Error Handling: Offer clear audio feedback for order errors and options to correct them.

### Order Management:

Objective: Assist users in managing and tracking customer orders.

Design Details:

Order Status Updates: Announce order status changes through voice prompts.

Order Modification: Enable users to modify orders through voice commands.

Order History: Offer an accessible history of past orders for reference.

### Ticket Management:

Objective: Manage and organize customer orders efficiently.

Design Details:

Ticket Organization: Display tickets clearly with relevant details.

Ticket Navigation: Allow users to swipe or use voice commands to switch between tickets.

Voice-Activated Actions: Enable voice commands to mark tickets as completed or in-progress.

### Transaction Management:

Objective: Handle transactions securely and accurately.

Design Details:

Secure Payment: Implement a secure payment interface.

Voice-Guided Payment: Guide users through the payment process using voice prompts.

Payment Confirmation: Provide audio confirmation of successful transactions.

Error Handling: Clearly communicate any payment errors.

### Product Management:

Objective: Maintain an up-to-date product catalog.

Design Details:

Accessible Product Listings: Present products with detailed descriptions and prices.

Voice Search: Allow users to search for products using voice commands.

Product Addition: Enable voice commands to add new products to the catalog.

### Payment Processing:

Objective: Efficiently process payments for customer orders.

Design Details:

Payment Gateway Integration: Integrate a reliable payment gateway for secure transactions.

Voice-Activated Payment: Enable voice commands for initiating payments.

Receipt Generation: Automatically generate and provide accessible receipts.

### Report:

Objective: Generate and access reports for business insights.

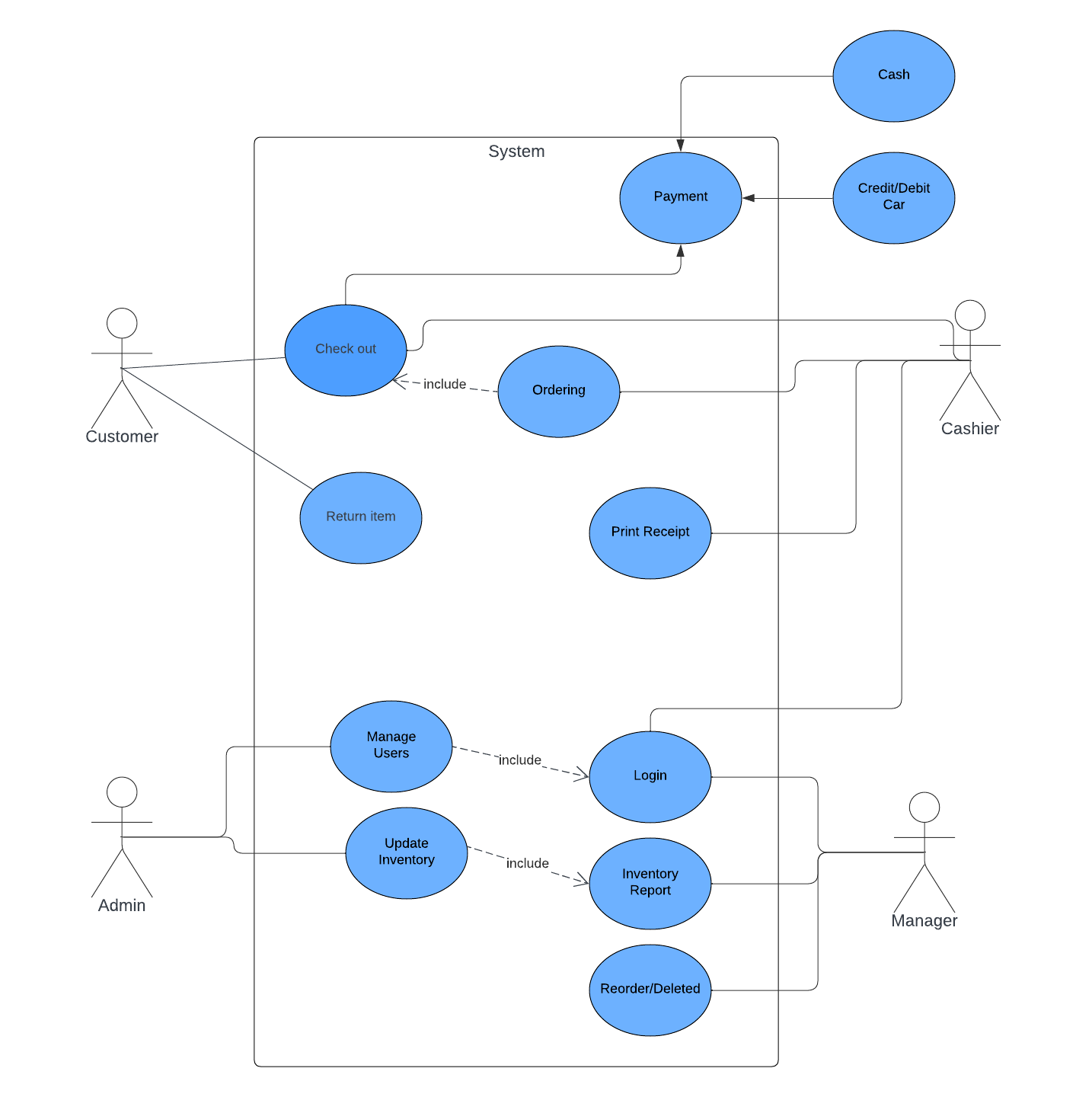
Design Details:

Accessible Reports: Create reports with clear and concise information.

Voice-Activated Report Access: Allow users to request specific reports using voice commands.

Data Export: Provide options for exporting reports in accessible formats.

## Use case Diagram:



### Use Case Diagram 1: User Authentication

* Actors:
* Customer: A user who interacts with the system to make purchases.
* Cashier: An employee who uses the system to process orders.
* Use Cases:
* Register Account: Allows customers to create an account.
* Login: Enables customers and cashiers to log into the system securely.
* Logout: Logs out the user from the system.

### Use Case Diagram 2: Order Processing

* Actors:
  + Customer: Places orders and makes purchases.
  + Cashier: Processes orders, adds/removes items, and finalizes transactions.
* Use Cases:
  + Create Order: Allows customers to initiate a new order.
  + Add Item to Order: Enables the cashier to add products to the order.
  + Remove Item from Order: Allows the cashier to remove items from the order.
  + Finalize Transaction: Completes the order and processes payment.

### Use Case Diagram 3: Reporting System

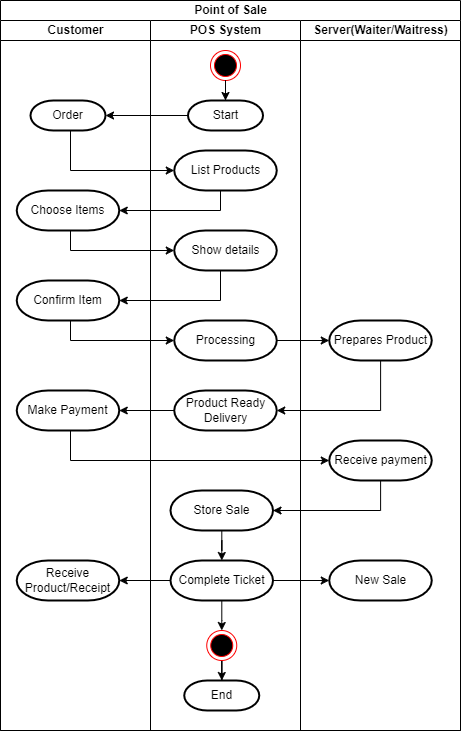
* Actors:
  + Business Owner: Analyzes reports and makes business decisions.
  + System: Automatically generates and manages reports.
* Use Cases:
  + Generate Daily Sales Report: Creates a report summarizing daily sales.
  + Generate Weekly Sales Report: Creates a report summarizing weekly sales trends.
  + Generate Monthly Sales Report: Creates a report summarizing monthly sales.
  + Export Report: Allows the business owner to export reports in CSV or PDF format.

### Use Case Diagram 4: Inventory Tracking

* Actors:
  + Business Owner: Manages and monitors inventory.
  + System: Automatically tracks inventory levels.
* Use Cases:
  + Receive Low Inventory Notification: Alerts the business owner when inventory is low.
  + Track Product Quantity: Allows the business owner to monitor the quantity of each product.
  + View Historical Inventory Data: Provides historical data on inventory changes.

## State Diagram

This state diagram visually represents the possible states and transitions within a POS system when a customer places an order and completes the payment.



## Prototyping:

In the process of developing an application, prototyping is an essential step since it enables testing and feedback that can be used to improve the program's user interface as well as its functionality. In order to guarantee that the application satisfies the requirements of blind users and that it is accessible and useable, multiple rounds of the prototype process will be carried out. Wireframing, mockups, and interactive prototypes are some of the tools and methods that will be utilized in the process of developing low-fidelity and high-fidelity prototypes, respectively. User testing will be undertaken with blind individuals in order to gain comments and insights that can be used to further improve the application's usability as well as its accessibility.

## Conclusion:

While the original goal was to enhance accessibility and inclusivity for blind users on social media, the current focus of the project is to develop a Point of Sale (POS) application that streamlines business operations and offers a user-friendly experience for both businesses and customers. This application aims to contribute to the efficiency and success of businesses while maintaining the highest quality and integrity standards.

# Installation

## Installing SQL Server Express:

Step 1: Download SQL Server Express:

1. Visit the [Microsoft SQL Server Express download page](https://www.microsoft.com/en-us/sql-server/sql-server-downloads).
2. Choose the edition of SQL Server Express that suits your needs. There are typically options for the latest version available. Click on the "Download now" button.

Step 2: Run the Installer:

1. Locate the downloaded installation file (usually a .exe file) and double-click on it to run the installer.
2. The SQL Server Installation Center will open.

Step 3: Choose Installation Type:

1. In the Installation Center, click on "New SQL Server stand-alone installation or add feature to an existing installation."
2. The setup will perform some checks on your system to ensure it meets the minimum requirements.

Step 4: Accept License Terms:

1. Read and accept the license terms by selecting the checkbox.
2. Click "Next" to proceed.

Step 5: Choose Installation Features:

1. Select the features you want to install. At a minimum, you'll need to install "Database Engine Services."
2. You can choose other features depending on your requirements.
3. Click "Next" to continue.

Step 6: Instance Configuration:

1. Choose the instance configuration. For most installations, you can use the default instance.
2. Click "Next" to proceed.

Step 7: Server Configuration:

1. Configure the SQL Server services. You can choose the service accounts and startup types. The default settings are usually sufficient for most installations.
2. Click "Next" to continue.

Step 8: Database Engine Configuration:

1. Choose authentication mode:
   * **Mixed Mode**: Allows both Windows Authentication and SQL Server Authentication. For this, set a strong password for the 'sa' (system administrator) account.
2. Add SQL Server administrators. Add yourself and any other users or groups who should have administrative access.
3. Click "Next" to proceed.

Step 9: Error Reporting and Usage:

1. Choose whether to send error reports to Microsoft.
2. Click "Next" to proceed.

Step 10: Ready to Install:

1. Review the summary of your selections.
2. Click "Install" to start the installation process.

Step 11: Installation Progress:

1. The installer will display the progress of the installation.
2. Once completed, you'll see a summary. Click "Next" to finish.

Step 12: Complete the Installation:

1. The last screen will indicate whether the installation was successful.
2. Click "Close" to exit the installer.

## Installing Internet Information Services (IIS):

Step 1: Open Control Panel:

1. Press **Win + X** and select "Control Panel" from the menu.

Step 2: Access "Programs and Features":

1. In the Control Panel, click on "Programs" and then click on "Turn Windows features on or off."

Step 3: Enable Internet Information Services (IIS):

1. In the "Windows Features" window, scroll down and find "Internet Information Services" (IIS).
2. Expand the node and select the components you want to install. At a minimum, select "Internet Information Services."
3. You may need to expand additional nodes to choose specific features like "World Wide Web Services" or "FTP Server" based on your requirements.
4. Click "OK" to start the installation.

Step 4: Installation Progress:

1. Windows will now install the selected features. This may take some time.

Step 5: Verify Installation:

1. Once the installation is complete, open a web browser and navigate to **http://localhost** or **http://127.0.0.1**. If IIS is installed correctly, you should see the default IIS welcome page.

## Steps to Restore a Database:

Step 1: Open SQL Server Management Studio (SSMS):

1. Launch SQL Server Management Studio.

Step 2: Connect to SQL Server:

1. Connect to your SQL Server Express instance. You can do this by providing the server name and authentication details.

Step 3: Locate "Databases" Node:

1. In the Object Explorer, locate the "Databases" node.

Step 4: Start the Restore Process:

1. Right-click on the "Databases" node and choose "Restore Database."

Step 5: Choose Source and Destination:

1. In the "General" page:
   * Select "Device" under the "Source" section.
   * Click the ellipsis (...) button to browse for your backup file.
   * Select the backup file (.bak) and click "OK."

