**VIETNAM – KOREA UNIVERSITY OF**

**INFORMATION AND COMMUNICATION TECHNOLOGY**

**Faculty of Computer Science**



**BASIC PROJECT 1**

**BOTSHOP - SHOPPING MANAGEMENT**

Student: **Ho Gia Bao**

Student ID: **23IT011**

Class: **23GIT**

Instructor: **Ph.D.** **Nguyen Van Loi**

**Da Nang, June 2024**

**VIETNAM – KOREA UNIVERSITY OF**

**INFORMATION AND COMMUNICATION TECHNOLOGY**

**Faculty of Computer Science**



**BASIC PROJECT 1**

**BOTSHOP - SHOPPING MANAGEMENT**

Student: **Ho Gia Bao**

Student ID: **23IT011**

Class: **23GIT**

Instructor: **Ph.D.** **Nguyen Van Loi**

**Da Nang, June 2024**

**FOREWORD**

In the modern era, the continuous advancement of technology has transformed the way we interact with the world around us, especially in the fields of shopping and business. In this context, the development and deployment of smart applications have become an indispensable part, providing convenience and better user experiences.

The project I present in this report is BotShop - a desktop application for managing online shopping, especially for individual sellers. BotShop is the result of my efforts, patience and dedication, with the aim of providing a comprehensive and effective shopping management solution for individual sellers, while also bringing benefits to end users in the shopping process.

BotShop is not just a regular online shopping application, but also a multi-functional sales management system. It connects individual sellers (Admin), buyers (User) and delivery staff (Shipper) in a convenient and flexible environment, helping them optimize their business operations.

In the following part of the report, I will present in detail the development process of BotShop, including the technologies used, the design and implementation of the system, as well as the testing and evaluation of the results achieved. I hope that through this report, BotShop can make a small contribution to the field of smart application development and create the best shopping experiences for both sellers and buyers.

**ACKNOWLEDGMENTS**

First of all, I would like to express my deepest gratitude to my instructor,   
Mr. Nguyen Van Loi, for his dedicated guidance throughout the project. His enthusiasm, dedication, and valuable advice have motivated me and helped me complete the project with outstanding results. I acknowledge and appreciate the precious knowledge and experience that he has shared with me.

Next, I would like to extend my sincere thanks to the lecturers of the Computer Science Department, who have nurtured and provided me with a solid foundation of knowledge during my studies at the university. Thanks to their dedication and passion, I have gained the confidence to embark on the project and complete it effectively.

I would also like to express my sincere gratitude to my classmates and colleagues in the Information Technology major, who have been wonderful companions, always supporting, encouraging, and sharing their valuable knowledge and experience during the project. Their cooperation and assistance have greatly contributed to the project's success.

However, I am aware that this report still has many limitations due to my own shortcomings in terms of knowledge and experience. I would like to receive valuable feedback from my teachers and classmates to improve the project and myself in the future.

INSTRUCTOR’S COMMENTS

Da Nang, 2024 ………………………

Instructor

**TABLE OF CONTENTS**

[LIST OF ACRONYMS](#_Toc168262942)

[LIST OF IMAGES](#_Toc168262943)

[CHAPTER 1. GENERAL INTRODUCTION 1](#_Toc168262944)

[1.1 Overview 1](#_Toc168262945)

[1.1.1 Reason for choosing the topic 1](#_Toc168262946)

[1.1.2 Problems to be solved 1](#_Toc168262947)

[1.1.3 To-do list 1](#_Toc168262948)

[1.2 Project tasks 2](#_Toc168262949)

[1.2.1 Approach to project execution 2](#_Toc168262950)

[1.2.2 Result 2](#_Toc168262951)

[1.3 Project report structure 2](#_Toc168262952)

[CHAPTER 2: GENERAL RESEARCH 3](#_Toc168262953)

[2.1 Tools used 3](#_Toc168262954)

[2.1.1 NetBeans IDE 3](#_Toc168262955)

[2.1.2 SQL Server Management Studio 3](#_Toc168262956)

[2.2 Technologies used 3](#_Toc168262957)

[2.2.1 Java programming language 4](#_Toc168262958)

[2.2.2 Swing (Java) 4](#_Toc168262959)

[2.2.2 Structured Query Language (SQL) 5](#_Toc168262960)

[2.2.3 Hyper Text Markup Language (HTML) 6](#_Toc168262961)

[2.2.4 Cascading Style Sheets (CSS) 6](#_Toc168262962)

[2.3 Application Architecture 7](#_Toc168262963)

[CHAPTER 3. SYSTEM ANALYSIS AND DESIGN 8](#_Toc168262964)

[3.1 System analysis 8](#_Toc168262965)

[3.1.1 User functionality 8](#_Toc168262966)

[3.1.2 Shipper Functionality 9](#_Toc168262967)

[3.1.3 Admin Functionality 10](#_Toc168262968)

[3.2 System Design 11](#_Toc168262969)

[3.2.1 Overview Use – Case Diagram 11](#_Toc168262970)

[3.2.2 User Use-Case Diagram 12](#_Toc168262971)

[3.2.3 Shipper Use-Case Diagram 12](#_Toc168262972)

[3.2.4 Admin Use-Case Diagram 13](#_Toc168262973)

[3.2.5 Database Relationship Diagram 13](#_Toc168262974)

[Chapter 4. Interface Design 13](#_Toc168262975)

[4.1 User interface 14](#_Toc168262976)

[4.2 Shipper interface 18](#_Toc168262977)

[4.3 Admin interface 20](#_Toc168262978)

[CHAPTER 5. CONCLUSION AND FUTURE DIRECTIONS 25](#_Toc168262979)

[5.1 Conclusion 25](#_Toc168262980)

[5.1.1 Achievements 25](#_Toc168262981)

[5.1.2 Limitations 25](#_Toc168262982)

[5.2 Future directions 25](#_Toc168262983)

# LIST OF ACRONYMS

|  |  |  |
| --- | --- | --- |
| **No.** | **ACRONYMS** | **CONTENT** |
| 1 | SQL | Structured Query Language |
| 2 | HTML | Hyper Text Markup Language |
| 3 | CSS | Cascading Style Sheets |
| 4 | MVC | Model - View - Controller |

# LIST OF FIGURES

[Figure 2.1 NetBeans IDE 3](#_Toc168262998)

[Figure 2.2 SQL Server Management Studio 3](#_Toc168263000)

[Figure 2.3 Java icon 4](#_Toc168263003)

[Figure 2.4 Swing Components 5](#_Toc168263005)

[Figure 2.5 SQL example 5](#_Toc168263007)

[Figure 2.5 HTML logo 6](#_Toc168263009)

[Figure 2.6 CSS logo 6](#_Toc168263011)

[Figure 2.7 MVC architecture 7](#_Toc168263013)

[Figure 3.1 User Functionality Diagram 8](#_Toc168263017)

[Figure 3.2 Shipper Functionality Diagram 9](#_Toc168263019)

[Figure 3.3 Admin Functionality Diagram 10](#_Toc168263021)

[Figure 3.4 Use – Case Diagram 11](#_Toc168263024)

[Figure 3.5 User Use – Case Diagram 12](#_Toc168263026)

[Figure 3.6 Shipper Use – Case Diagram 12](#_Toc168263028)

[Figure 3.7 Admin Use – Case Diagram 13](#_Toc168263030)

[Figure 3.8 Database Relationship Diagram 13](#_Toc168263032)

[Figure 4.1 Login interface 14](#_Toc168263034)

[Figure 4.2 User Dashboard interface 14](#_Toc168263036)

[Figure 4.3 Purchase interface 15](#_Toc168263037)

[Figure 4.4 Purchase Details interface 15](#_Toc168263038)

[Figure 4.5 My Account interface 16](#_Toc168263039)

[Figure 4.6 Sign Up interface 16](#_Toc168263040)

[Figure 4.7 Forgot Password interface 17](#_Toc168263041)

[Figure 4.8 Send OTP to Email 17](#_Toc168263042)

[Figure 4.9 Shipper Dashboard interface 18](#_Toc168263044)

[Figure 4.10 Delivery interface 18](#_Toc168263045)

[Figure 4.11 My Deliveries interface 19](#_Toc168263046)

[Figure 4.12 My Account interface 19](#_Toc168263047)

[Figure 4.13 Admin Dashboard interface 20](#_Toc168263049)

[Figure 4.14 Manage Categories interface 20](#_Toc168263050)

[Figure 4.15 Manage Products interface 21](#_Toc168263051)

[Figure 4.16 Manage Users interface 21](#_Toc168263052)

[Figure 4.17 Add Shipper interface 22](#_Toc168263053)

[Figure 4.18 Manage Shippers interface 22](#_Toc168263054)

[Figure 4.19 Select Shipper interface 23](#_Toc168263055)

[Figure 4. 20 Transaction interface 23](#_Toc168263056)

[Figure 4.21 About “Chat” 24](#_Toc168263057)

[Figure 4.22 Demo Chat 24](#_Toc168263058)

# CHAPTER 1. GENERAL INTRODUCTION

## 1.1 Overview

### 1.1.1 Reason for choosing the topic

**Technology and the internet have changed the way we shop.** Nowadays, consumers can easily search for and shop for anything they want with just a few clicks. Brands and retailers are also leveraging the power of technology to expand their reach and provide a better shopping experience for their customers.

**The BotShop application was born** to meet the increasing shopping needs of consumers. **BotShop** is an online shopping management application designed to bring users the best shopping experience and provide a comprehensive sales management solution for individual sellers. At the same time, the project also aims to build a comprehensive ecosystem, connecting the three main elements: sellers, buyers, and delivery people, to bring the best selling and shopping experience for everyone.

### 1.1.2 Problems to be solved

**Inconsistent User Experience**: Many current e-commerce applications fail to provide a seamless and convenient experience for users, from product search and order placement to order tracking and delivery.

**Challenges in Sales Management**: Retailers, especially individual sellers, face difficulties in effectively managing orders, inventory, and sales-related activities in a centralized manner.

**Fragmented Services**: Currently, the online shopping process is often not integrated between stages such as selling, purchasing, and delivery, leading to inefficiencies and difficulties for both sellers and buyers.

### 1.1.3 To-do list

**User Interface (UI/UX) Design**: Ensure a user-friendly, intuitive interface that optimizes the user experience from product search to order placement and payment.

**Sales Management System**: Develop features for order management, inventory control, and revenue tracking in a visual and optimized manner.

**Ecosystem Integration**: Connect sellers, buyers, and delivery personnel to create a synchronized and cohesive ecosystem, enabling a seamless and efficient shopping experience.

**Security and Privacy**: Implement robust security measures, such as password encryption and email verification, to protect users' personal information and transactions. This will enhance user confidence and trust in the application.

## 1.2 Project tasks

### 1.2.1 Approach to project execution

**Requirement Analysis:** Begin by analyzing the requirements of the online shopping application, including necessary functionalities, user interactions with the application, and other factors such as security and performance.

**User Interface Design:** Sketch the user interface based on the requirement analysis, using tools such as hand-drawing or design software. Identify the necessary interface components and how they interact to ensure the best user experience.

**Interface Implementation:** After completing the design, implement the interface using Java Swing. Utilize Swing's graphical components to create windows, buttons, input fields, and other necessary elements for the application.

**Application Logic Programming:** Once the interface is implemented, proceed to program the application logic to handle functions such as product management, order processing, and payment handling. Ensure that the logic operates correctly and efficiently.

**Testing and Debugging:** Finally, conduct application testing to detect and fix any potential bugs. Ensure that the application runs smoothly and meets the previously analyzed requirements.

### 1.2.2 Result

The outcome is an online shopping application developed in Java, fully meeting the analyzed requirements. This application not only offers a user-friendly and intuitive interface but also optimizes the user experience, making online shopping simpler and more efficient.

## 1.3 Project report structure

 Chapter 1. General Introduction: Provides an overview of the topic, methodology, and objectives of the project.

 Chapter 2. General Research: Presents an overview of the technologies and tools used in developing the application.

