Hi-Tech Order Management System Development Report

Author: Paula Miano

Project Description

Here is the description of the project.

Project Design

Database Design

List of tables

1. Status Table:
    - StatusID (PK)
    - State
    - StateDescription

2. Categories Table:
    - CategoryID (PK)
    - CategoryName

3. Publishers Table:
    - PublisherID (PK)
    - Name
    - StatusID (FK – Status)

4. Authors Table:
    - AuthorID (PK)
    - FirstName
    - LastName
    - Email
    - StatusID (FK – Status)

5. Books Table:
    - ISBN (PK)
    - Title
    - Price
    - YearPublished
    - QOH
    - CategoryID (FK – Categories)
    - PublisherID (FK - Publisher)
    - StatusID (FK – Status)

6. BooksAuthors Junction Table:
    - (BookID, AuthorID) (composite PK)
    - BookID (FK – Books)
    - AuthorID (FK – Authors)

7. Positions Table:
    - PositionID (PK)
    - PositionTitle
    - PositionDescription

8. Employees Table:
    - EmployeeID (PK)
    - FirstName
    - LastName
    - Email
    - PhoneNumber
    - PositionID (FK – Positions)
    - Department
    - StatusID (FK – Status)

9. UserAccount Table:
    - UserID (PK)
    - EmployeeID (FK – Employees)
    - Username
    - Password (hashed)
    - Role
    - StatusID (FK – Status)
    - DateCreated
    - DateModified
    - StatusID (FK – Status)

10. EmployeesUserAccounts Junction Table:
    - (EmployeeID, UserID) (composite PK)
    - EmployeeID (FK - Employees)
    - UserID (FK – Users)

11. Customers Table:
    - CustomerID (PK)
    - Name
    - Street
    - StreetNumber
    - City
    - PostalCode
    - PhoneNumber
    - FaxNumber
    - CreditLimit

- StatusID (FK – Status)

12. Orders Table:
    - OderID (PK)
    - CustomerID (FK – Customers)
    - OrderDate
    - TotalAmount
    - StatusID (FK – Status)
    - DateCreated
    - DateModified

13. OrderDetails Table:
    - OrderDetailID (PK)
    - OrderID (FK - Orders)
    - BookID (FK – Books)
    - Quantity
    - Price

Relationships between tables

1. Status to UserAccount, Customers, Orders, Employees: One-to-Many

A single status can apply to multiple records in the UserAccount, Customers, Orders, and Employees tables. For instance, a status like "Active" could apply to many users, customers, orders, and employees.

Relationship: 1 (Status): 0...* (UserAccount, Customers, Orders, Employees)

2. Categories to Books: One-to-Many

Each category can include multiple books, but each book belongs to exactly one category.

Relationship: 1 (Categories): 0...* (Books)

3. Publishers to Books: One-to-Many

Each publisher can publish multiple books, but each book is published by exactly one publisher.

Relationship: 1 (Publishers): 0...* (Books)

4. Books to Authors (via BooksAuthors): Many-to-Many

Each book can have multiple authors, and each author can write multiple books. This is represented through the BooksAuthors junction table.

Relationship: * (Books): * (Authors) via (BooksAuthors)

5. Positions to Employees: One-to-Many

Each position can be held by multiple employees, but each employee holds exactly one position at a time.

Relationship: 1 (Positions): 0...* (Employees)

6. Employees to UserAccount: One-to-Many

Each employee can have multiple user accounts, but each user account is associated with exactly one employee.

Relationship: 1 (Employees): 0...* (UserAccount)

7. Customers to Orders: One-to-Many

Each customer can place multiple orders, but each order is associated with exactly one customer.

Relationship: 1 (Customers): 0...* (Orders)

8. Employees to Orders: One-to-Many

Each employee can create many orders, but one order is associated with exactly one order.

Relationship: 1 (Employees): 0…* (Orders)

9. Orders to OrderDetails: One-to-Many

Each order can contain multiple order details (representing different books ordered), but each order detail line is associated with exactly one order.