Step 6: Select Backup Sets to Restore:

1. In the "Select a page" pane on the left, choose "Options."
2. You can select specific options like "Overwrite the existing database" or "Preserve the replication settings."
3. Adjust the settings based on your requirements.

Step 7: Review and Start Restore:

1. Click on the "Script" button if you want to generate a script instead of directly executing the restore operation. This step is optional.
2. Click "OK" to start the restore process.

Step 8: Monitor the Restore Process:

1. The progress of the restore operation will be displayed in the SSMS window.
2. Once the restore is complete, you should see a success message.

Step 9: Verify the Restore:

1. Expand the "Databases" node in the Object Explorer to check if the restored database is listed.

## Steps to Deploy a Web Application to IIS – server side

Step 1: Copy Files to the Server:

1. Copy the packaged version of your web application to the server where IIS is installed. You can use various methods like FTP, SCP, or shared network folders.

Step 2: Open IIS Manager:

1. On the server, open **Internet Information Services (IIS) Manager**.

Step 3: Create a New Site:

1. In IIS Manager, navigate to the server node.
2. Right-click on "Sites" and choose "Add Website."
3. Provide a **Site name**, set the **Physical path** to the directory where your web application is located, and assign a **Port** (e.g., 80 for HTTP).
4. Click "OK" to create the new site.

Step 4: Configure Application Pool:

1. In IIS Manager, go to the "Application Pools" node.
2. Create a new application pool or use an existing one.
3. In the "Advanced Settings" of the application pool, ensure that the correct version of the .NET CLR is selected (e.g., ".NET CLR Version" for a .NET application).
4. Assign the application pool to your website.

Step 5: Configure Web Application:

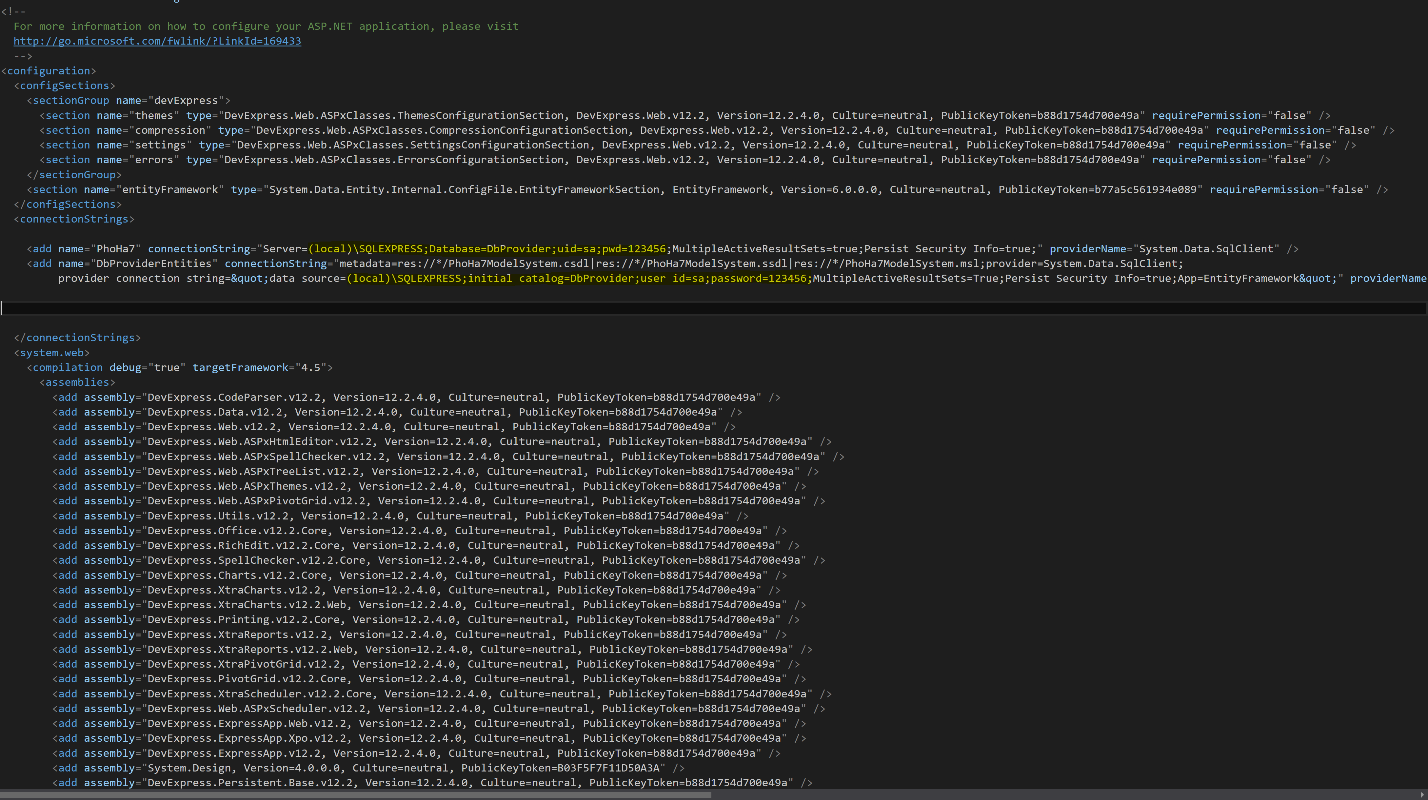
1. Under your site in IIS Manager, select "Explore."
2. Ensure that your web application files are in the correct directory.
3. Check and modify settings in the "Features View" to match your application's requirements (e.g., authentication, authorization).

Step 6: Test the Web Application:

1. Open a web browser and navigate to the URL of your web application (e.g., [http://localhost](http://localhost/)).

**Notes:**

Web.config Configuration: Double-check the web.config file for any environment-specific settings. Replace the server name, SQL server information by highlighting.



## Steps for the Client (KItchen display screen User):

Step 1: Click the Installation file Setup.exe:

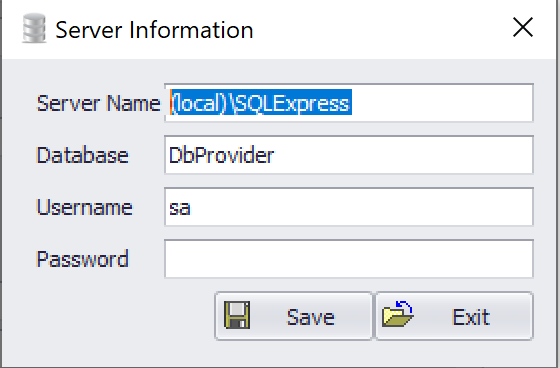
1. End users click on the installation link you provided.

Step 2: Follow the Installation Wizard:

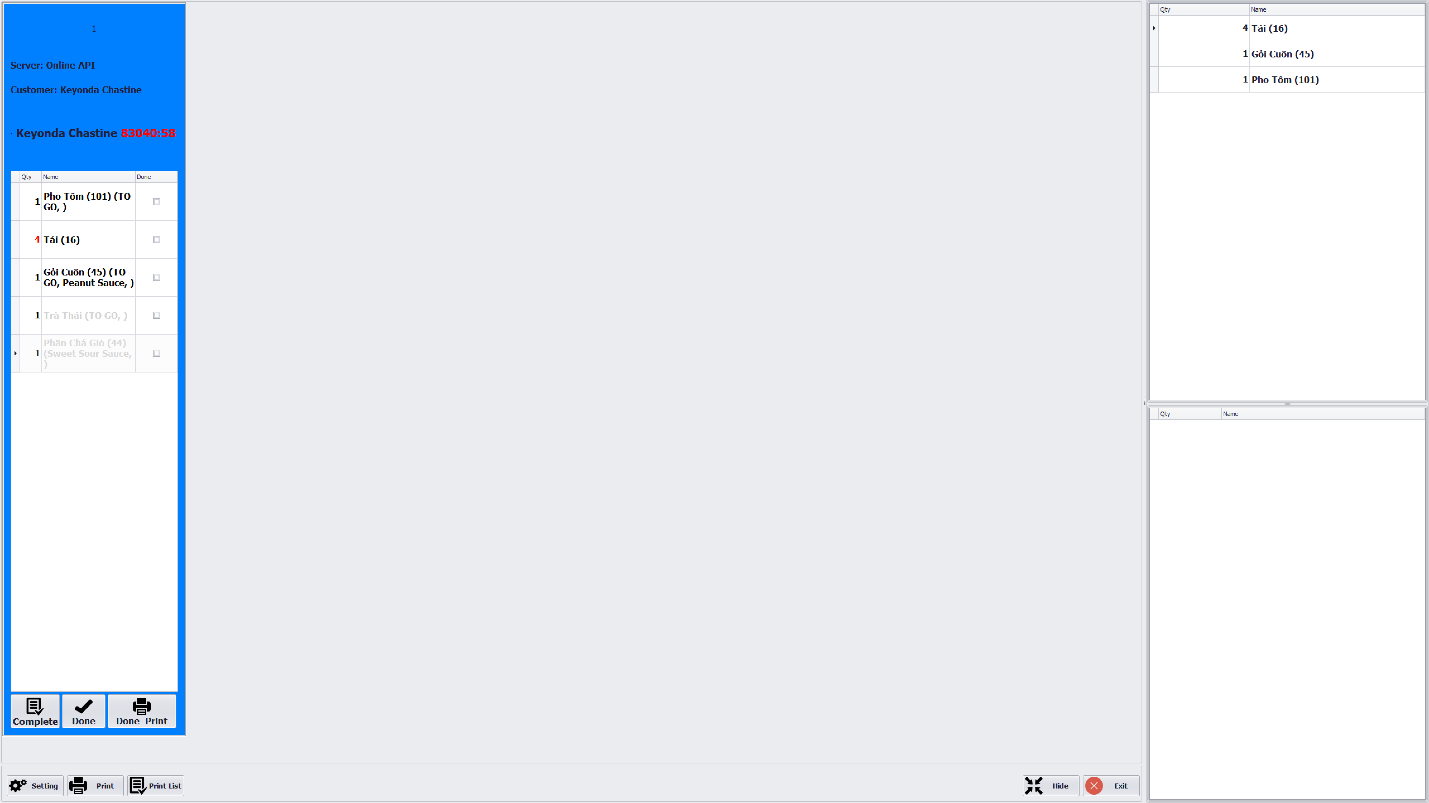
1. Users will see an installation wizard that guides them through the installation process.
2. The application will be downloaded and installed on their machines.

Step 3: Run the Application:

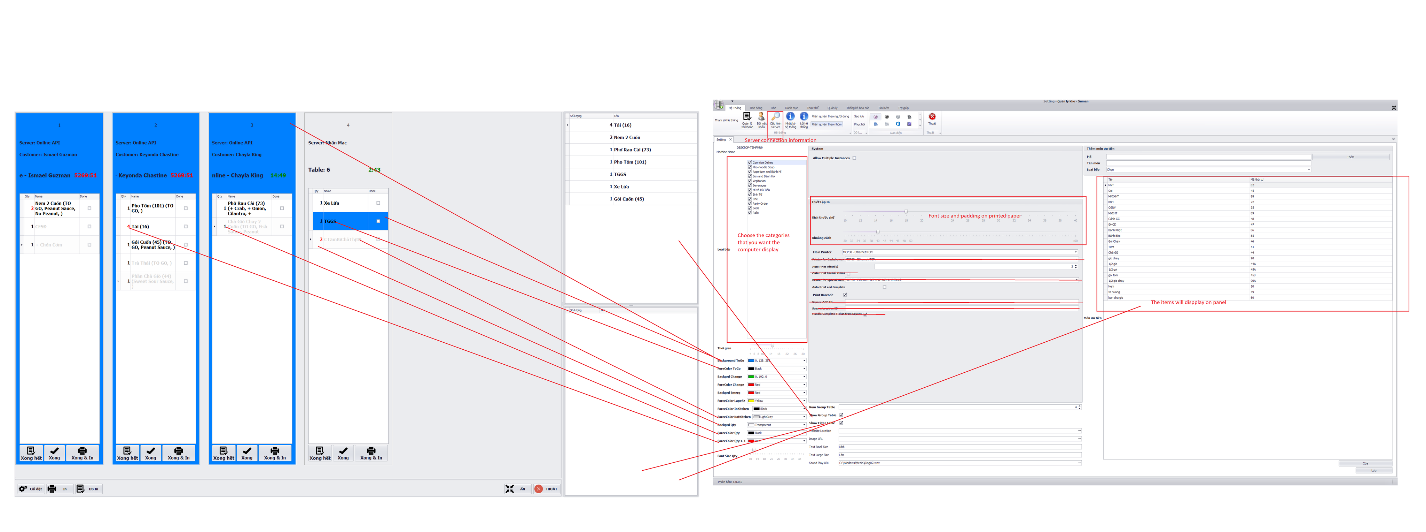
1. Once the installation is complete, the application is ready to run.
2. Users can find the application in their Start menu or desktop.
3. Application start in the first time will ask for configuration server information. Input the information that entered in SQL Server installation step



1. After the application start. Kitchen display screen will display.



1. [Following the picture to configuration the application](KitchentDisplayScreentConfig.png)