 Chapter 3. System Analysis and Design: Analyzes specific requirements and designs the system to ensure performance and scalability.

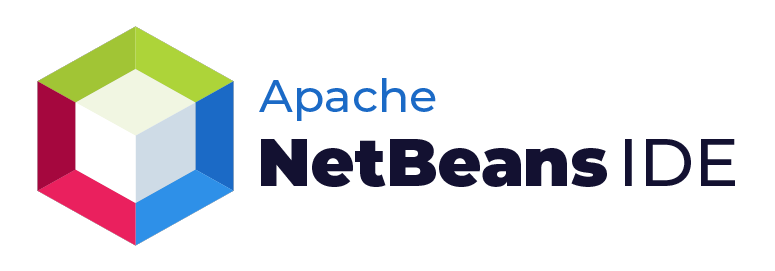
 Chapter 4. Interface Design: Creates a user-friendly and engaging user interface.

 Chapter 5. Conclusion and Future Directions: Summarizes the project and proposes future development directions.

# CHAPTER 2: GENERAL RESEARCH

## 2.1 Tools used

### 2.1.1 NetBeans IDE



#### Figure 2.1 NetBeans IDE

NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE (including JavaFX), Java ME, web, EJB, and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactorings, version control (supporting CVS, Subversion, Git, Mercurial, and Clearcase).

### ****2.1.2 SQL Server Management Studio****



#### **Figure 2.2** **SQL Server Management Studio**

**SQL Server Management Studio (SSMS) is an essential tool for effectively managing SQL Server databases. Developed by Microsoft, SSMS provides a comprehensive, integrated environment where database administrators, developers, and other users can efficiently work with their SQL Server instances.**

It allows DBAs and database developers to **configure, manage**, and **administer** all components within SQL Server. Its main functionality is to create databases and tables, execute SQL queries for inserting, updating, and deleting data, creating and managing stored procedures, triggers, views, and cursors. It also enables us to set privileges (securities) on databases and their objects.

## 2.2 Technologies used

### 2.2.1 Java programming language



#### Figure 2.3 Java icon

Java is a versatile programming language that is widely used for developing a variety of applications, including desktop, web, and mobile applications. It provides a robust and secure environment for application development and is known for its platform independence. This means that Java programs can run on different operating systems without requiring major modifications.

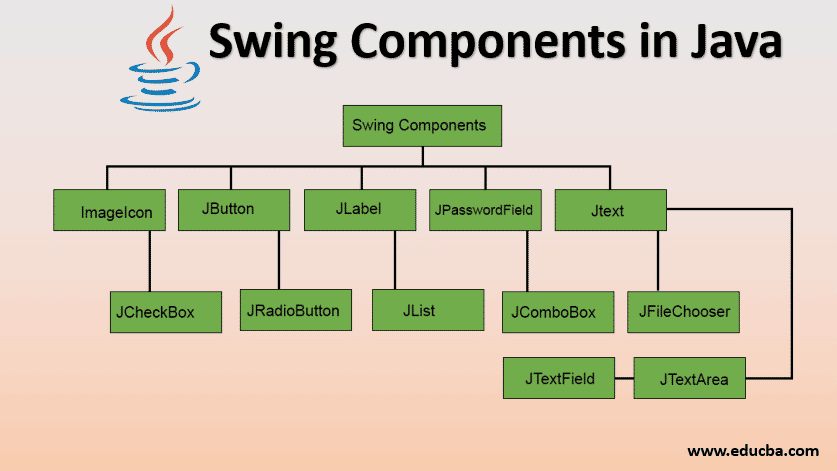
Java follows the principles of object-oriented programming (OOP), which is a popular programming paradigm that focuses on organizing code into reusable objects that interact with each other. It offers a wide range of libraries, frameworks, and tools that facilitate the development process and enable developers to create efficient and scalable applications.

By embracing OOP principles, Java promotes code reusability, modularity, and abstraction, making it a powerful language for building complex and robust software systems. Its wide range of libraries, frameworks, and tools, combined with its platform independence, make Java a popular choice for developing enterprise-level applications and large-scale software projects.

### 2.2.2 Swing (Java)

Swing is a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) [widget toolkit](https://en.wikipedia.org/wiki/Widget_toolkit) for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)).[[1]](https://en.wikipedia.org/wiki/Swing_(Java)#cite_note-1) It is part of [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation)'s [Java Foundation Classes](https://en.wikipedia.org/wiki/Java_Foundation_Classes) (JFC) – an [API](https://en.wikipedia.org/wiki/Application_programming_interface) for providing a [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI [components](https://en.wikipedia.org/wiki/Software_component) than the earlier [Abstract Window Toolkit (AWT)](https://en.wikipedia.org/wiki/Abstract_Window_Toolkit). Swing provides a [look and feel](https://en.wikipedia.org/wiki/Look_and_feel) that emulates the look and feel of several platforms, and also supports a [pluggable look and feel](https://en.wikipedia.org/wiki/Pluggable_look_and_feel) that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.[[2]](https://en.wikipedia.org/wiki/Swing_(Java)#cite_note-2)

Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent.

#### Figure 2.4 Swing Components

### 2.2.2 Structured Query Language (SQL)

#### Figure 2.5 SQL example

Structured Query Language (SQL) is a programming language designed for storing and processing information in relational databases. Relational databases store information in tables with rows and columns representing data attributes, and various relationships between data values. You can use SQL statements to store, update, delete, search, and retrieve information from the database. SQL can also be used to maintain and optimize database performance, as well as connect to SQL Server to perform management tasks and interact with the database from popular applications or database management tools

### 2.2.3 ****Hyper Text Markup Language**** (HTML)



#### Figure 2.5 HTML logo

HTML, short for Hyper Text Markup Language, is a markup language used to create and format web pages. HTML is a simple language used to describe the structure of a web page using tags and elements. Each HTML tag defines a specific part of the web page, such as headings, paragraphs, images, links, tables, forms, and other interactive elements.

HTML does not contain information about how the web page will be displayed or what it will look like; it only describes the logical structure of the page. This means that HTML does not specify font styles, colors, or layouts of the elements on the web page. Instead, the properties and values of HTML tags can be adjusted and customized using CSS (Cascading Style Sheets) to create an aesthetically pleasing and readable web interface.

### 2.2.4 Cascading Style Sheets (CSS)



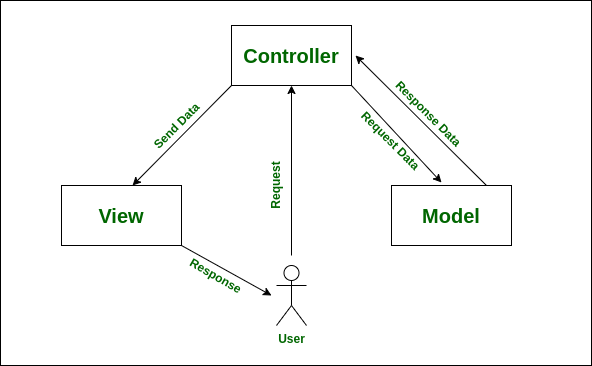
#### Figure 2.6 CSS logo

CSS stands for Cascading style sheets. It describes to the user how to display HTML elements on the screen in a proper format. CSS is the language that is used to style HTML documents. In simple words, cascading style sheets are a language used to simplify the process of making a webpage.

CSS is used to handle some parts of the webpage. With the help of CSS, we can control the color of text and style of fonts, and we can control the spacing between the paragraph and many more things. CSS is easy to understand but provides strong control on the Html documents.CSS is combined with HTML.

## 2.3 Application Architecture

The full form of MVC is Model-View-Controller. It is a software architectural pattern commonly used for developing user interfaces that divide the application into three interconnected components. This separation of concerns makes it easier to manage and scale complex applications.



#### Figure 2.7 MVC architecture

Components of MVC:

* Model: The Model component represents the application’s data and business logic. It is responsible for managing the data, including retrieving, inserting, updating, and deleting data from the database or any other data source.
* View: The View component is responsible for presenting the data from the Model to the user. It is the user interface of the application, consisting of all the UI components such as text boxes, dropdowns, buttons, etc. The View displays data to the user and sends user actions (e.g., button clicks) to the Controller.
* Controller: The Controller acts as an intermediary between the Model and the View. It receives input from the user via the View, processes the input (often with the help of the Model), and returns the output to be displayed in the View. The Controller is responsible for controlling the flow of the application and managing user interactions.

# CHAPTER 3. SYSTEM ANALYSIS AND DESIGN

## 3.1 System analysis

### 3.1.1 User functionality

#### Figure 3.1 User Functionality Diagram

* Sign Up: Users create a new account by providing personal information (name, email, password, phone number).
* Login: Users log in using the registered email and password.
* Forgot Password: Users request a password reset, receive an OTP code via email, and then enter the OTP code to set a new password.
* Account Management: Users view and update personal information, change passwords, or delete their accounts.
* Purchase: Users browse products, add them to the shopping cart, and proceed to checkout.
* Purchase Details: Users view purchase history, track order status, cancel pending or received orders (within 30 days), and view order details.
* Chat: Participate in a common chat system between delivery personnel, sellers, and buyers to exchange information, provide customer support, and resolve order-related issues.
* Statistics: Users can view statistics on the number of product categories, items purchased, and total spending.

### 3.1.2 Shipper Functionality

#### Figure 3.2 Shipper Functionality Diagram

* Login: Shipper log in using the accounts provided by the Admin.
* Delivery: View and update delivery status for orders.
* View Delivered Orders: Review the history of delivered orders.
* Account Management: View and update personal information.
* Chat: Participate in a common chat system between delivery personnel, sellers, and buyers to exchange information, provide customer support, and resolve order-related issues.
* Statistics: Analyze delivered orders.

### 3.1.3 Admin Functionality

#### Figure 3.3 Admin Functionality Diagram

* Login
* Product Management: Includes adding, editing, and deleting products from the product list, as well as updating detailed information such as name, description, price, and quantity.
* Category Management: Involves managing product categories and their attributes, and updating product information including name, descriptions, and prices.
* User Management: View the list of users and their personal information (with editing rights).
* Shipper Management: View the list of delivery personnel and their information (with editing rights)
* Selecting Shipper: Selecting delivery personnel.
* Transaction Tracking: Aggregates information on completed transactions, including orders, payments, and shipments.
* Chat: Participate in a common chat system between delivery personnel, sellers, and buyers to exchange information, provide customer support, and resolve order-related issues.
* Overall Statistics: Summarizes information about the entire system, such as the number of product types, quantity of products, and other relevant data.