Relationship: 1 (Orders): 0...* (OrderDetails)

10. Books to OrderDetails: One-to-Many

A single book can appear in multiple order details (across different orders), but each order detail line references exactly one book.

Relationship: 1 (Books): 0...* (OrderDetails)

11. EmployeesOrder Junction Table to Employees and Orders: Many-to-Many

Each employee can be associated with multiple orders, and each order can involve multiple employees. This relationship is managed through the EmployeesOrder Junction Table, establishing a many-to-many relationship between the Employees and Orders tables.

Relationship: * (Employees): * (Orders) via (EmployeesOrder Junction Table)

Tables Fields

1. Status Table

Field Name	Data Type	Design Notes	Example
StatusID	INT	Primary key, auto-increments	1
State	NVARCHAR(50)	Describes the status name	'Active'
StateDescription	NVARCHAR(255)	Detailed description of the state	'The record is active and operational.'

2. Categories Table

Field Name	Data Type	Design Notes	Example
CategoryID	INT	Primary key, auto-increments	1
CategoryName	NVARCHAR(100)	Name of the book category	'Science Fiction'

3. Publishers Table

Field Name	Data Type	Design Notes	Example
PublisherID	INT	Primary key, auto-increments	1
Name	NVARCHAR(255)	Name of the publisher	'Penguin Random House'
StatusID	INT	Foreign key to Status table	1

4. Authors Table

Field Name	Data Type	Design Notes	Example
AuthorID	INT	Primary key, auto-increments	1
FirstName	NVARCHAR(100)	First name of the author	'Jane'
LastName	NVARCHAR(100)	Last name of the author	'Doe'
Email	NVARCHAR(255)	Email address of the author	'jane.doe@example.com'
StatusID	INT	Foreign key to Status table	1

5. Books Table

Field Name	Data Type	Design Notes	Example
ISBN	VARCHAR(13)	Primary key, ISBN number of the book	'978-3-16-148410-0'
Title	NVARCHAR(255)	Title of the book	'Introduction to Database Systems'
Price	DECIMAL(10,2)	Price of the book	49.99
YearPublished	INT	Year the book was published	2020
QOH	INT	Quantity on hand	50
CategoryID	INT	Foreign key to Categories table	1
PublisherID	INT	Foreign key to Publishers table	1
StatusID	INT	Foreign key to Status table	1

6. Positions Table

Field Name	Data Type	Design Notes	Example
PositionID	INT	Primary key, auto-increments	1
PositionTitle	NVARCHAR(100)	Title of the position	'Manager'
PositionDescription	NVARCHAR(255)	Description of the position's duties	'Oversees department operations'

7. Employees Table

Field Name	Data Type	Design Notes	Example
EmployeeID	INT	Primary key, auto-increments	1
FirstName	NVARCHAR(100)	First name of the employee	'John'
LastName	NVARCHAR(100)	Last name of the employee	'Doe'
Email	NVARCHAR(255)	Email address of the employee	'john.doe@example.com'
PhoneNumber	NVARCHAR(15)	Contact phone number of the employee	'555-1234'
PositionID	INT	Foreign key to Positions table	1
Department	NVARCHAR(100)	Department where the employee works	'IT'
StatusID	INT	Foreign key to Status table	1

8. UserAccount Table

Field Name	Data Type	Design Notes	Example
UserID	INT	Primary key, auto-increments	1
EmployeeID	INT	Foreign key to Employees table	1
Username	NVARCHAR(50)	Unique username for user login	'johndoe123'
Password	NVARCHAR(255)	User's password (plain text for simplicity)	'password123'
Role	NVARCHAR(50)	Role of the user in the system	'Administrator'
DateCreated	DATETIME	Date and time the account was created	'2023-01-01 12:00:00'
DateModified	DATETIME	Date and time the account was last modified	'2023-01-02 12:00:00'
StatusID	INT	Foreign key to Status table	1