# User Manual

## Instructions for Using the Web Application for the First Time:

Step 1: Install Square Point of Sale on iOS devices:

A screenshot of a phone

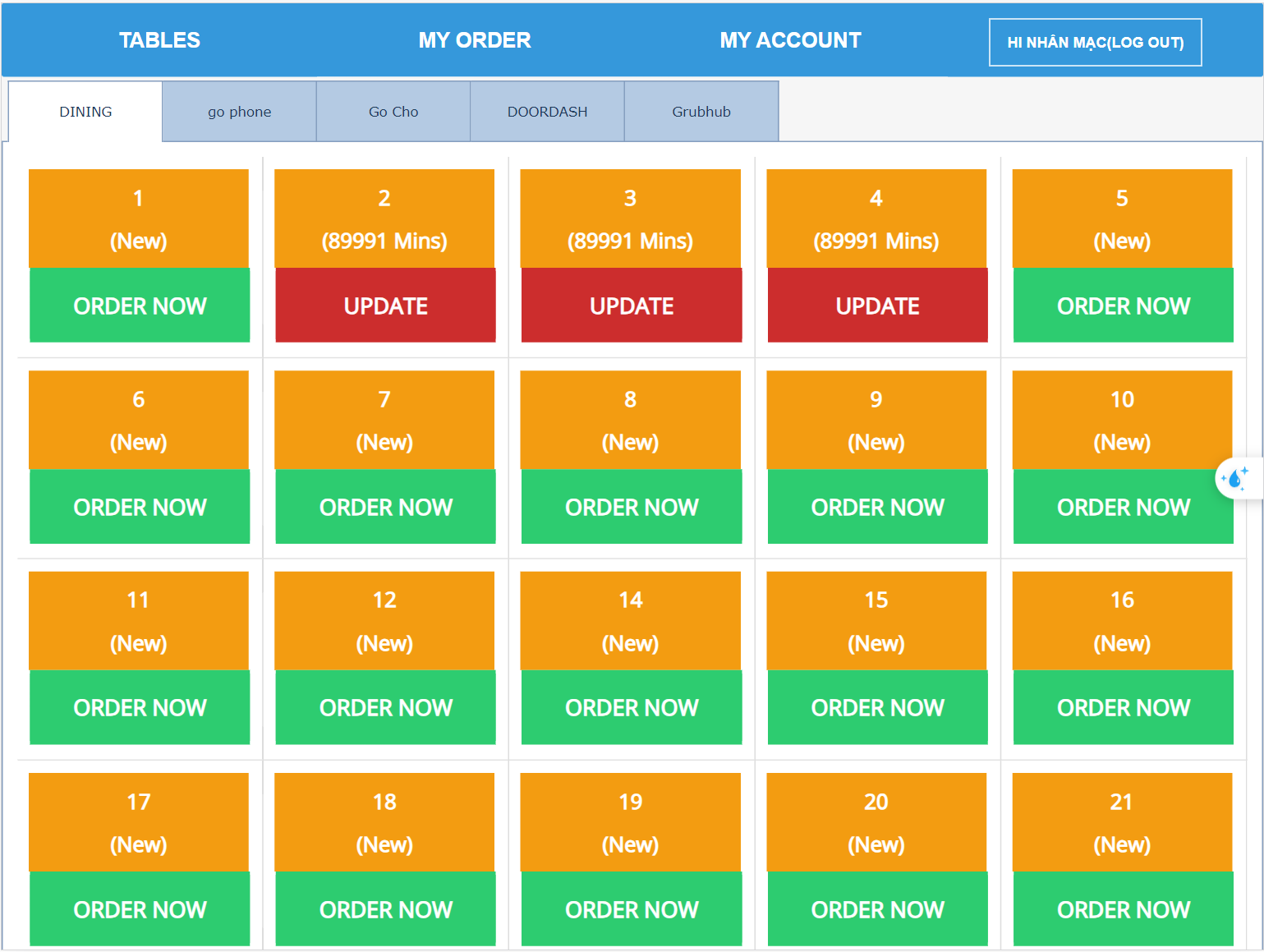
Description automatically generated

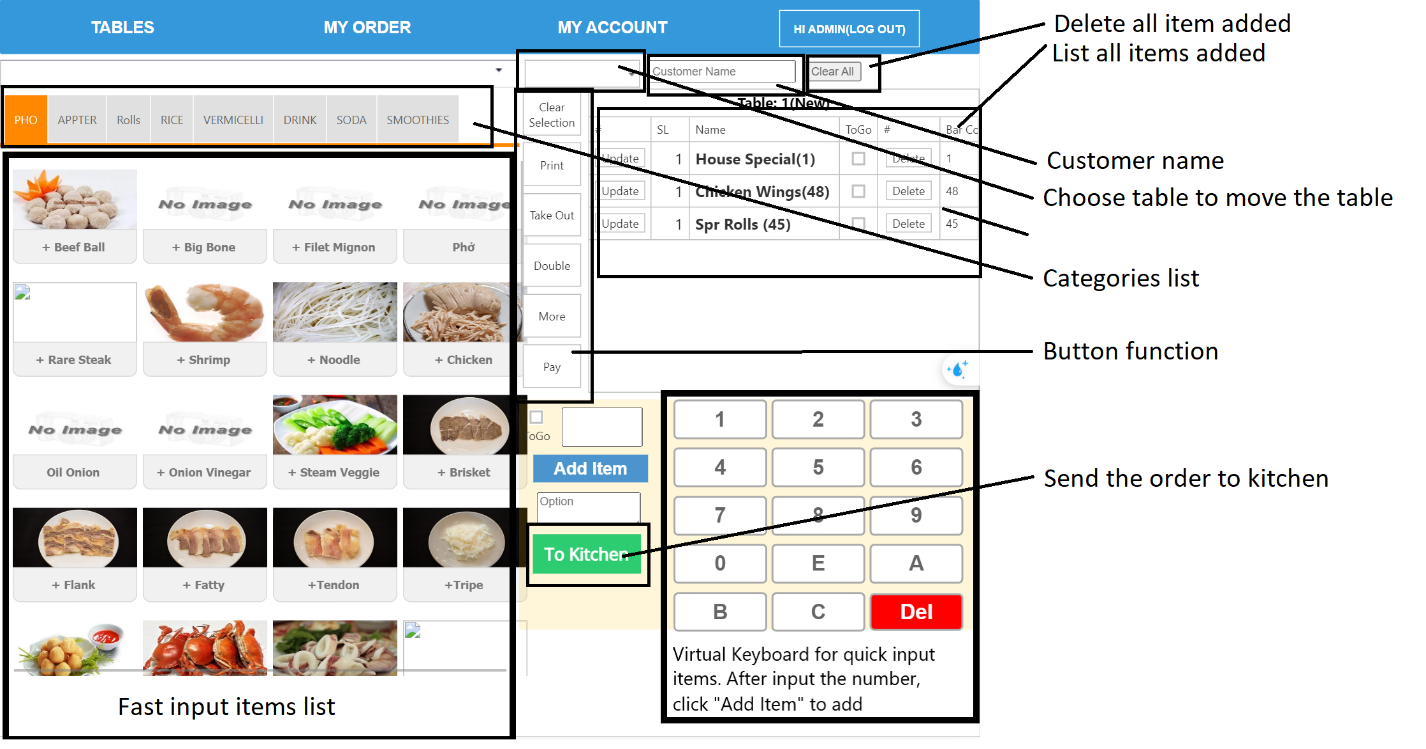
Step 2: Login to the Application:

1. Open the web application in your preferred web browser.
2. On the login page, enter "admin" as the username.
3. Enter the passcode "147852369" as the password.
4. Click the "Login" button to access the application.

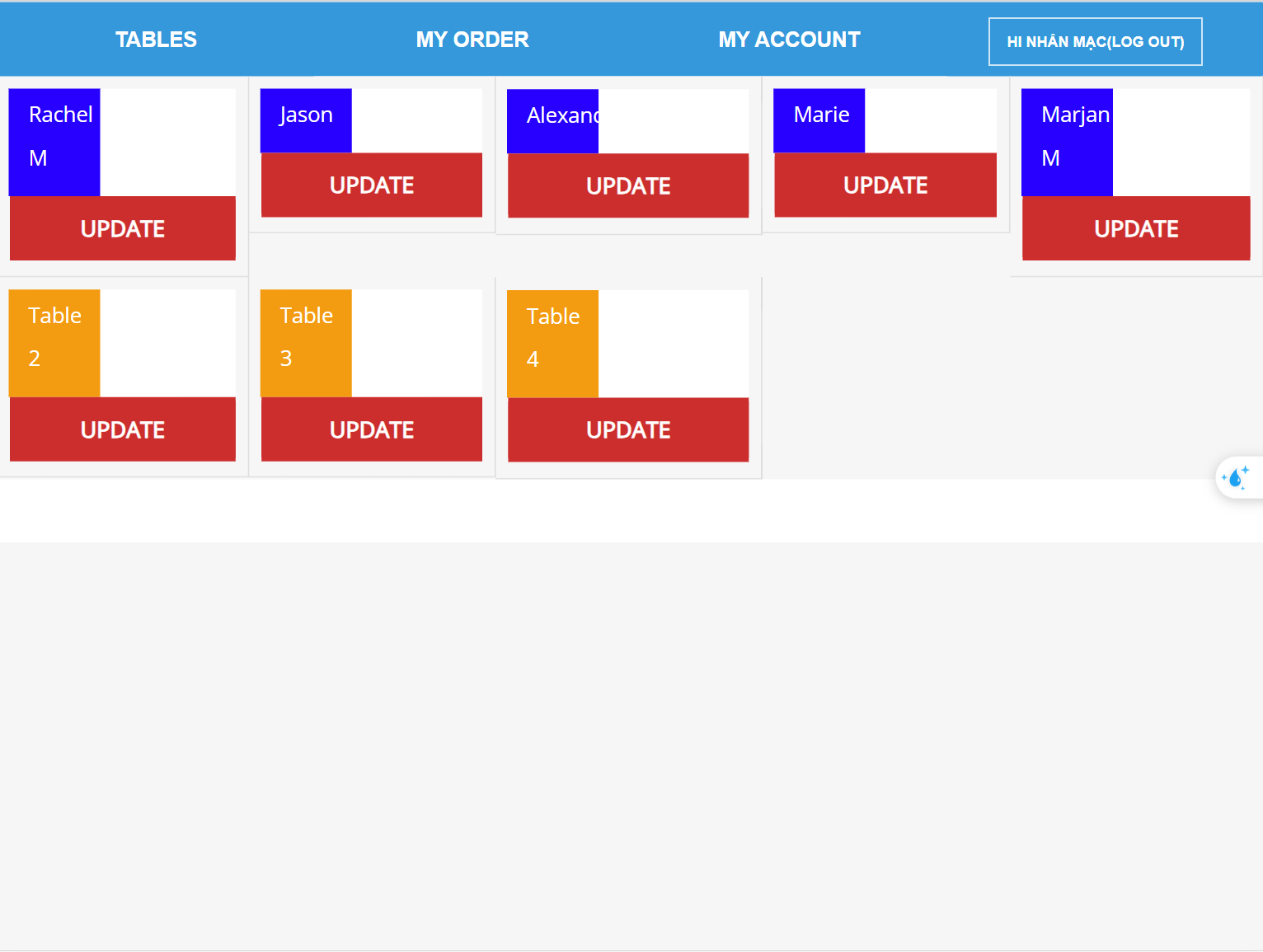
Step 3: Navigate the Main Tabs:

1. Explore the three main tabs:
   * **Table Tab**: Lists all available tables in the restaurant.

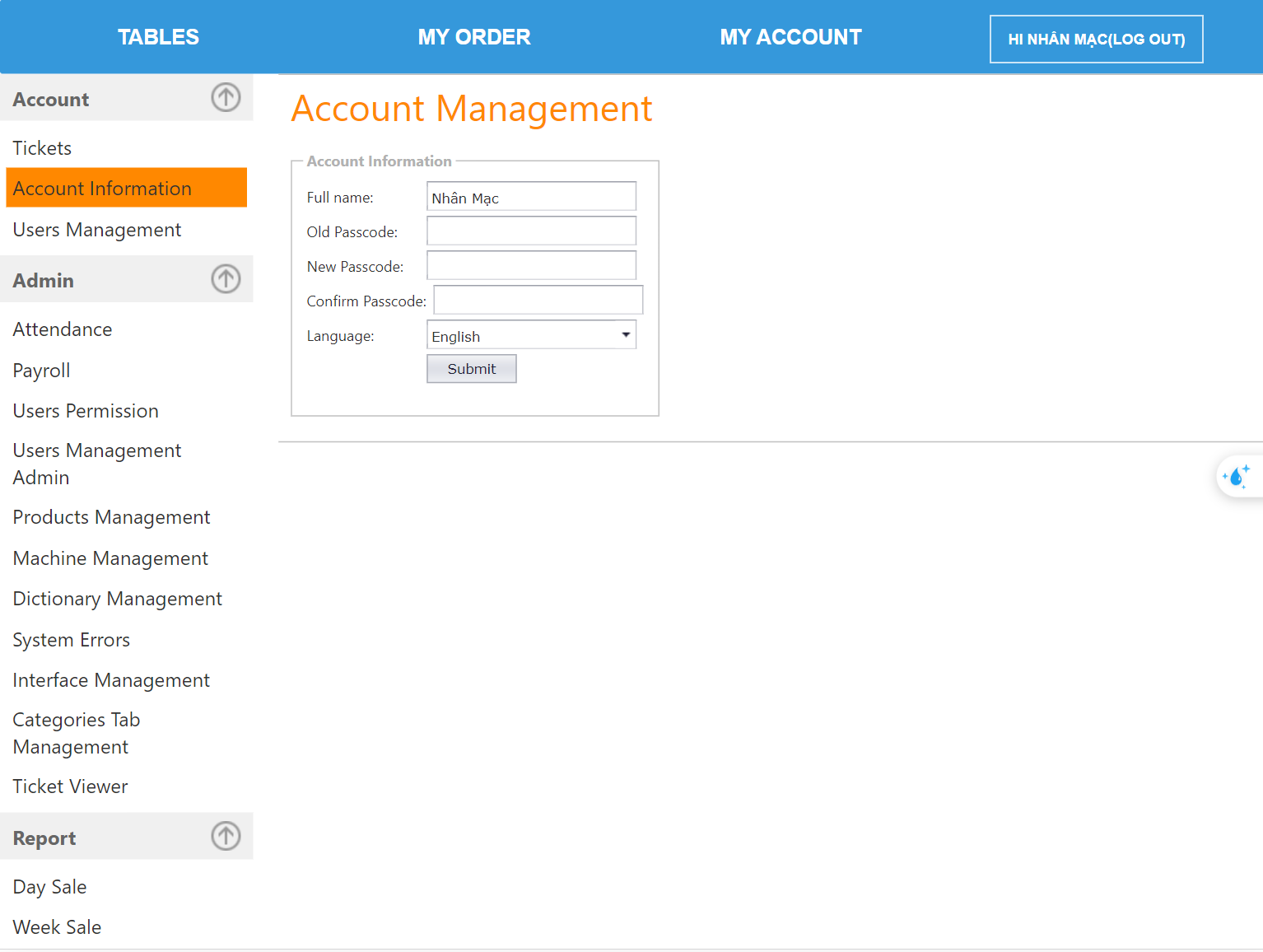
****



* + **My Order Tab**: Displays tables where the user has taken orders.

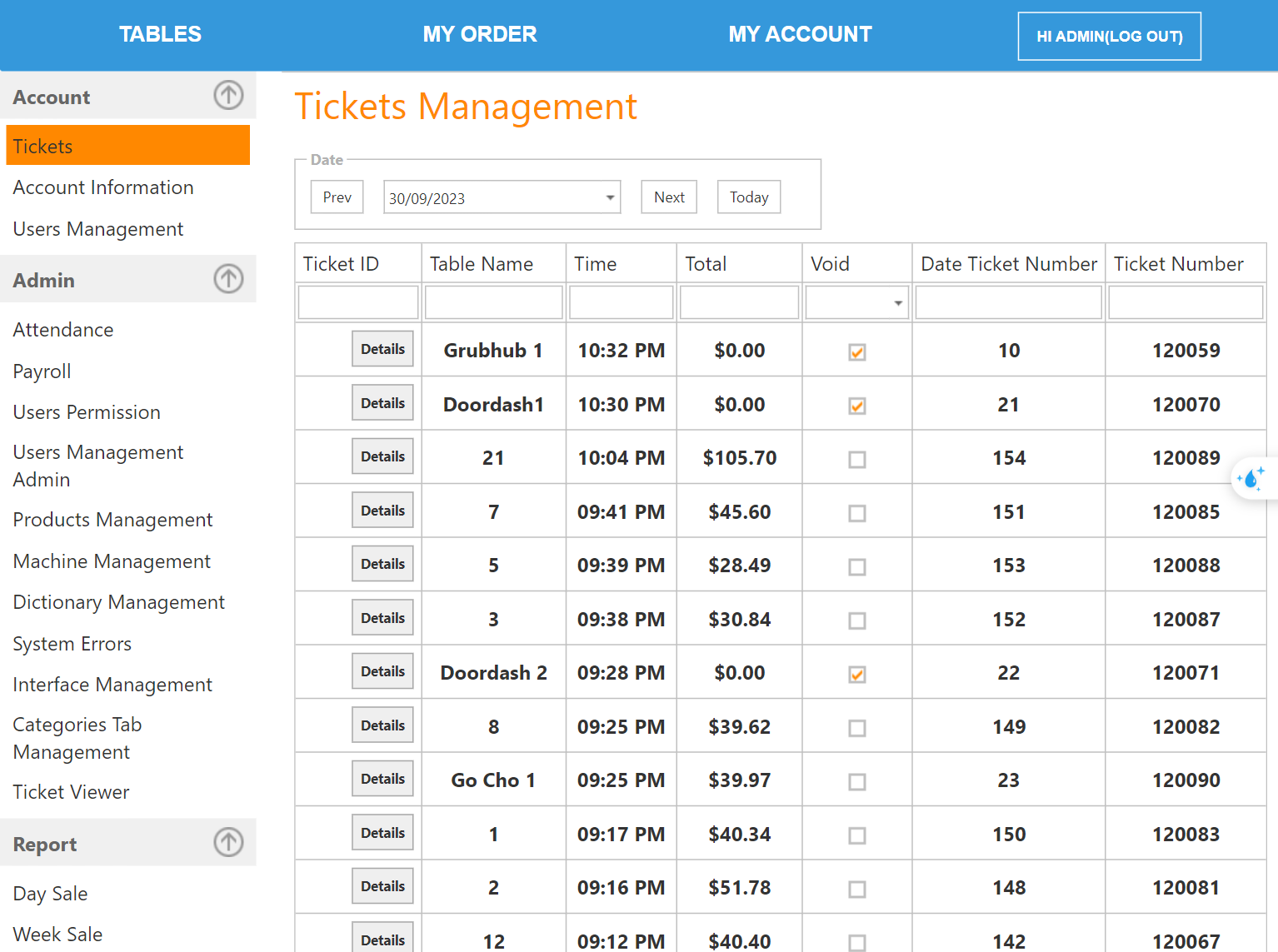


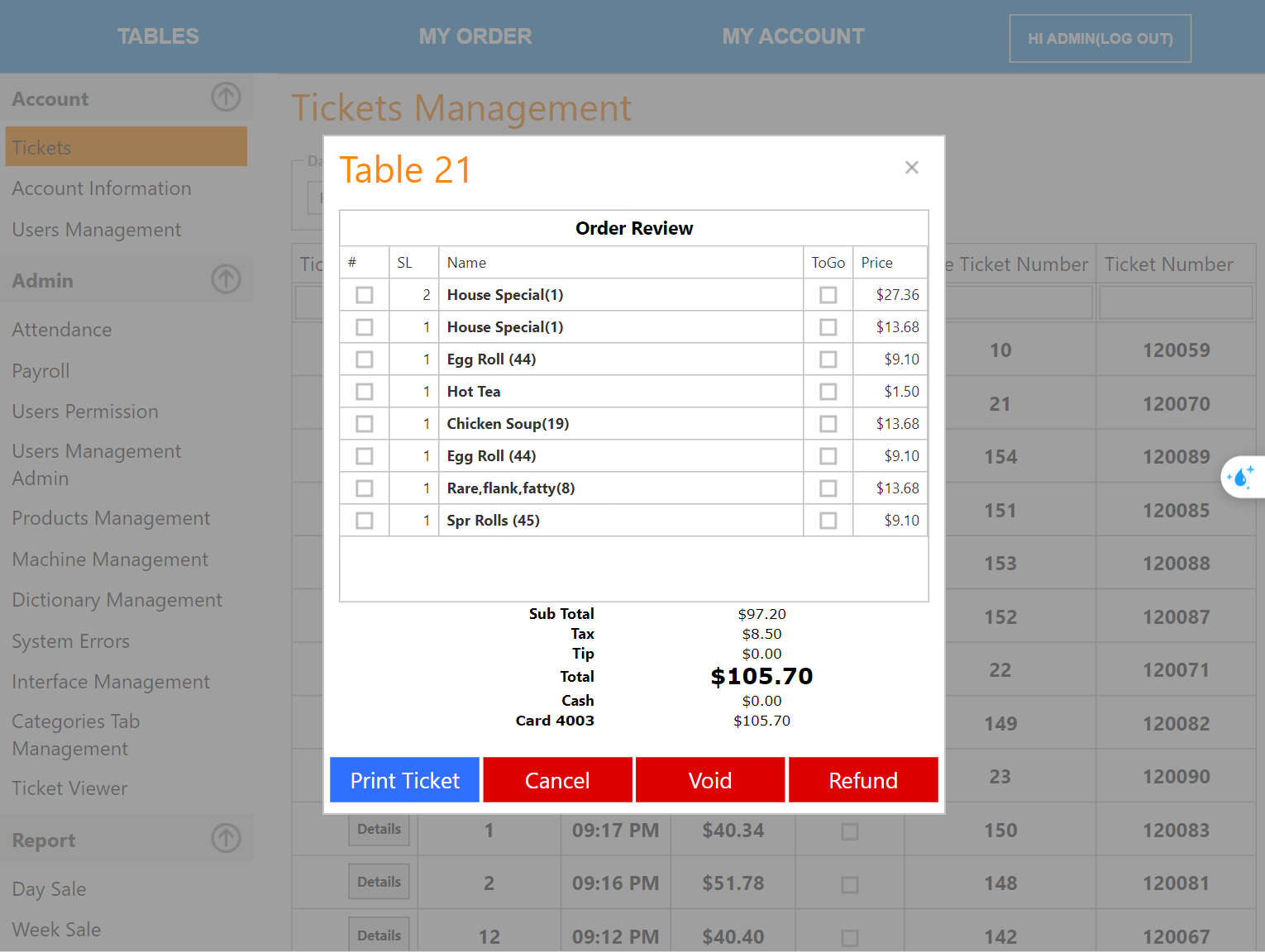
* + **My Account Tab**: Provides management functionalities for the application.



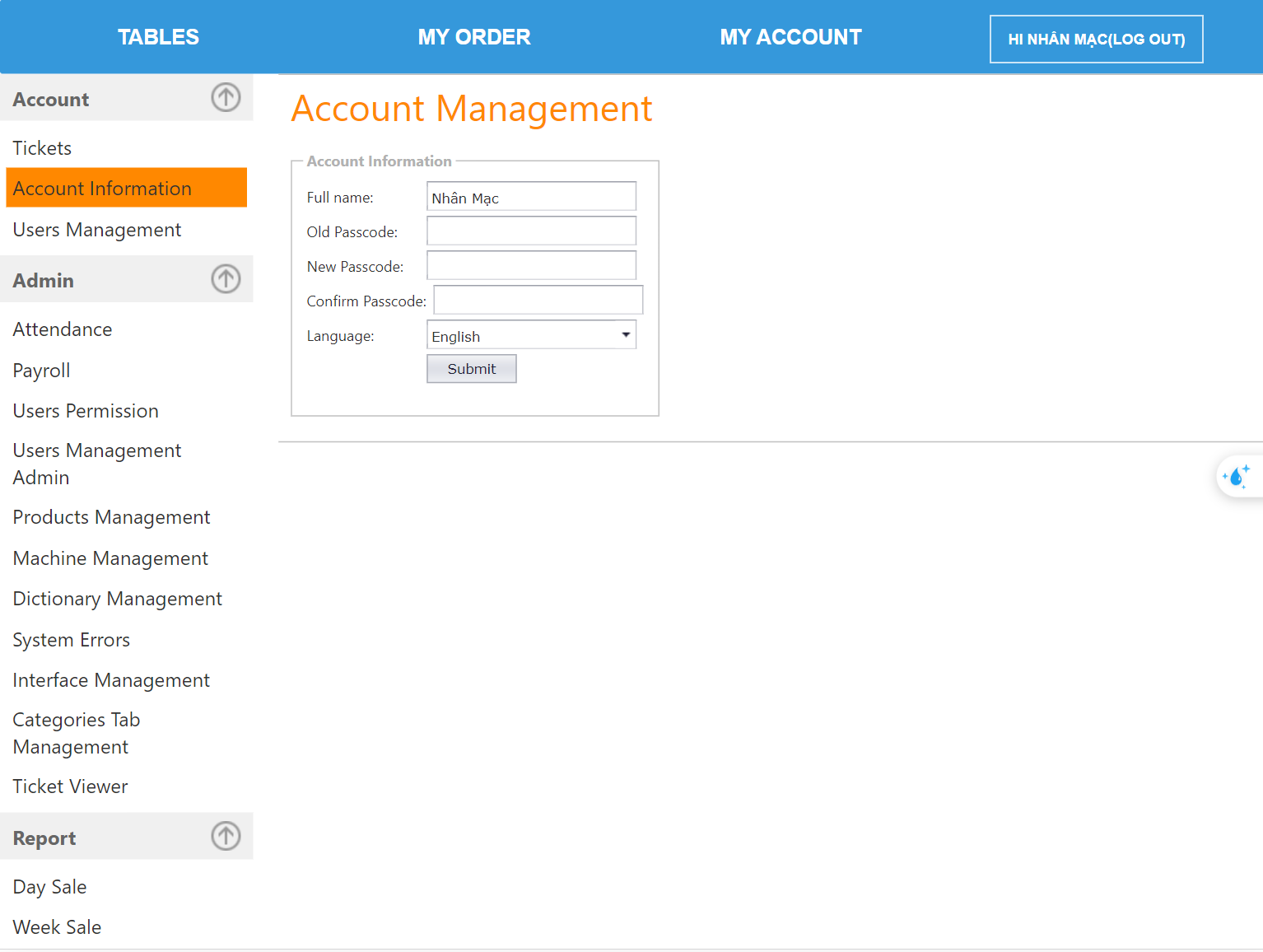
Step 4: Explore My Account Tab:

1. In the "My Account" tab, find the following functions:
   * **Tickets Management**: Manage tickets for orders.

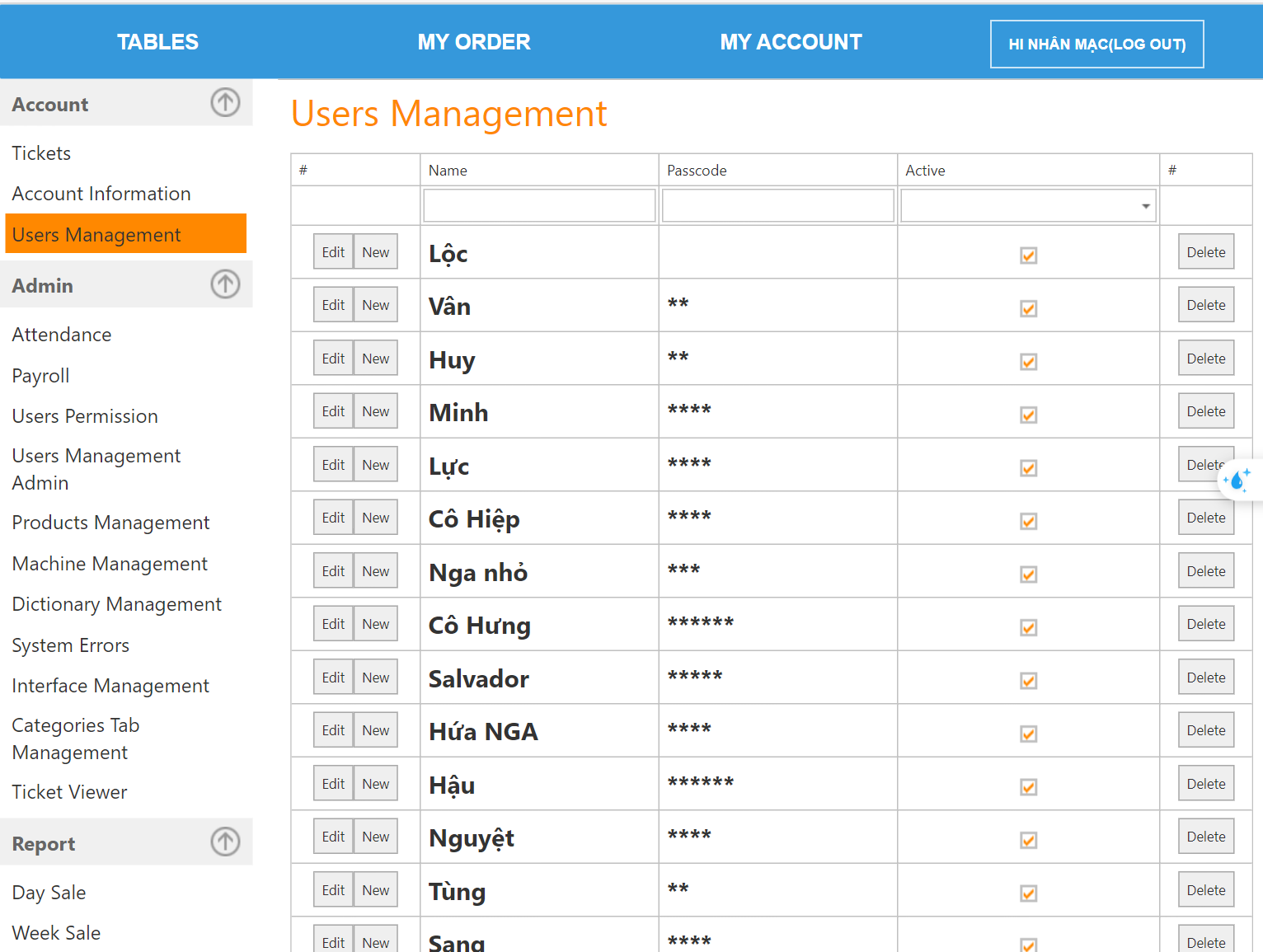




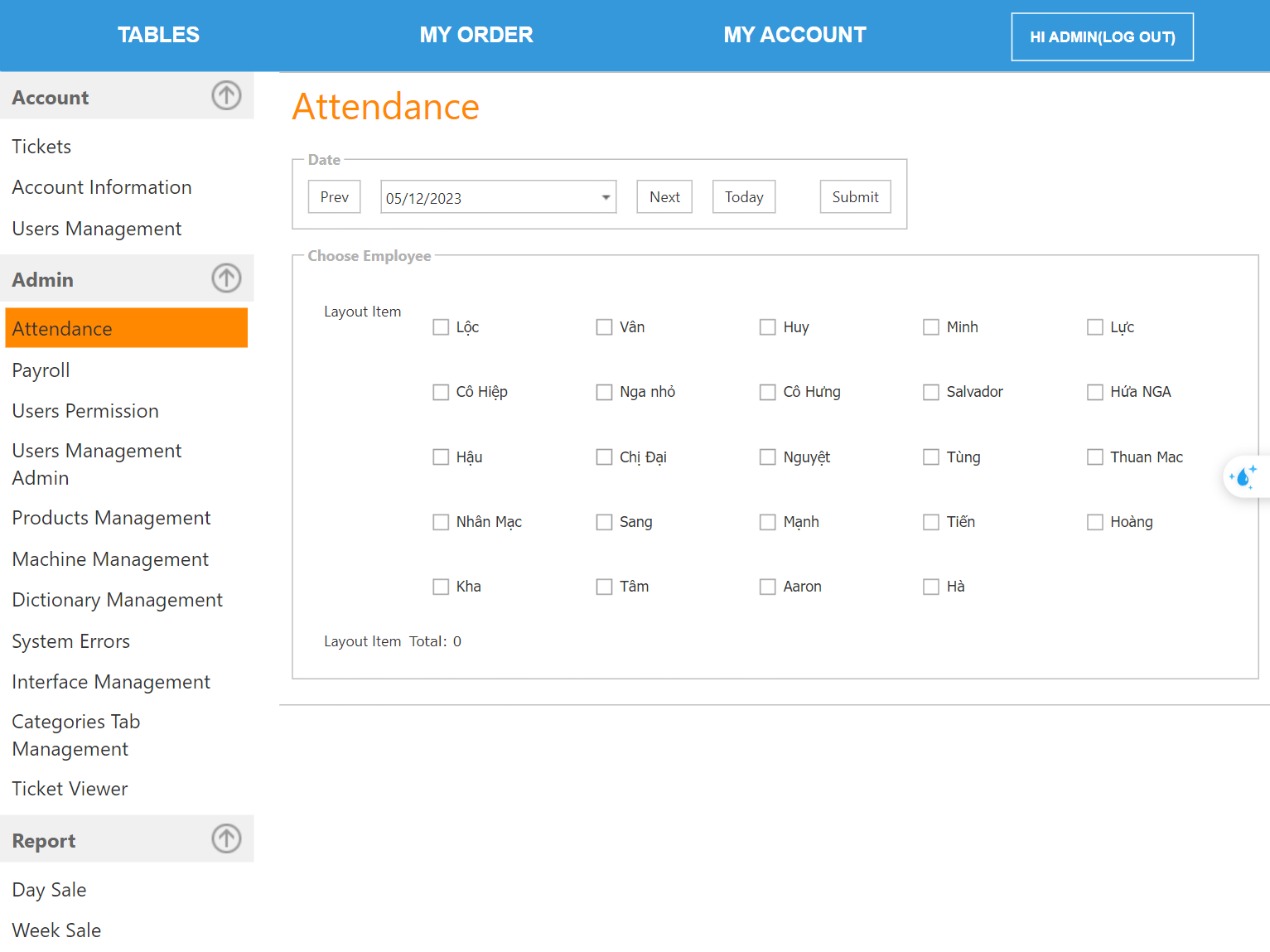
* + **Account Information**: View and modify user account details.



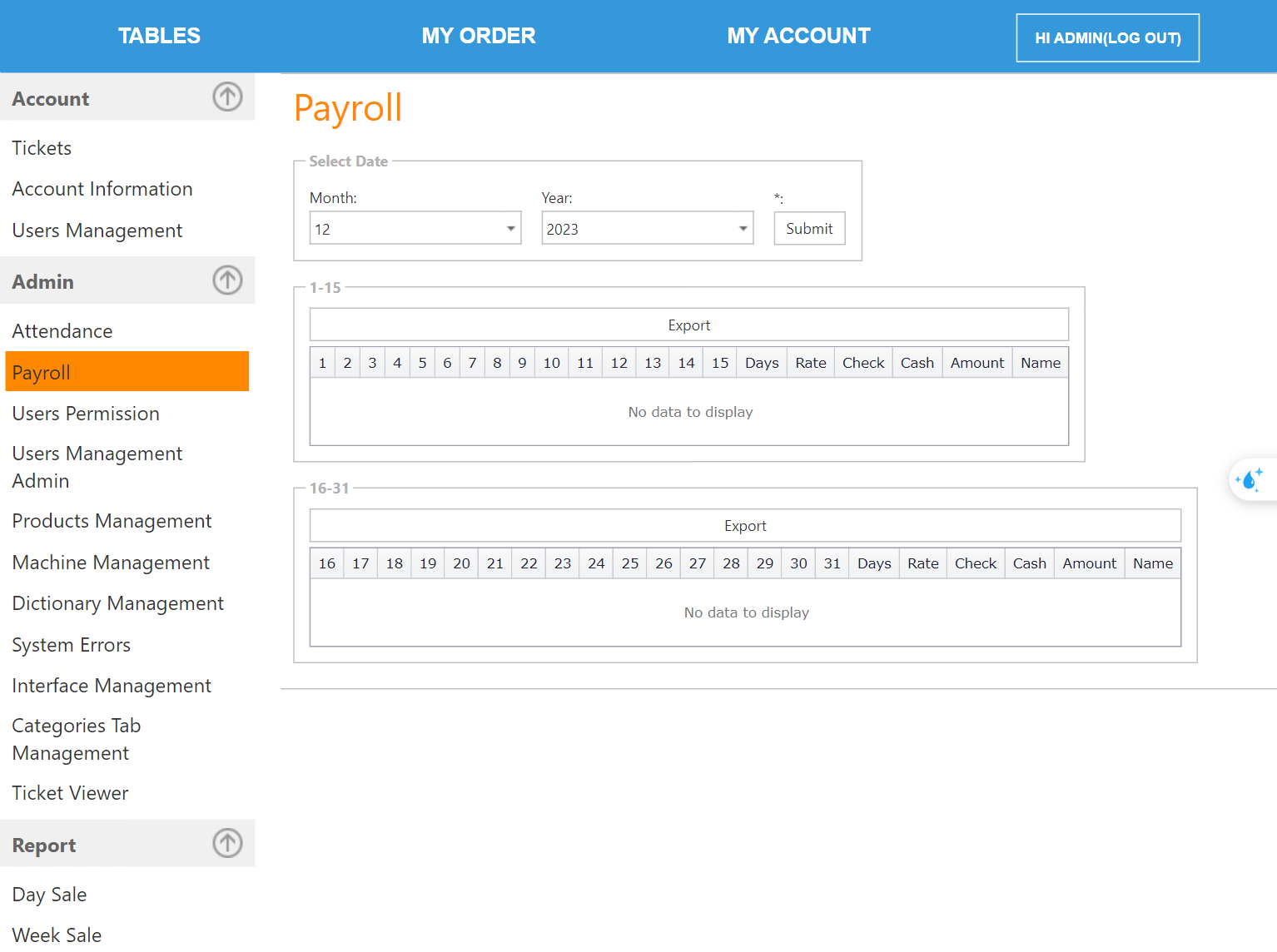
* + **Users Management**: Manage user accounts.