## 3.2 System Design

### 3.2.1 Overview Use – Case Diagram

#### Figure 3.4 Use – Case Diagram

### 3.2.2 User Use-Case Diagram

#### Figure 3.5 User Use – Case Diagram

### 3.2.3 Shipper Use-Case Diagram

#### Figure 3.6 Shipper Use – Case Diagram

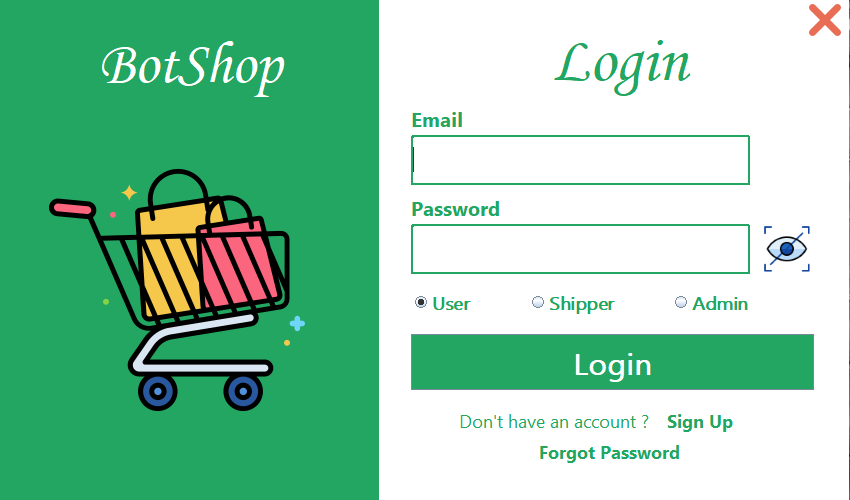
### 3.2.4 Admin Use-Case Diagram

#### Figure 3.7 Admin Use – Case Diagram

### 3.2.5 Database Relationship Diagram

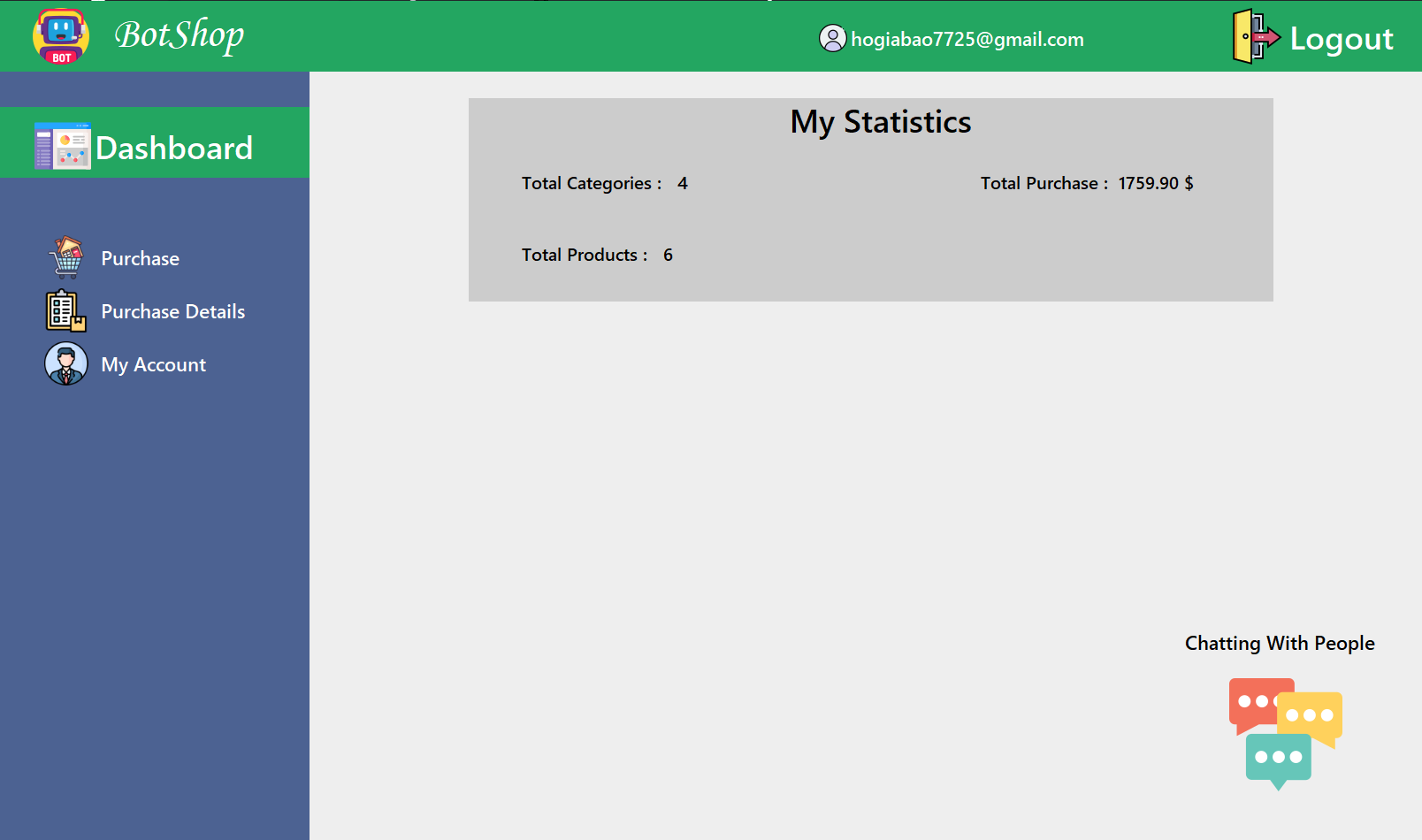
#### Figure 3.8 Database Relationship Diagram

# Chapter 4. Interface Design

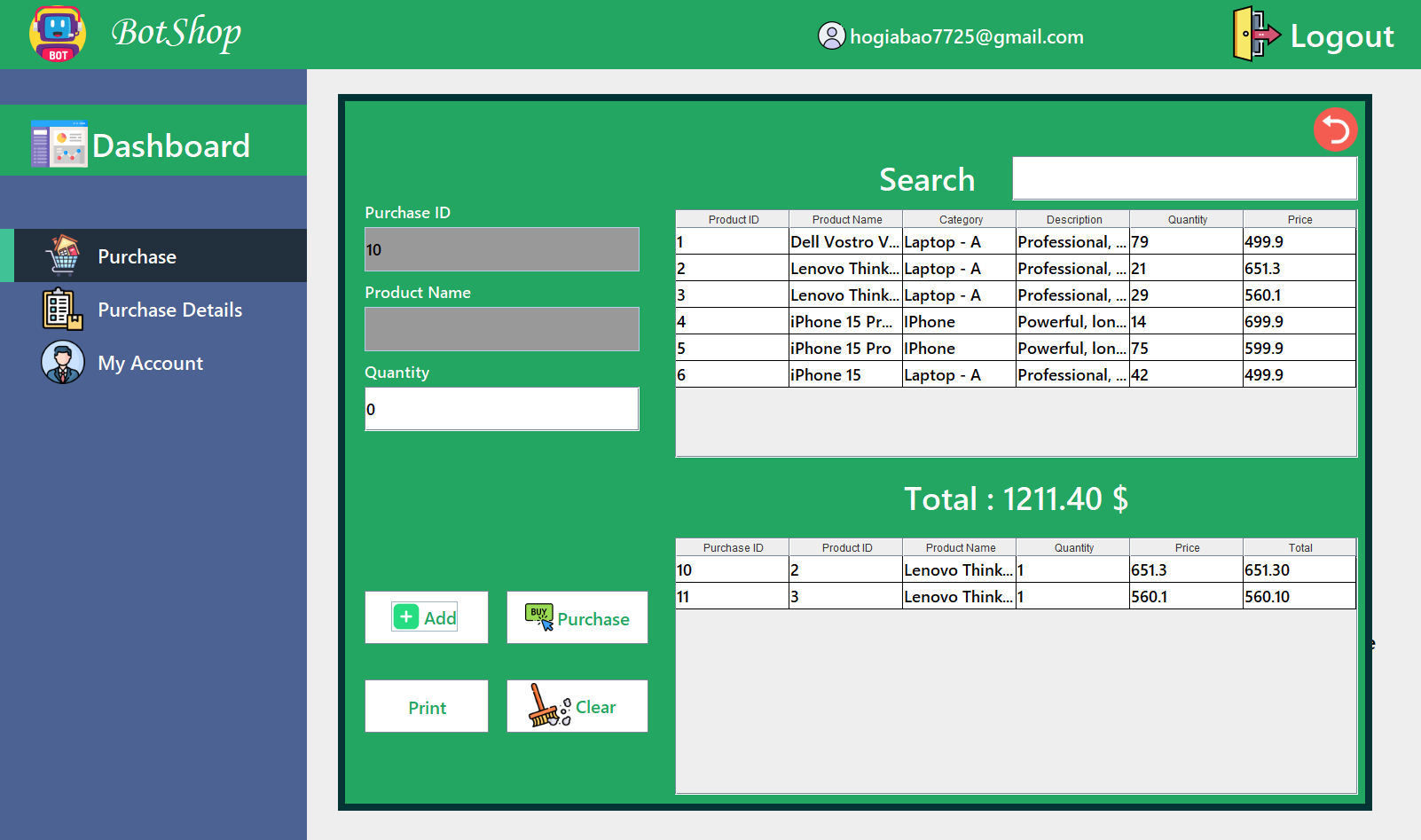
To run the application, first execute the **Login.java** file in the View\_User package. The following is the interface that will appear after running the file.

#### Figure 4.1 Login interface

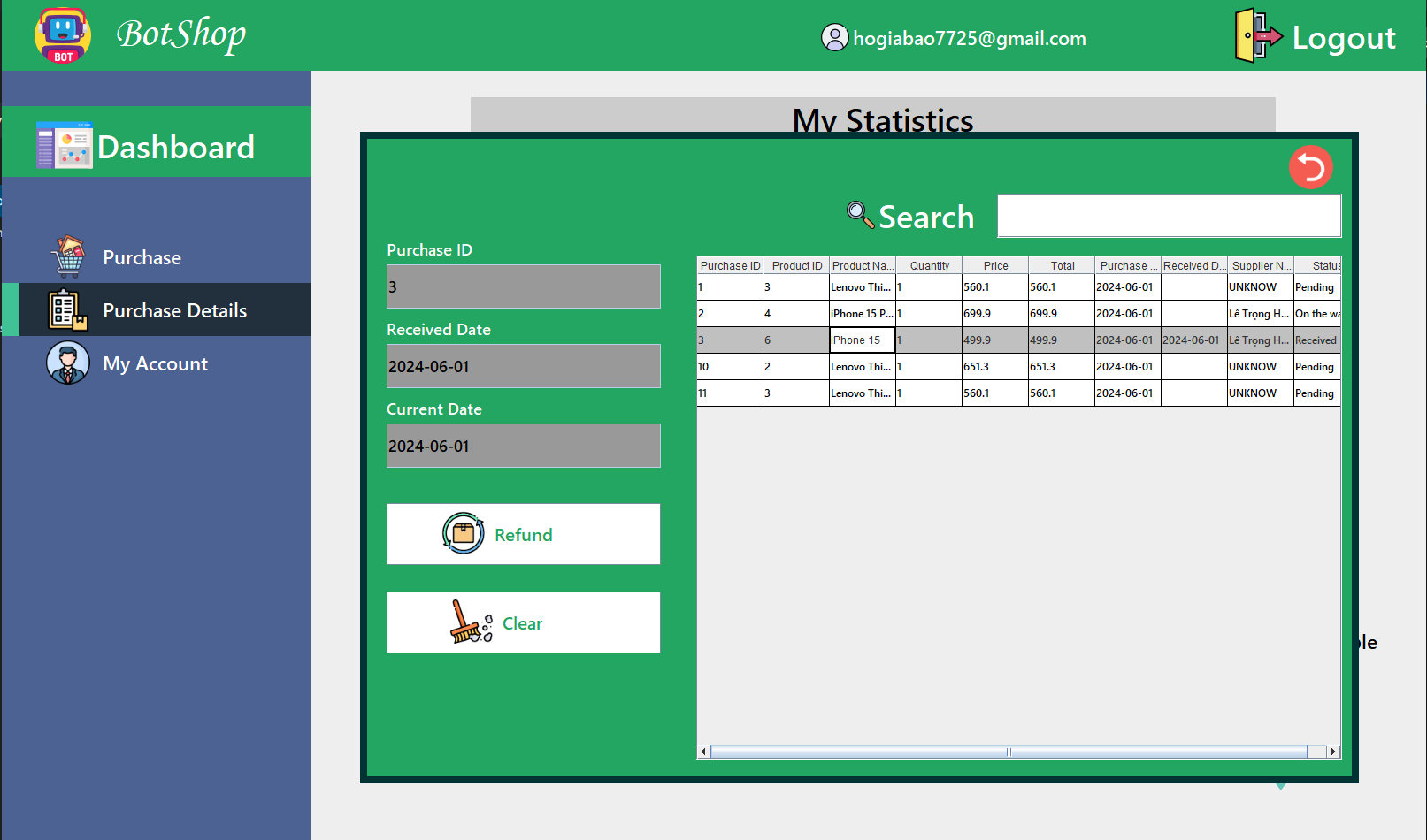
## 4.1 User interface

🡪 User interface after **successful user login**.