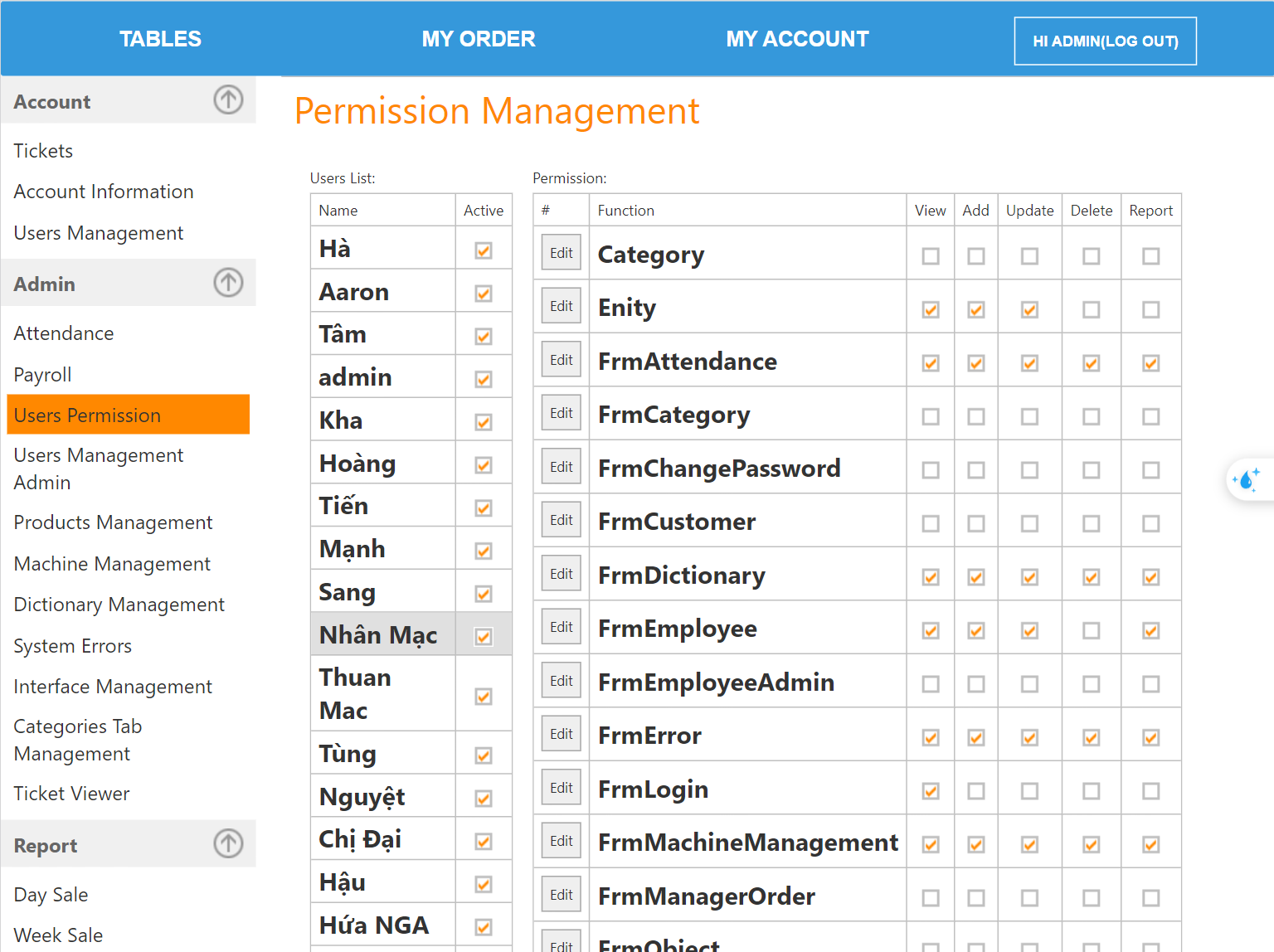
* + **Attendance**: Track attendance records.



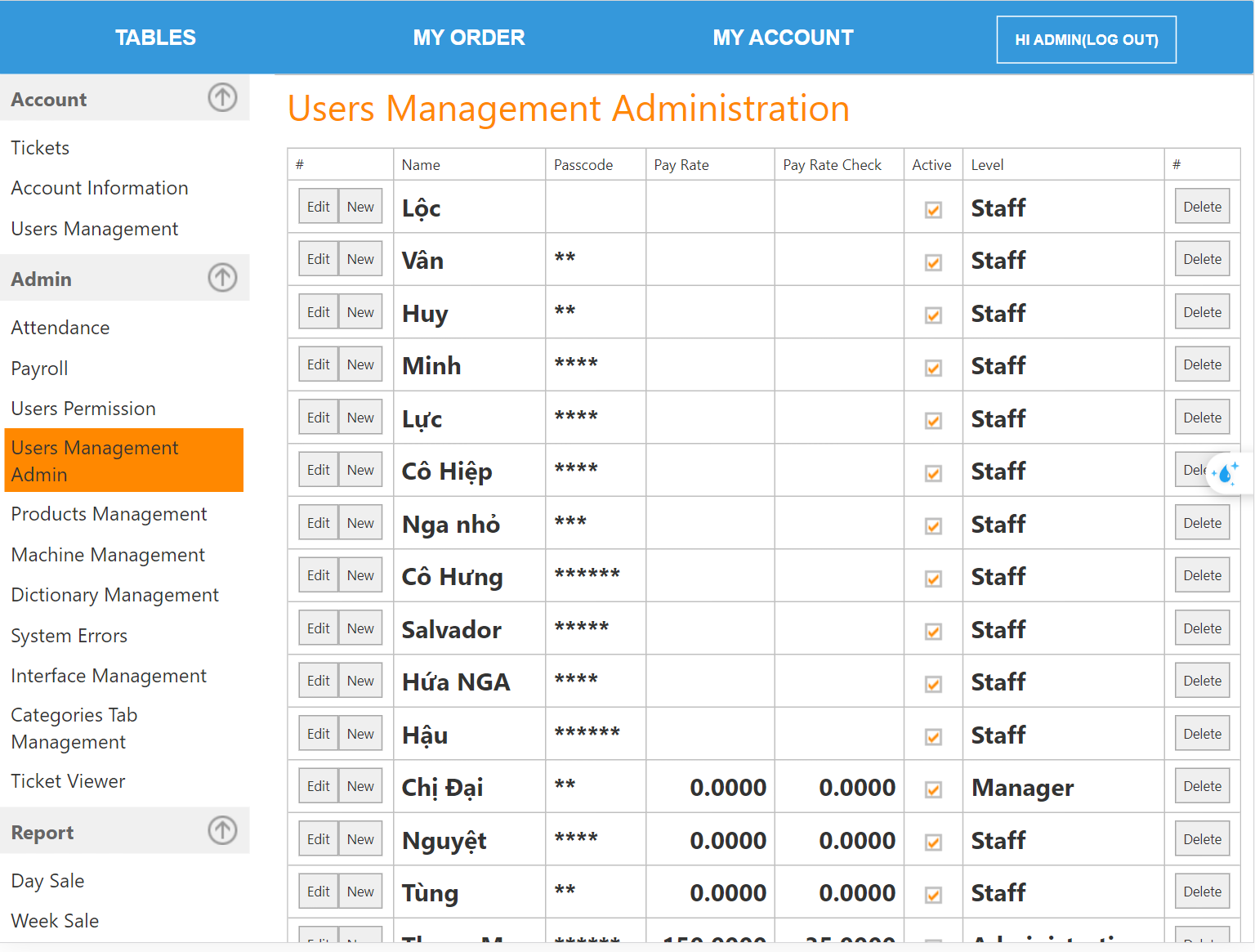
* + **Payroll**: Manage payroll information.



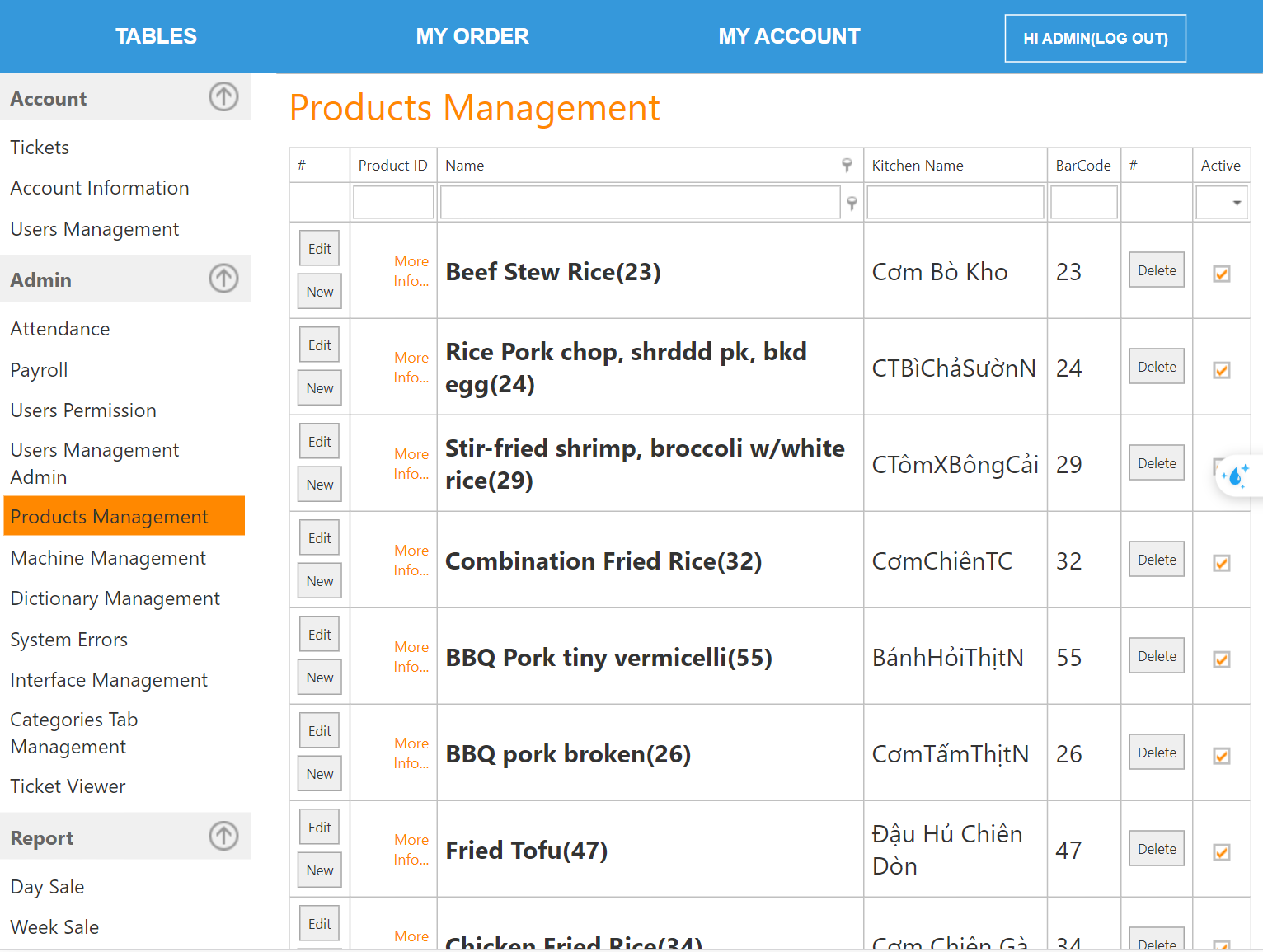
* + **Users Permission**: Set permissions for different user roles.