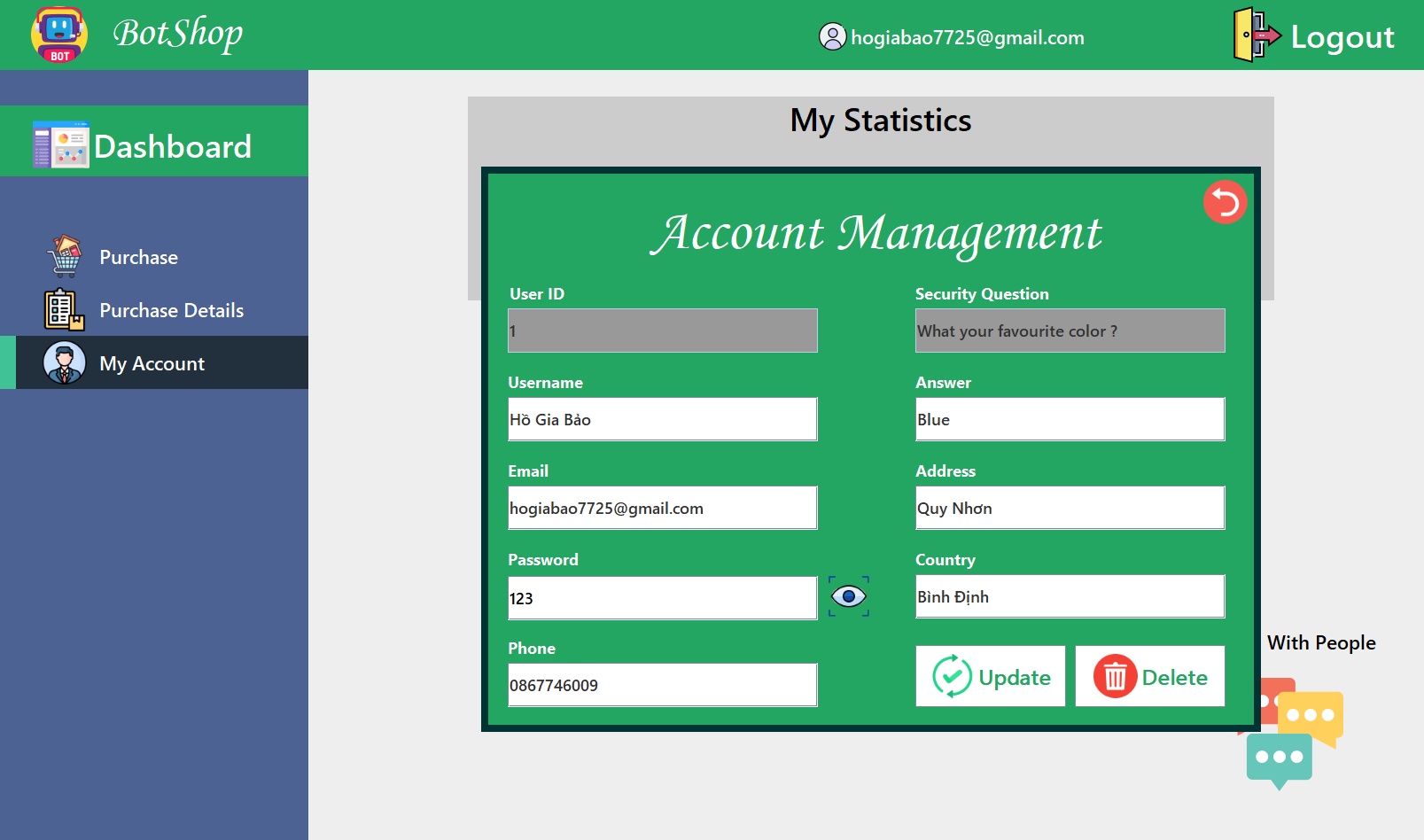
#### Figure 4.2 User Dashboard interface

* “Purchase” interface: On this interface, users can purchase the products they need by clicking the "Add" button to add them to their cart and then clicking the "Purchase" button to checkout.

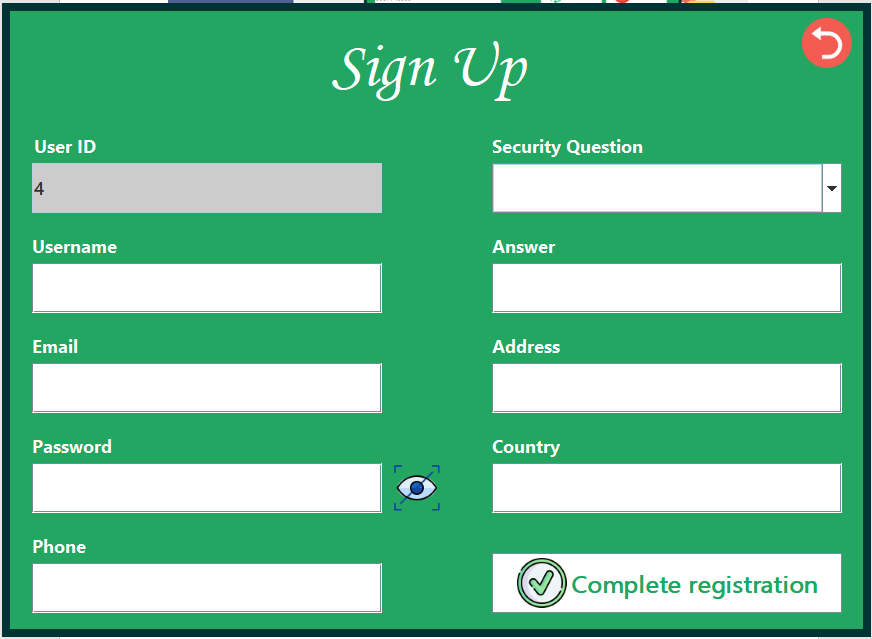
#### Figure 4.3 Purchase interface

* “Purchase Details” interface: Users can track the orders they have purchased, and can return items if they have been received for no more than 30 days.

#### Figure 4.4 Purchase Details interface

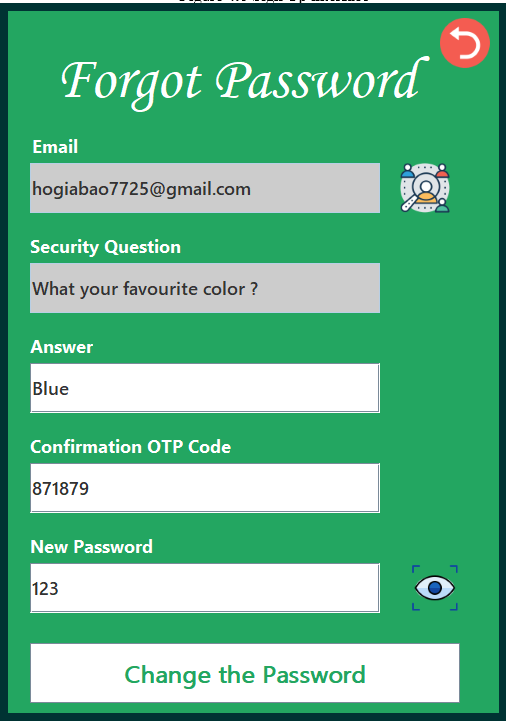
* “My Account” interface: This interface will help users to modify their information such as name, email and so on.

#### Figure 4.5 My Account interface

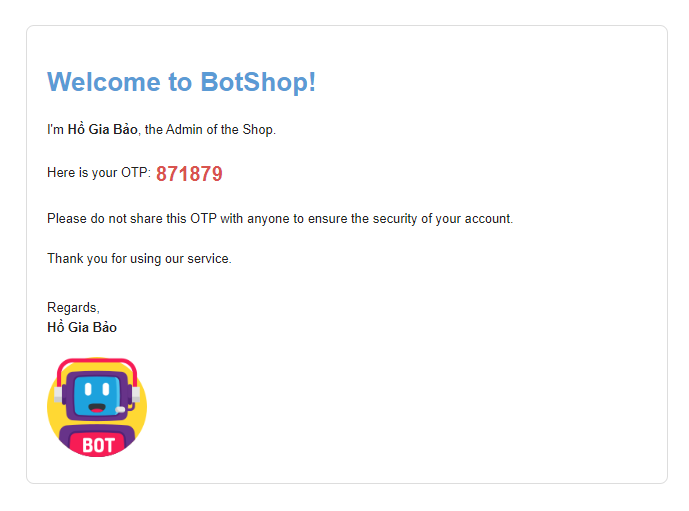
* “Sign up” interface: If users do not have an account, they can sign up by clicking on the 'Sign Up' label on the 'Login' interface and filling in all the necessary information to complete the registration process.

#### Figure 4.6 Sign Up interface

* “Forgot Password” interface: If users forget their password, they can click on the 'Forgot Password' label on the ‘Login’ interface and follow these steps:



* Step 1: Enter their Email and click on the search icon.
* Step 2: If the email is correct, an OTP code will be sent to their Email.
* Step 3: Enter the OTP code, answer the 'Security Question', and set a new password.
* Step 4: Click on the 'Change Password' button. If the password is correct, it will be changed.