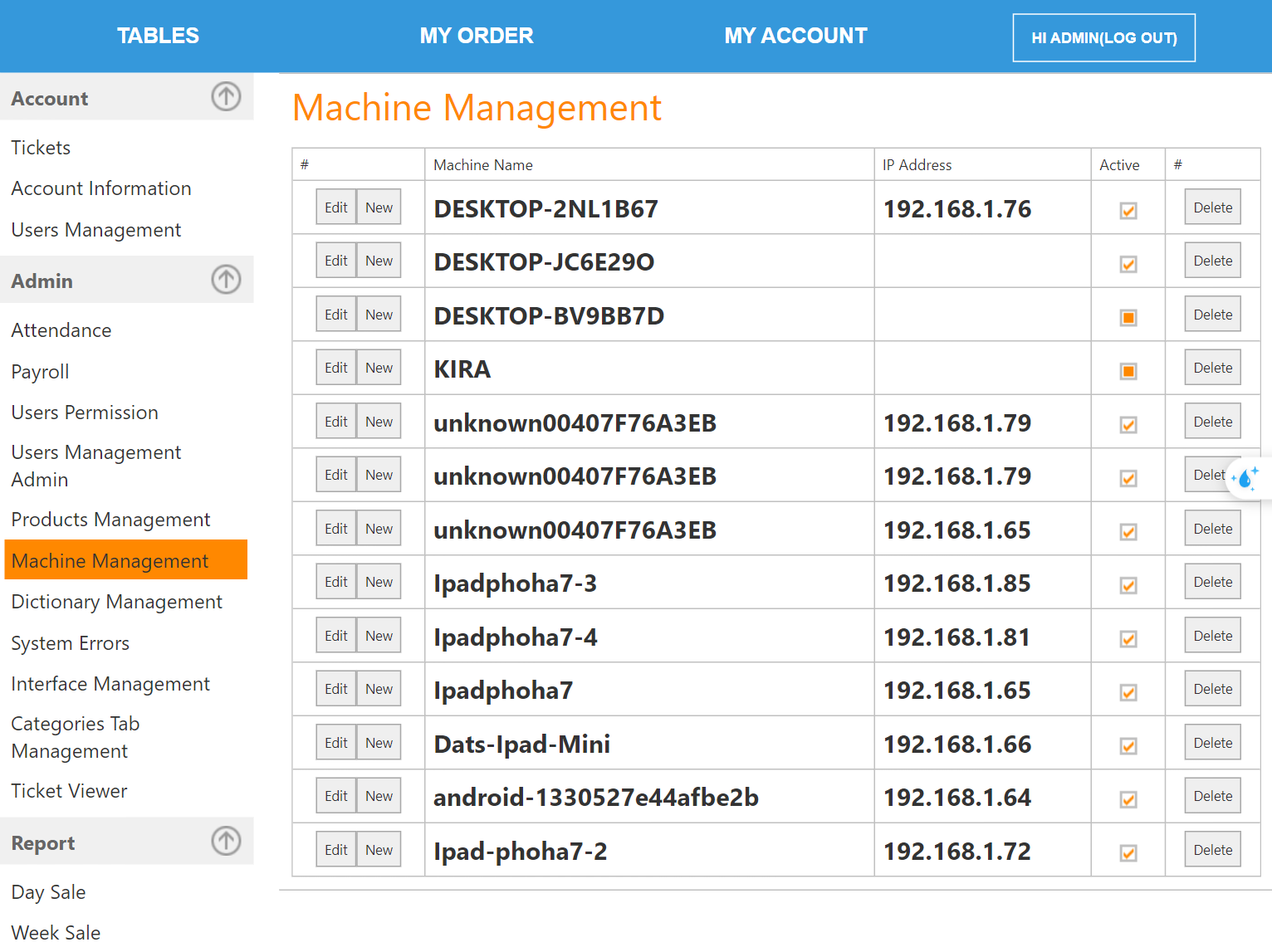
* + **Users Management Admin**: Admin-specific user management.



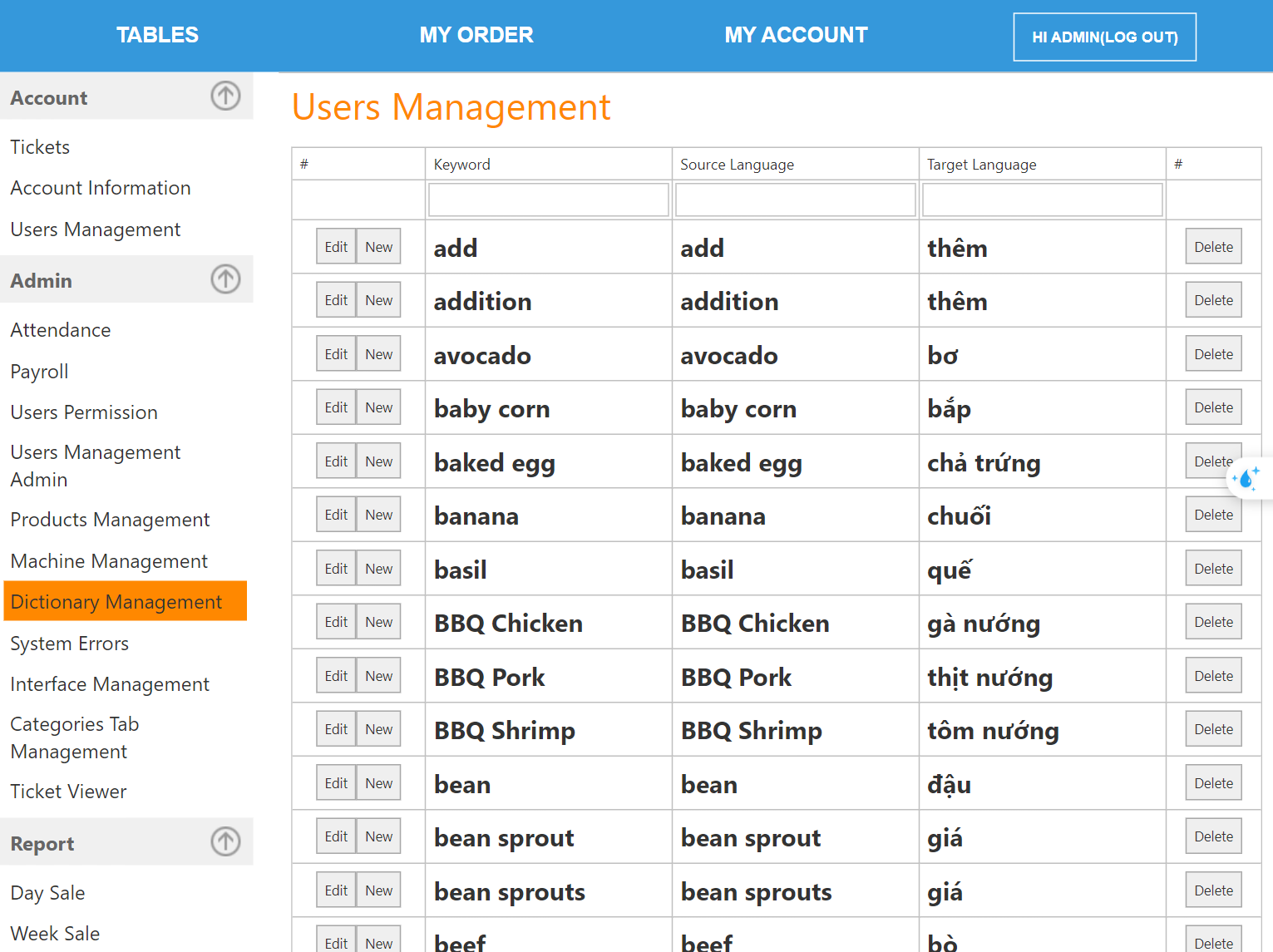
* + **Product Management**: Manage restaurant products.



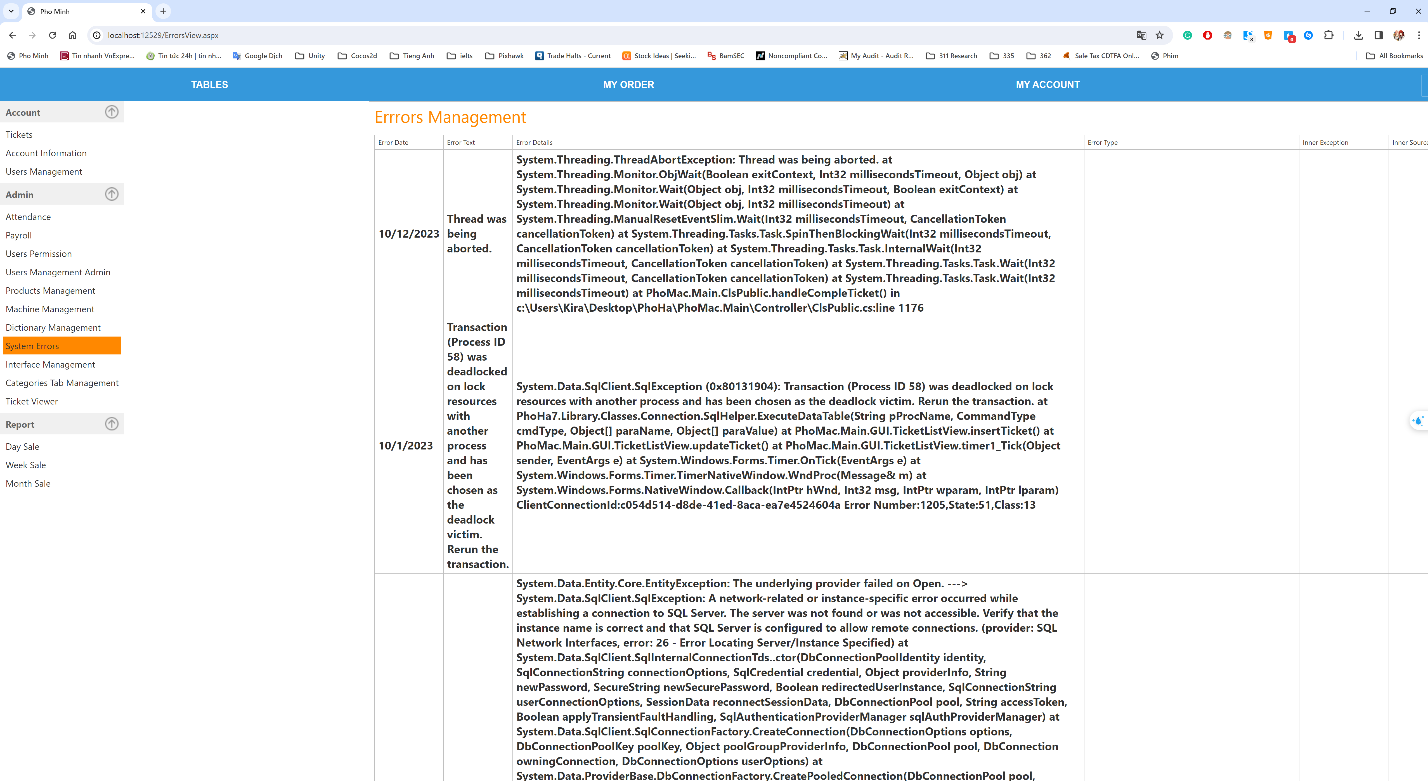
* + **Machine Management**: Handle machine-related configurations.



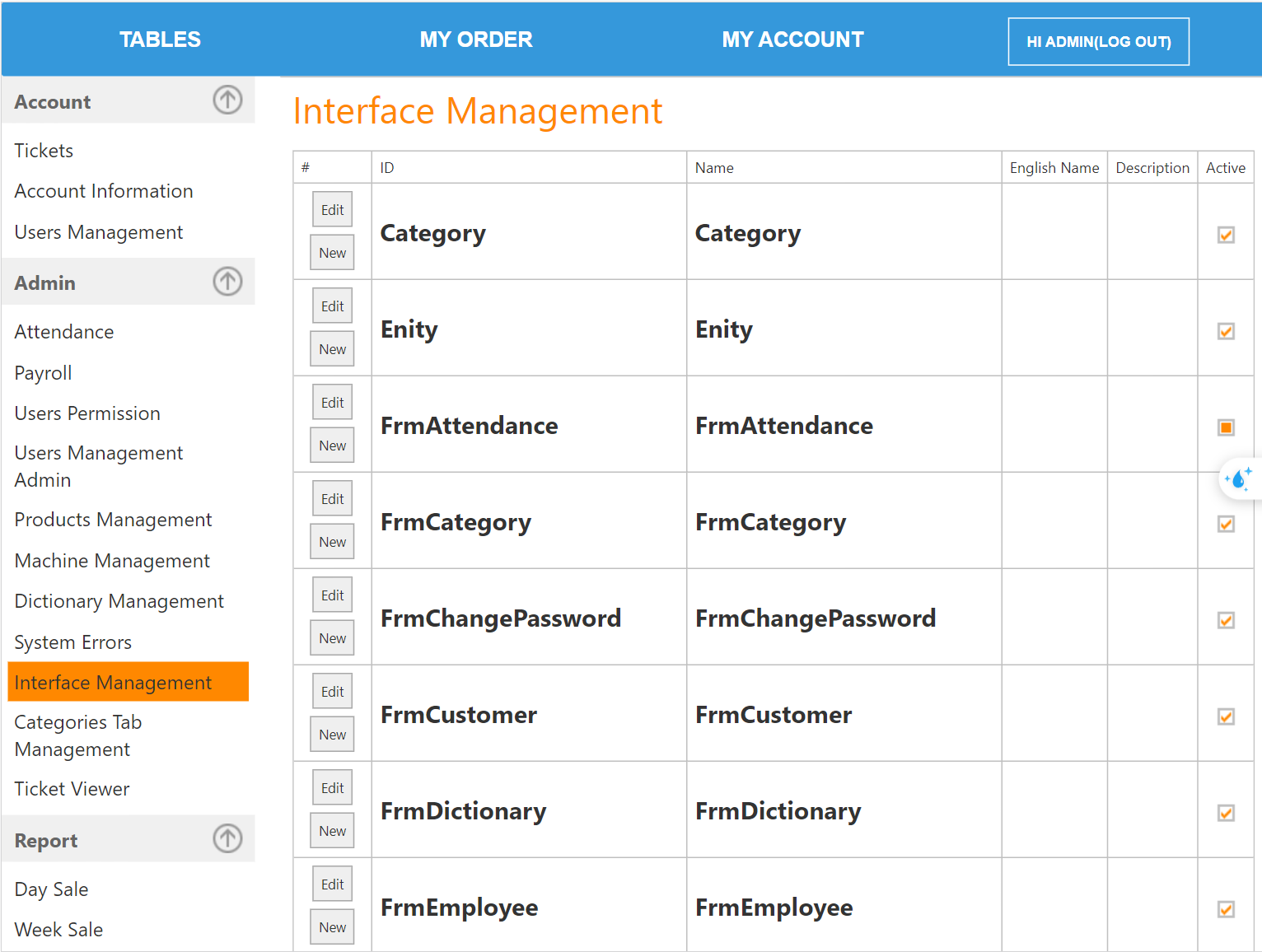
* + **Dictionary Management**: Manage application dictionaries.