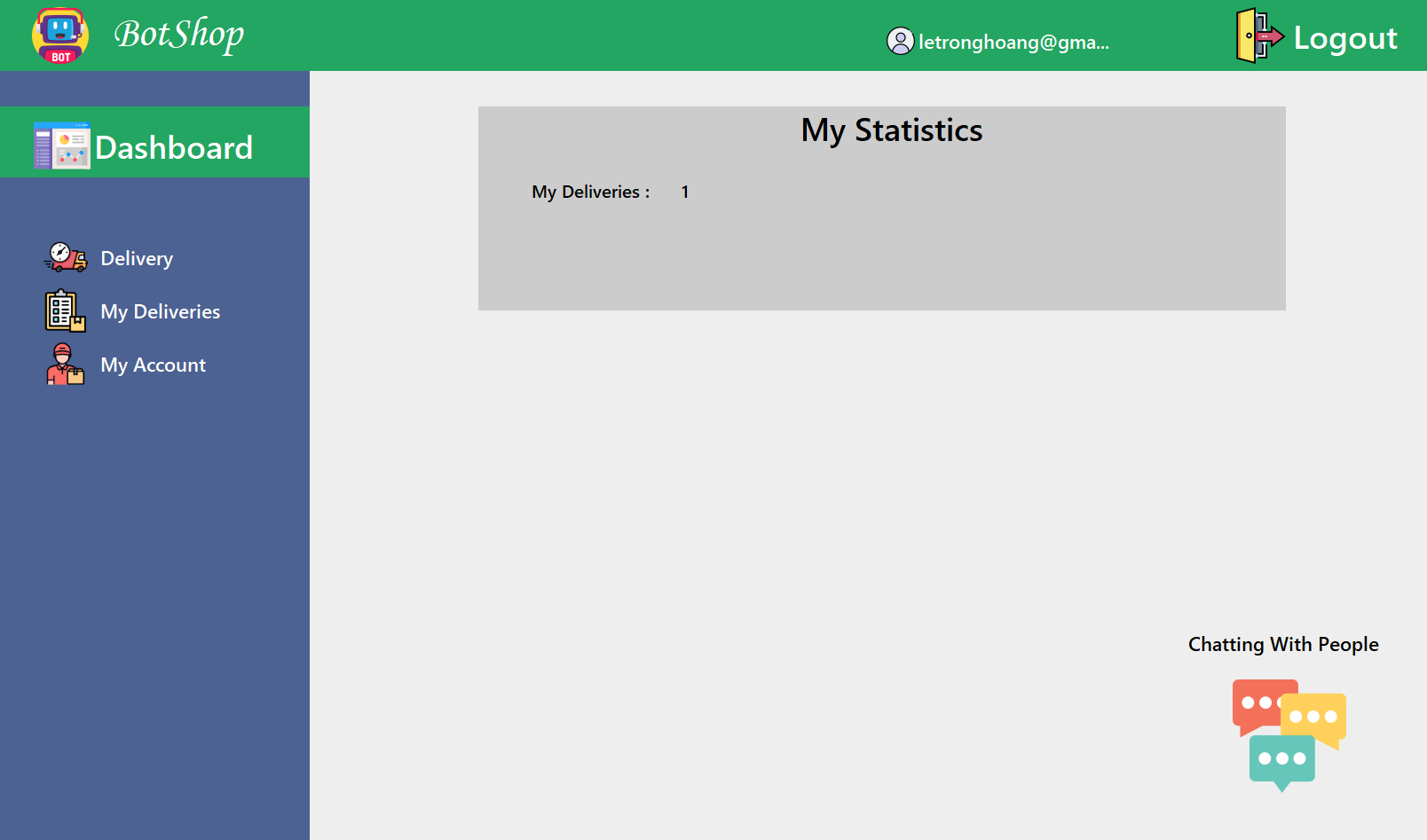


#### 

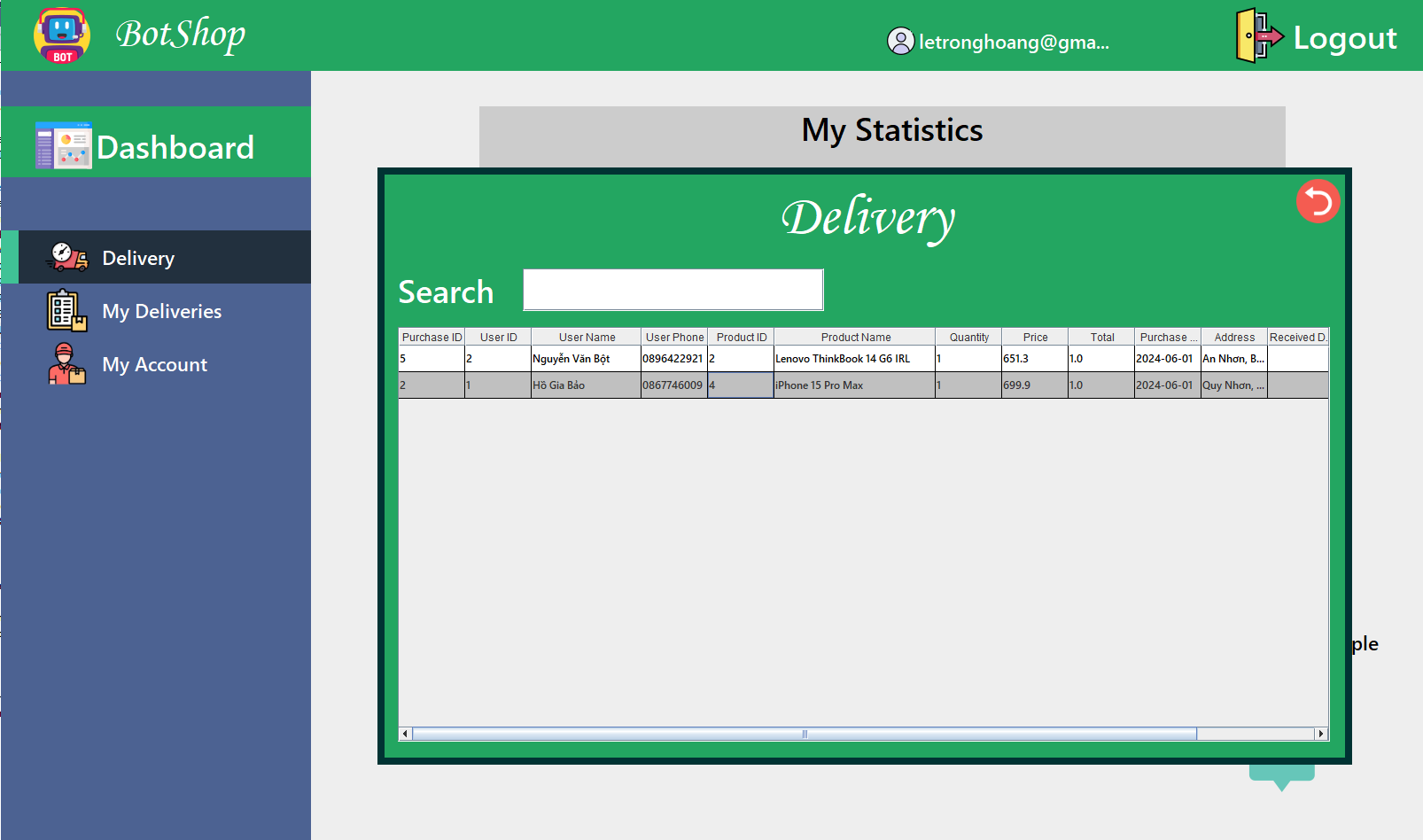
#### Figure 4.7 Forgot Password interface

#### Figure 4.8 Send OTP to Email

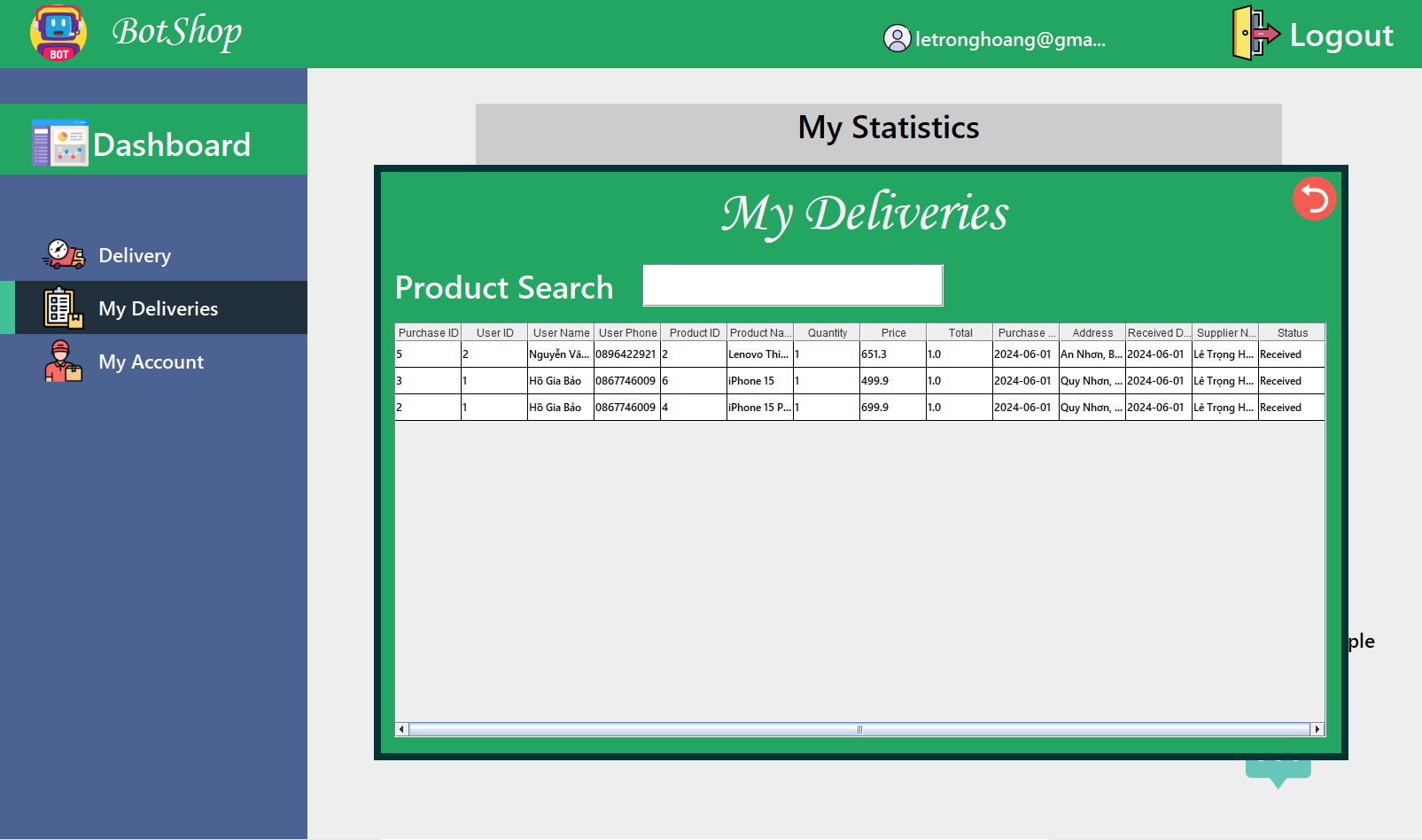
## 4.2 Shipper interface

🡪 Shipper interface after **successful shipper login**.

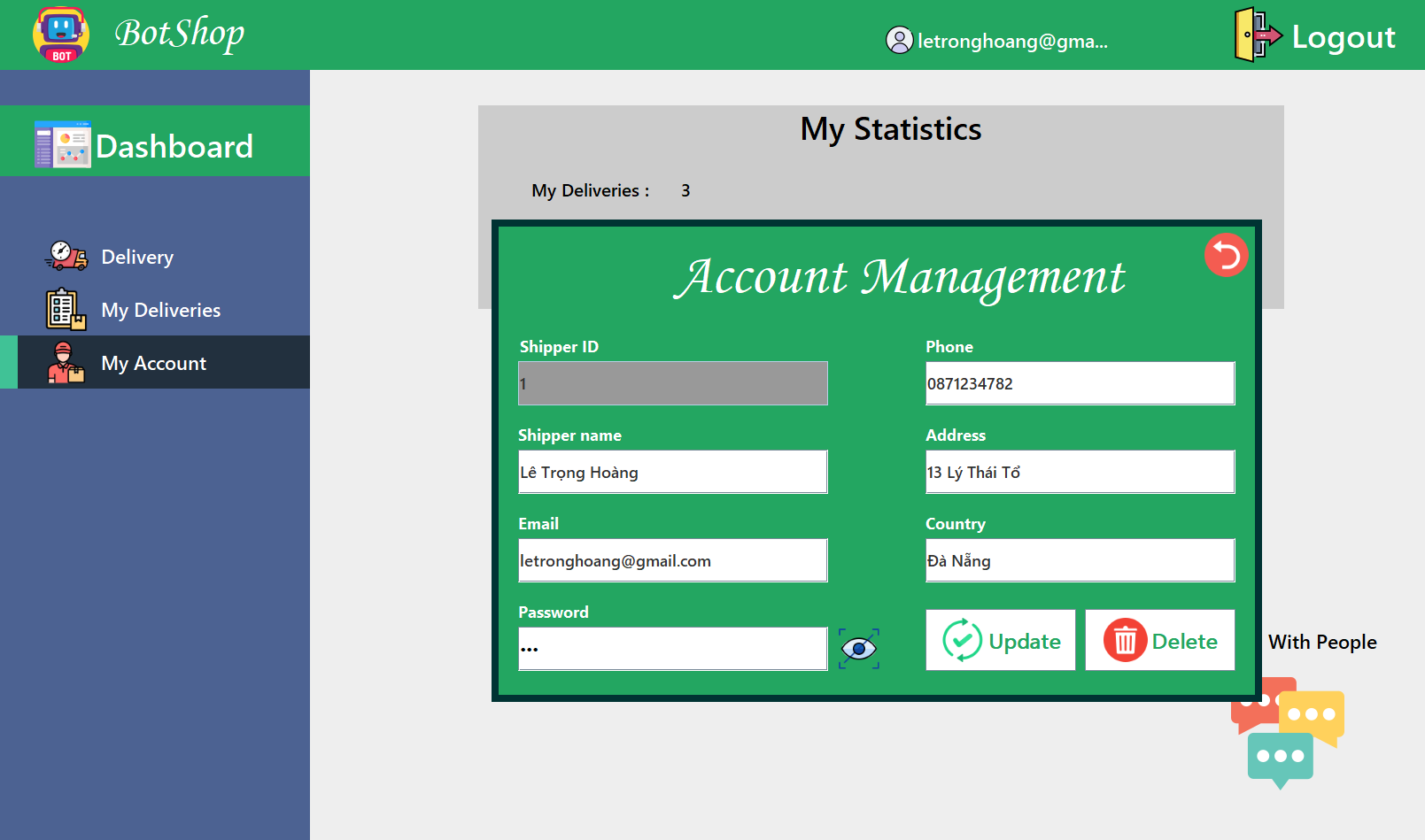
#### Figure 4.9 Shipper Dashboard interface

* “Delivery” interface: In this interface, the shipper will see the orders they need to deliver. By clicking on the respective rows in the table corresponding to the delivered orders, they can confirm that the orders have been successfully delivered.

#### Figure 4.10 Delivery interface

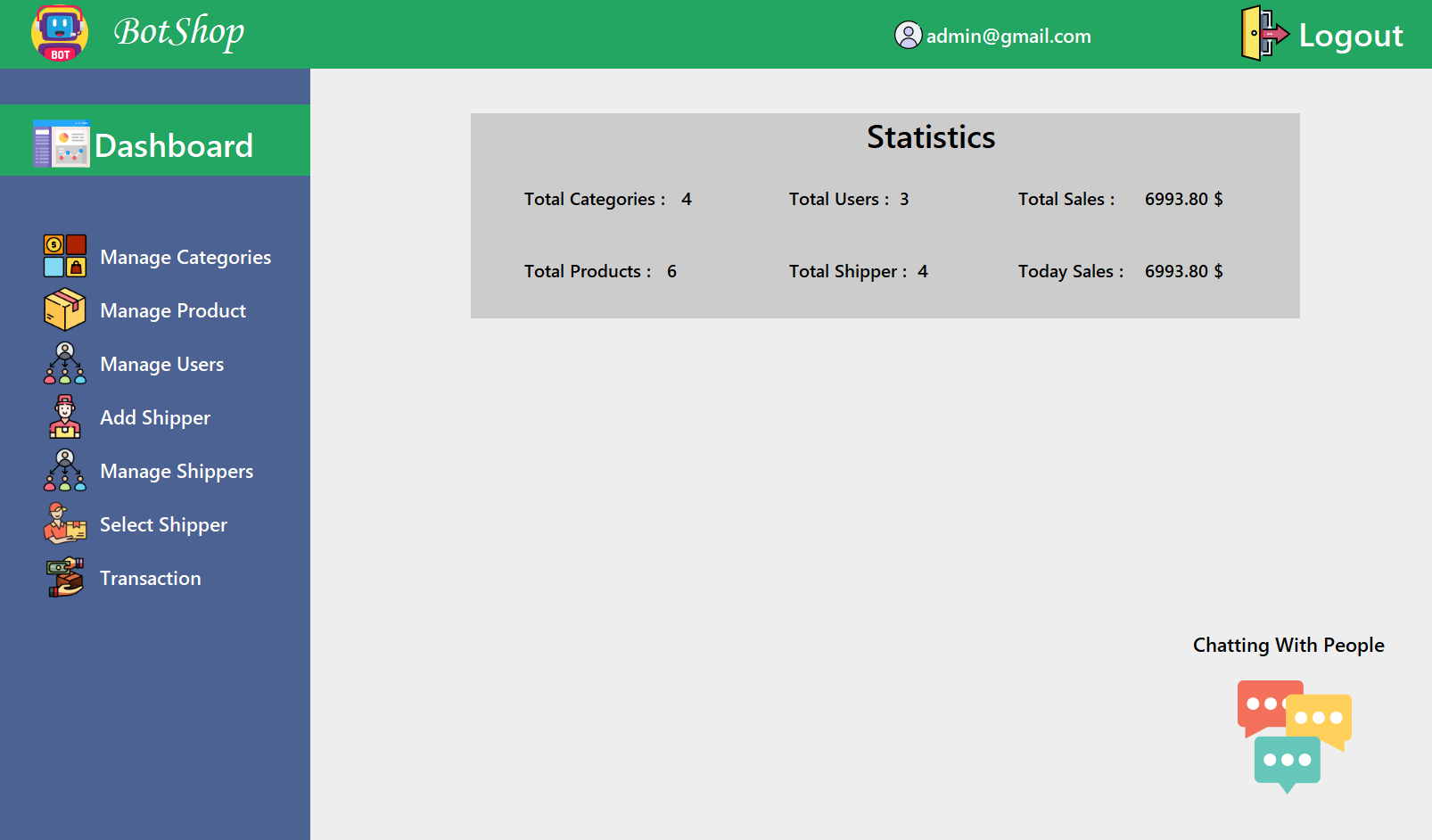
* “My Deliveries”: This interface will inform the shipper about the orders they have successfully delivered.

#### Figure 4.11 My Deliveries interface

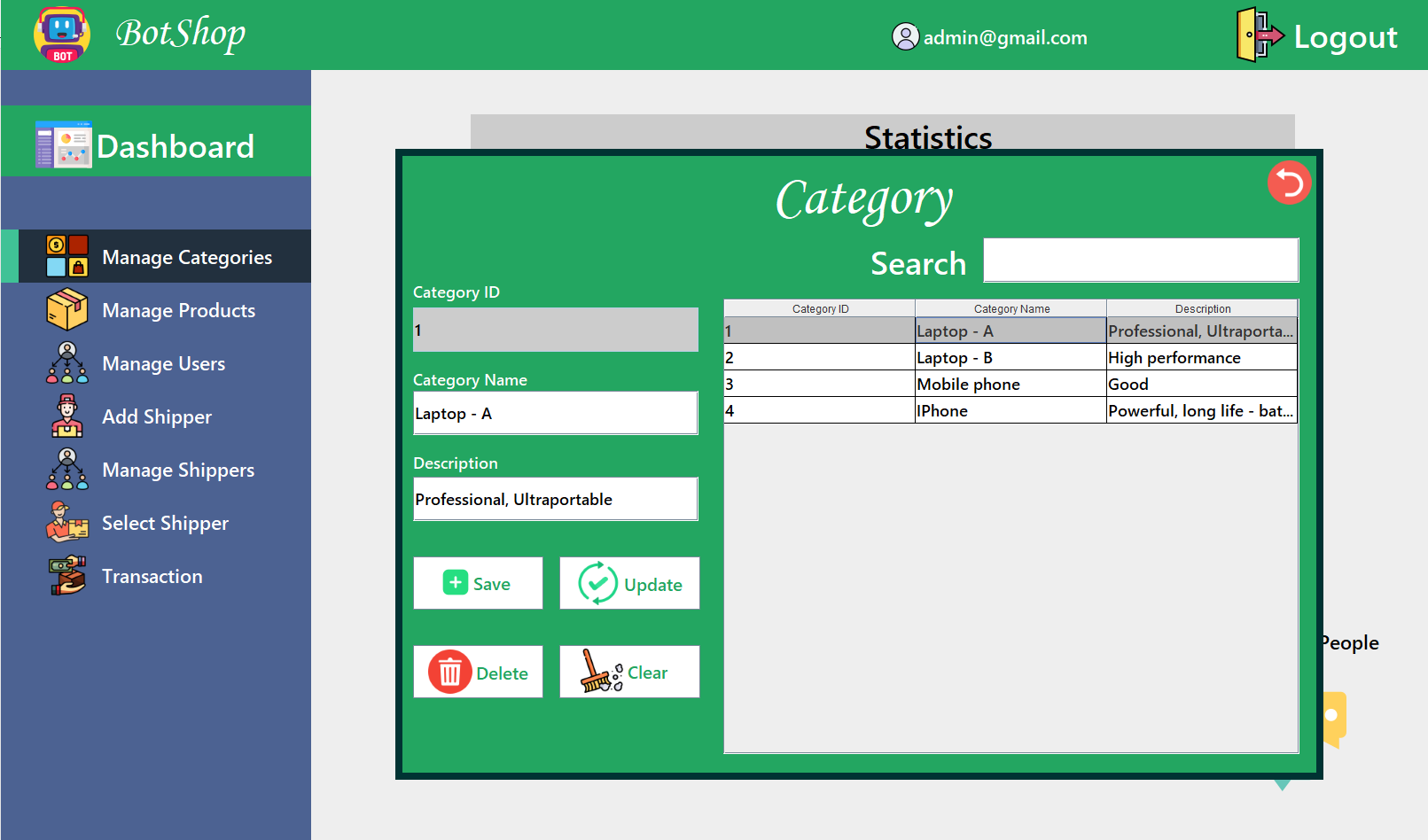
* “My Account” interface: The shipper can update their information if they want.

#### Figure 4.12 My Account interface

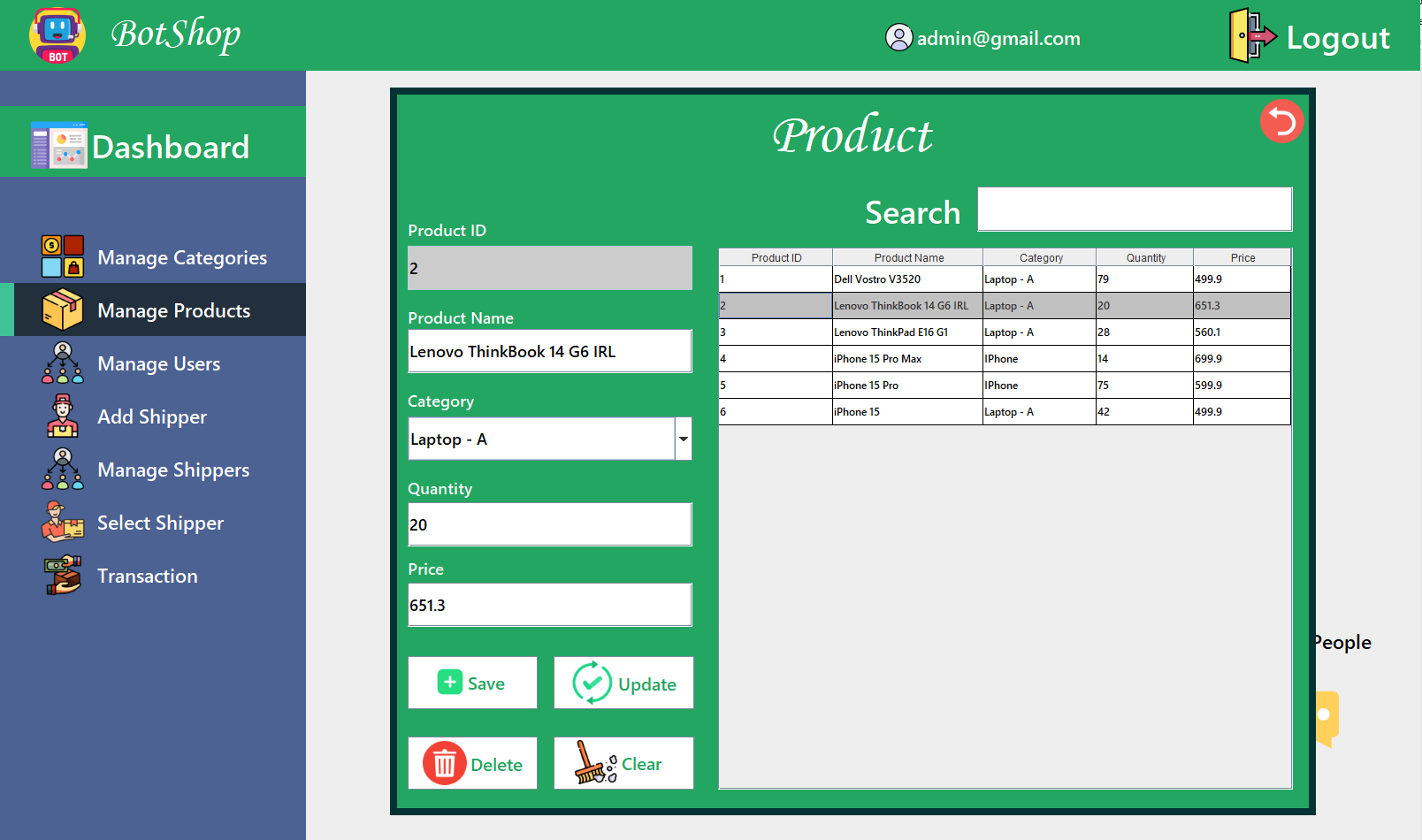
## 4.3 Admin interface

🡪 Admin interface after **successful Admin login**.