* + **System Error Logs**: View logs for system errors.



* + **Interface Management**: Configure application interfaces.



* + **Categories Tab Management**: Manage categories for tables and orders.

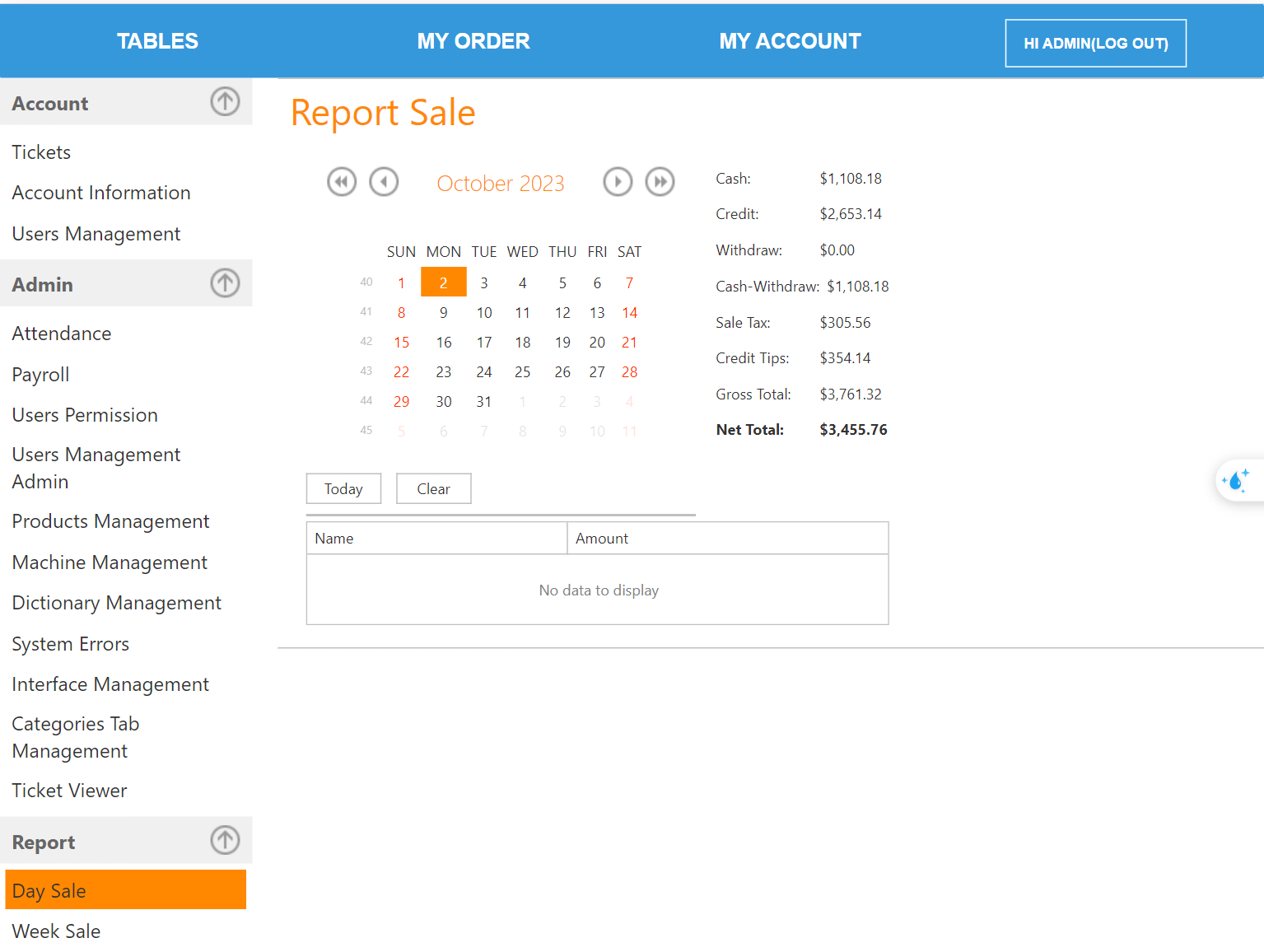
A screenshot of a menu

Description automatically generated

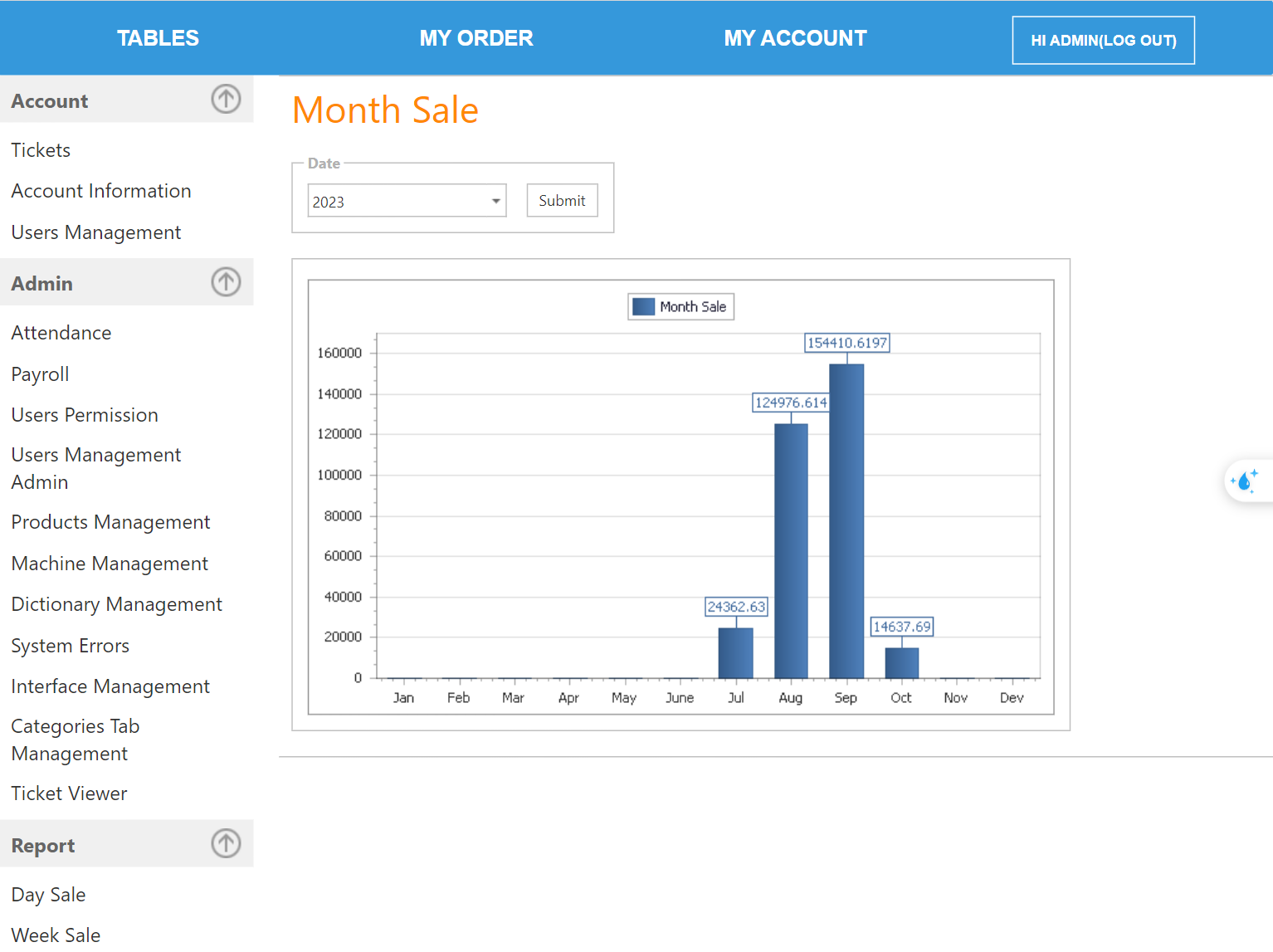
* + **Ticket Viewer**: View and manage customer orders.



* + **Daily Sale Report**: Access daily sales reports.



* + **Weekly Sale Report**: Access weekly sales reports.



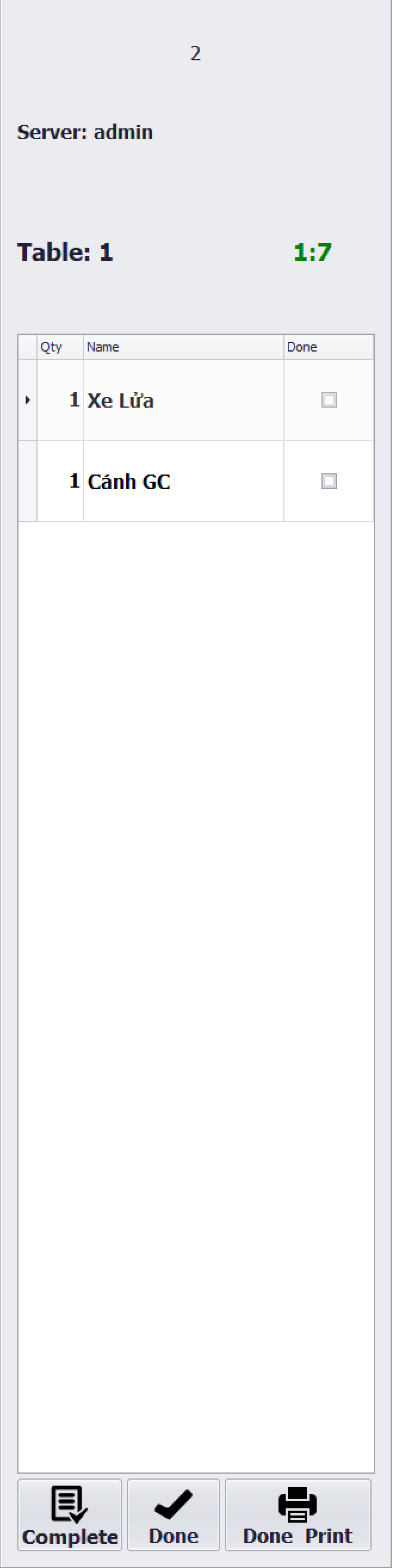
* + **Monthly Sale Report**: Access monthly sales reports.

A screenshot of a computer screen

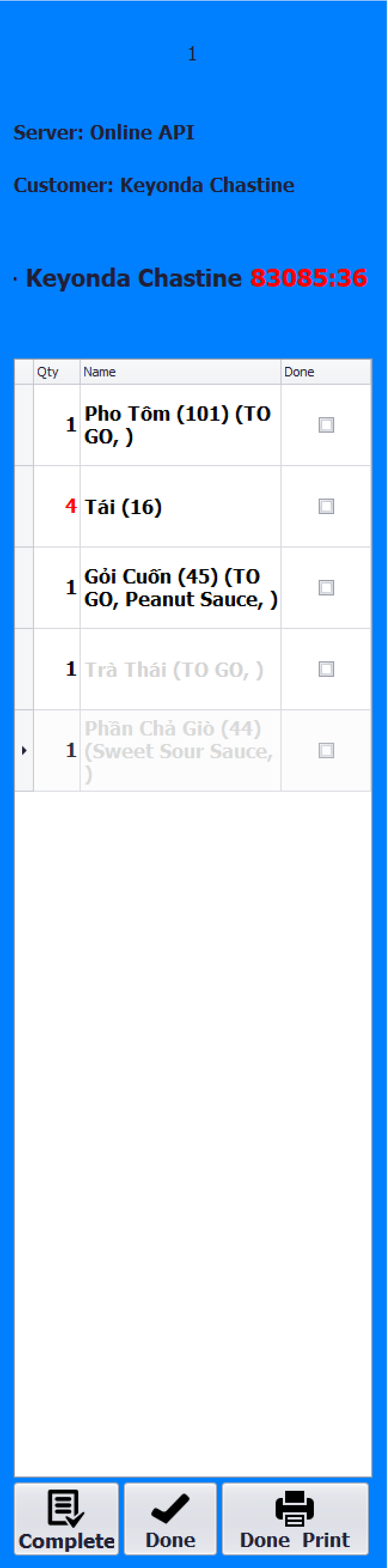
Description automatically generated

## Kitchen Display screen – client side

1. Dining ticket: the ticket will show up without color. When the cooker have a complete the whole ticket, click “Complete”. When the cooker has completed one item, choose the item completed and then click “Done & Print”



1. Take out Ticket: ticket will show up with blue color, similar to dining ticket when cooker finish the item.



**References.**

“Square Developer Documentation.” Square Developer, developer.squareup.com/docs. Accessed 29 Sept. 2023.