#### Figure 4.13 Admin Dashboard interface

* “Manage Categories” interface: Admin can add, edit, and delete categories.

#### Figure 4.14 Manage Categories interface

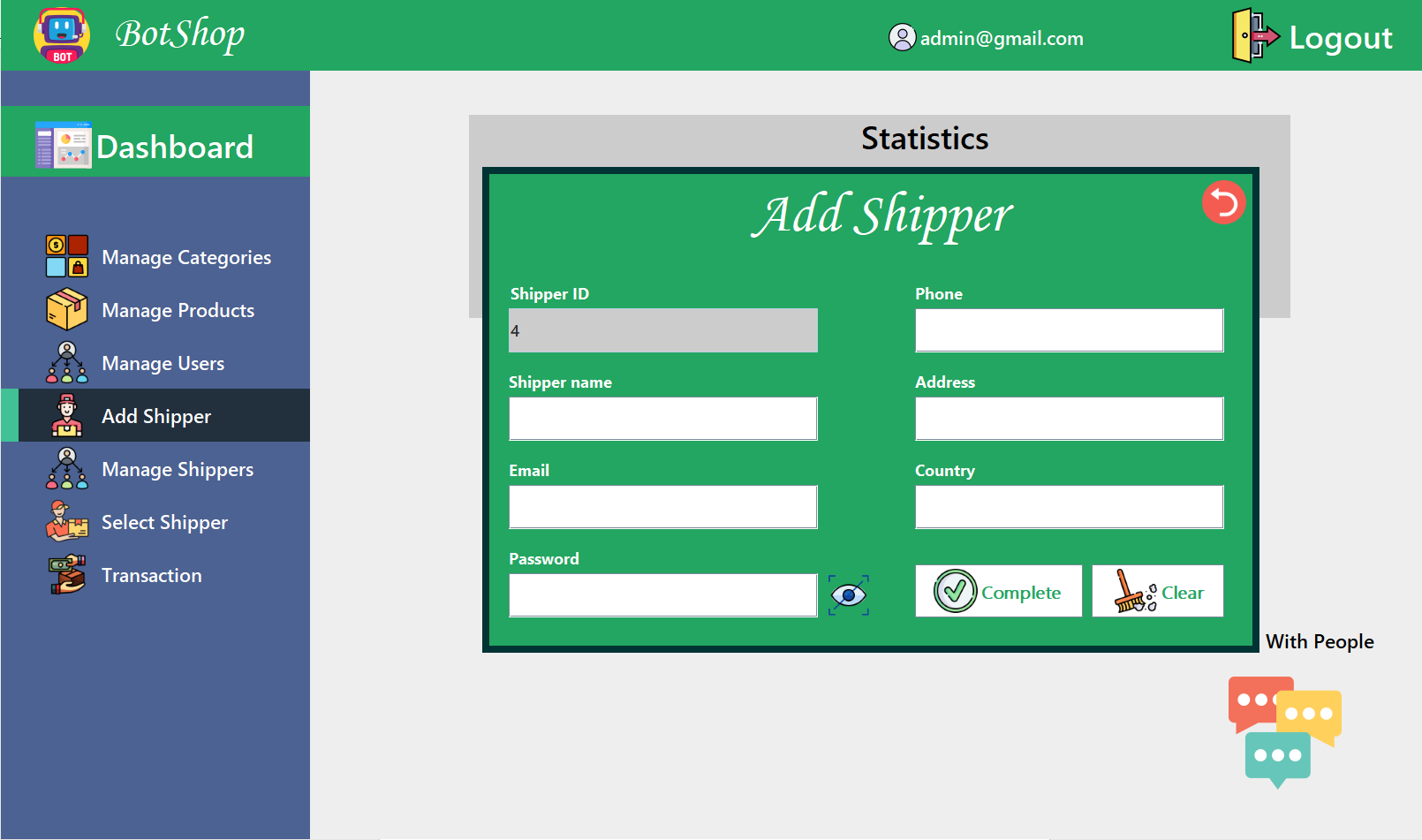
* “Manage Products” interface: Admin can add, edit, and delete products.

#### Figure 4.15 Manage Products interface

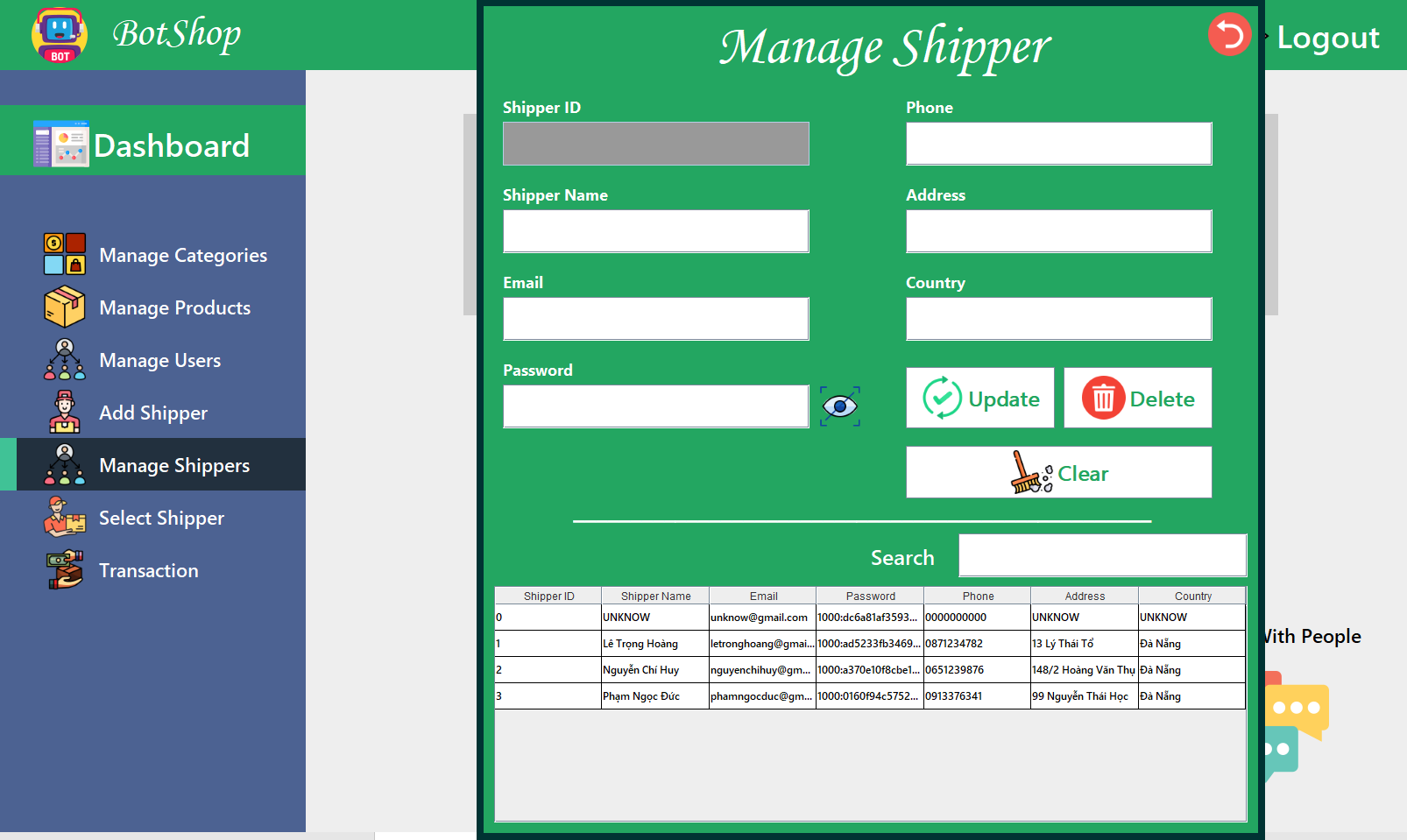
* “Manage Users”: Admin will manage their users by viewing, editing their information, or deleting users.

#### 

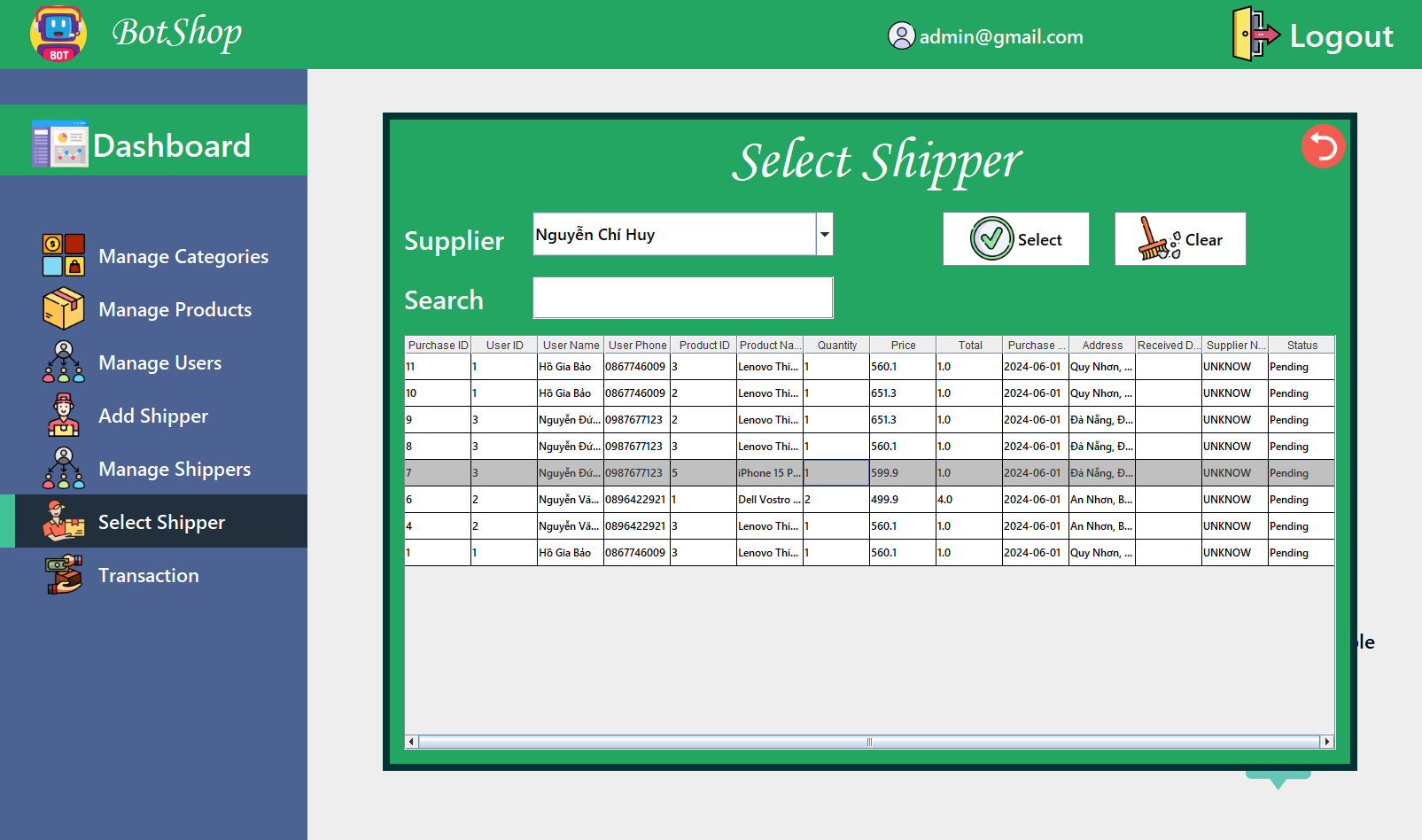
#### Figure 4.16 Manage Users interface

* “Add Shipper”: Add a or more shippers to your shop.

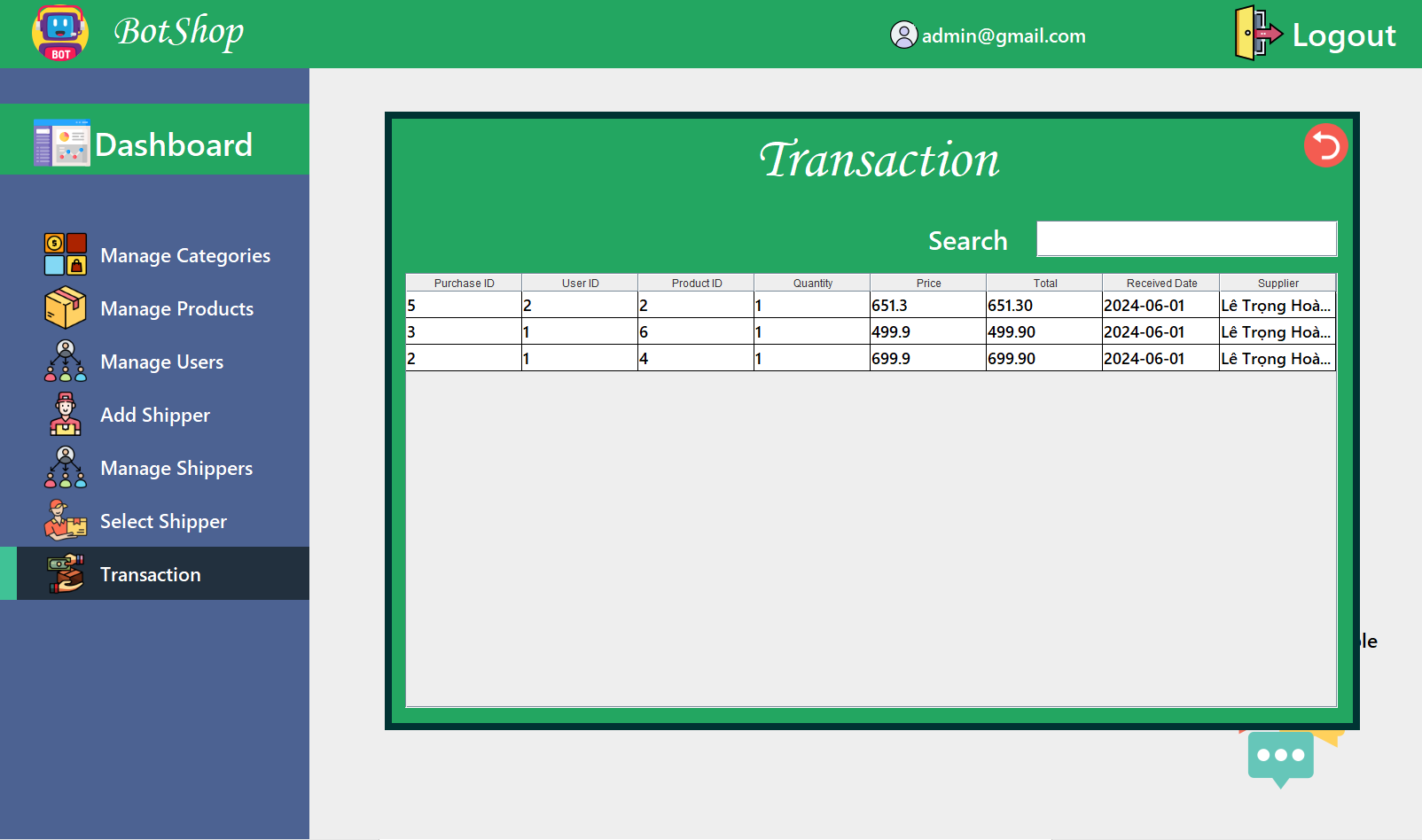
#### Figure 4.17 Add Shipper interface

* “Manage Shippers”: Admin will manage their Shippers by viewing, editing their information, or deleting them.

#### Figure 4.18 Manage Shippers interface

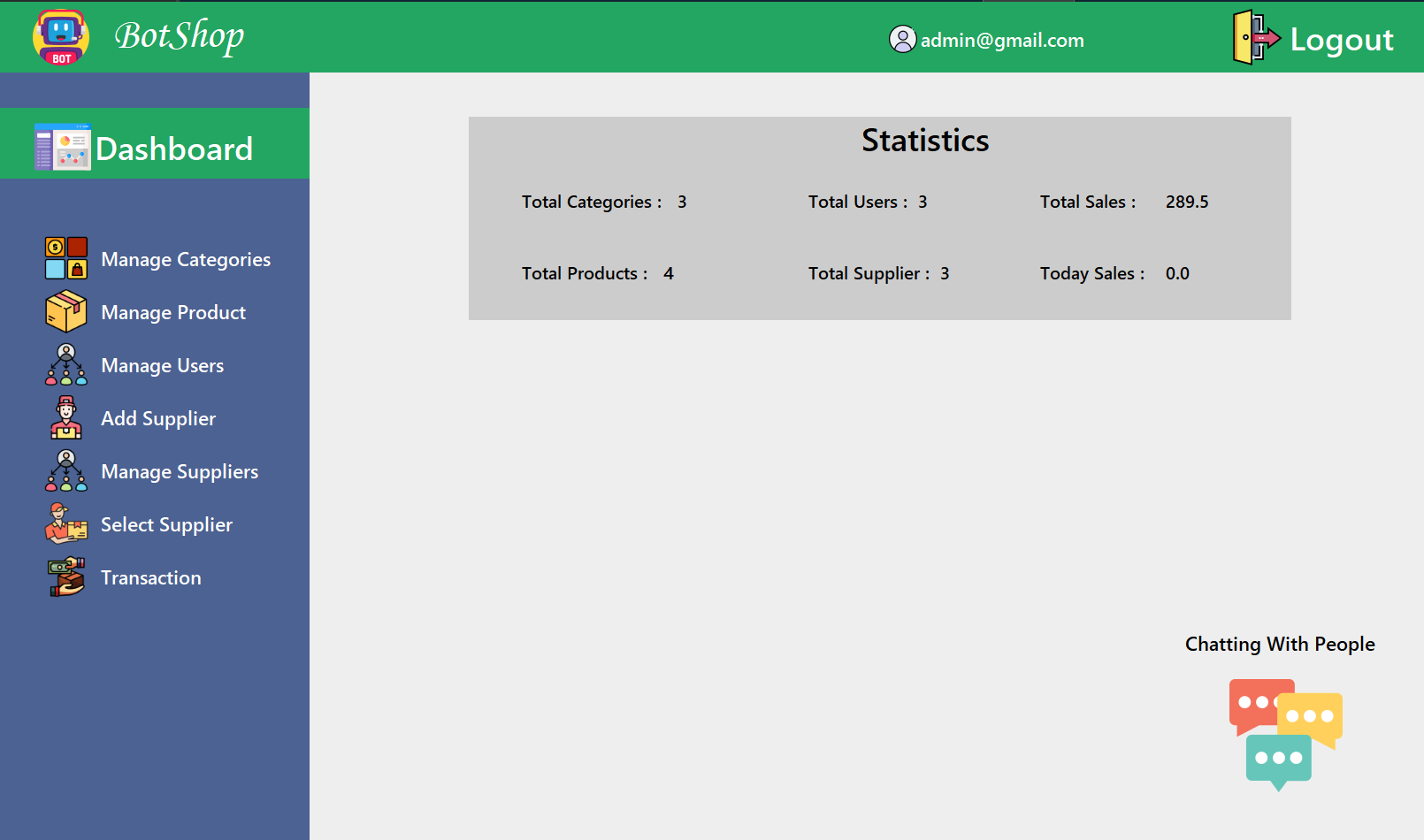
* “Select Shipper” interface: Select a shipper for on-demand delivery.

#### Figure 4.19 Select Shipper interface

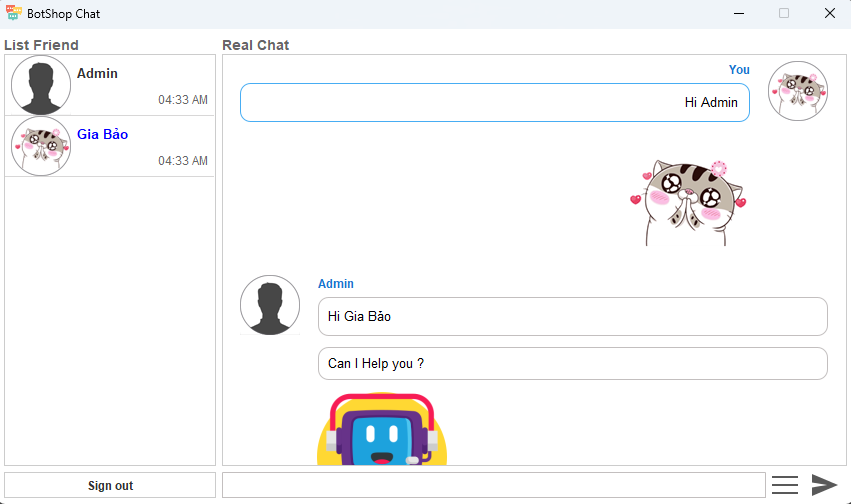
* “Transaction” interface: Review orders that have been paid for (successfully delivered).

#### Figure 4.20 Transaction interface

\* Group chat feature between users, admin, and shippers

On each user's dashboard after logging in, there will be a “Chat” icon at the bottom right corner of the screen. Click on it to open the chat interface.

#### Figure 4.21 About “Chat”



#### Figure 4.22 Demo Chat

# CHAPTER 5. CONCLUSION AND FUTURE DIRECTIONS

## 5.1 Conclusion

### 5.1.1 Achievements

Completed core functionalities: The application successfully meets all specified requirements, including connecting to the SQL Server database, performing CRUD operations on data, displaying data on the user interface, and handling user events...

Intuitive user interface: The interface is designed to be simple and user-friendly, facilitating easy and convenient user interaction.

Stable and smooth performance: The application runs smoothly and quickly, meeting the user's needs well.

Basic security features integrated: Basic security features are implemented to ensure data protection.

### 5.1.2 Limitations

**Reliance on SQL Server:** The application currently only works with the SQL Server database management system and does not support other management systems such as MySQL, Oracle, PostgreSQL.

**Unimpressive user interface:** Although simple and easy to use, the current user interface is not yet attractive and does not make a strong impression on users.

**Lack of advanced features:** The application currently focuses on basic functions and does not integrate advanced features such as reporting, data analysis.

**Limited scalability:** The scalability of the application is limited, making it difficult to meet the needs of large and complex projects.

## 5.2 Future directions

**Expand database management system support:** Support additional popular database management systems such as MySQL, Oracle, PostgreSQL to meet diverse user needs.

**Improve user interface:** Design a more beautiful, impressive and user-friendly user interface to provide a better user experience.

**Add advanced features:** Integrate advanced features such as reporting, data analysis, and user management to meet the needs of large projects.

**Enhance scalability:** Design the application with better scalability to meet the needs of large and complex projects.

**Deploy web application:** Deploy the application to a web platform so that users can access and use the application anytime, anywhere.

**REFERENCES**

1. Theory of SQL Server Management Studio (SSMS):

<https://www.javatpoint.com/sql-server-management-studio>

2. Theory of Java:

<https://medium.com/@damian.hill/why-java-is-a-versatile-programming-language-6bffef2a388f>

3. Theory of Java Swing:

<https://www.javatpoint.com/java-swing>

4. Theory of SQL:

<https://www.geeksforgeeks.org/sql-tutorial/>

5. Theory of HTML and CSS:

<https://www.geeksforgeeks.org/html-css>

6. Provide support and debug issues:

<https://chatgpt.com/>

<https://stackoverflow.blog/>

7. Source for icons for the project:

<https://www.flaticon.com>