9. Customers Table

Field Name	Data Type	Design Notes	Example
CustomerID	INT	Primary key, auto-increments	1
Name	NVARCHAR(255)	Full name or company name of the customer	'Jane Doe'
Street	NVARCHAR(255)	Street address of the customer	'123 Main St'
StreetNumber	NVARCHAR(10)	Street number, if applicable	'123'
City	NVARCHAR(100)	City where the customer resides	'Anytown'
PostalCode	NVARCHAR(10)	Postal code of the customer's address	'12345'
PhoneNumber	NVARCHAR(15)	Contact phone number of the customer	'555-6789'
FaxNumber	NVARCHAR(15)	Fax number of the customer, if applicable	'555-6788'
CreditLimit	DECIMAL(10, 2)	Credit limit extended to the customer	5000.00
StatusID	INT	Foreign key to Status table	1

10. Orders Table

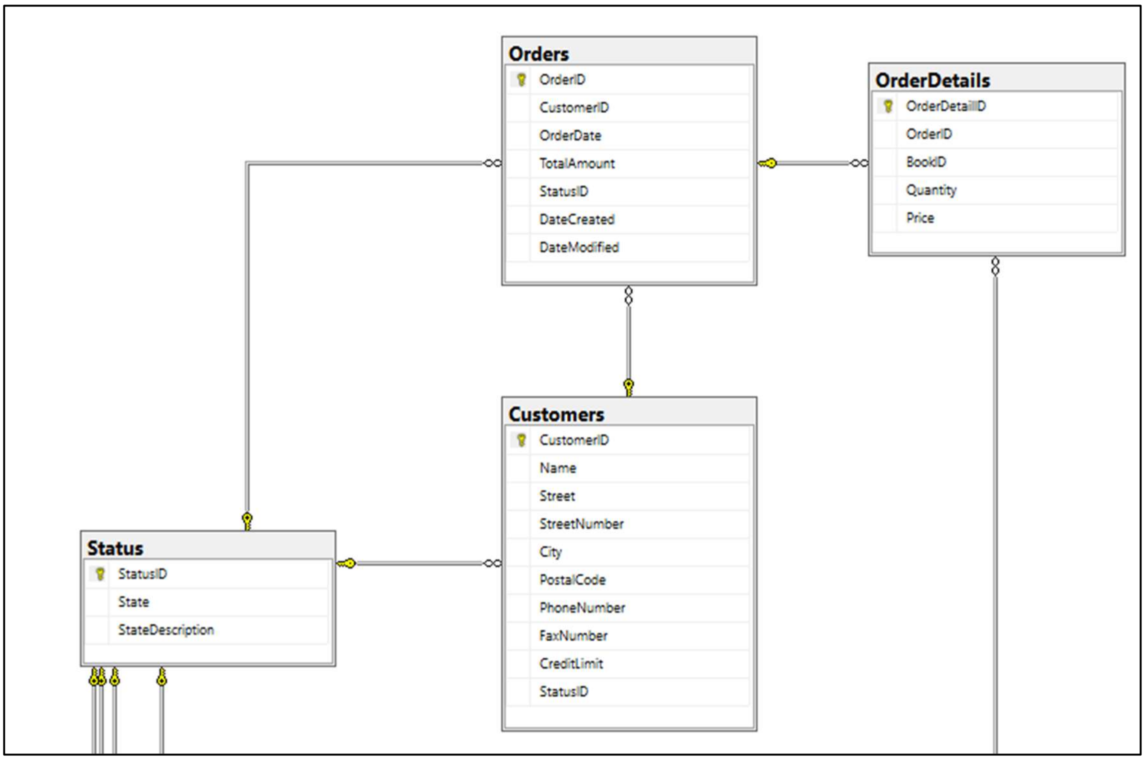
Field Name	Data Type	Design Notes	Example
OrderID	INT	Primary key, auto-increments	1
CustomerID	INT	Foreign key to Customers table	1
OrderDate	DATETIME	Date and time the order was placed	'2023-01-03 10:00:00'
TotalAmount	DECIMAL(10,2)	Total cost of the order	99.99
StatusID	INT	Foreign key to Status table	1
DateCreated	DATETIME	Date and time the order record was created	'2023-01-03 10:00:00'
DateModified	DATETIME	Date and time the order record was last modified	'2023-01-04 10:00:00'

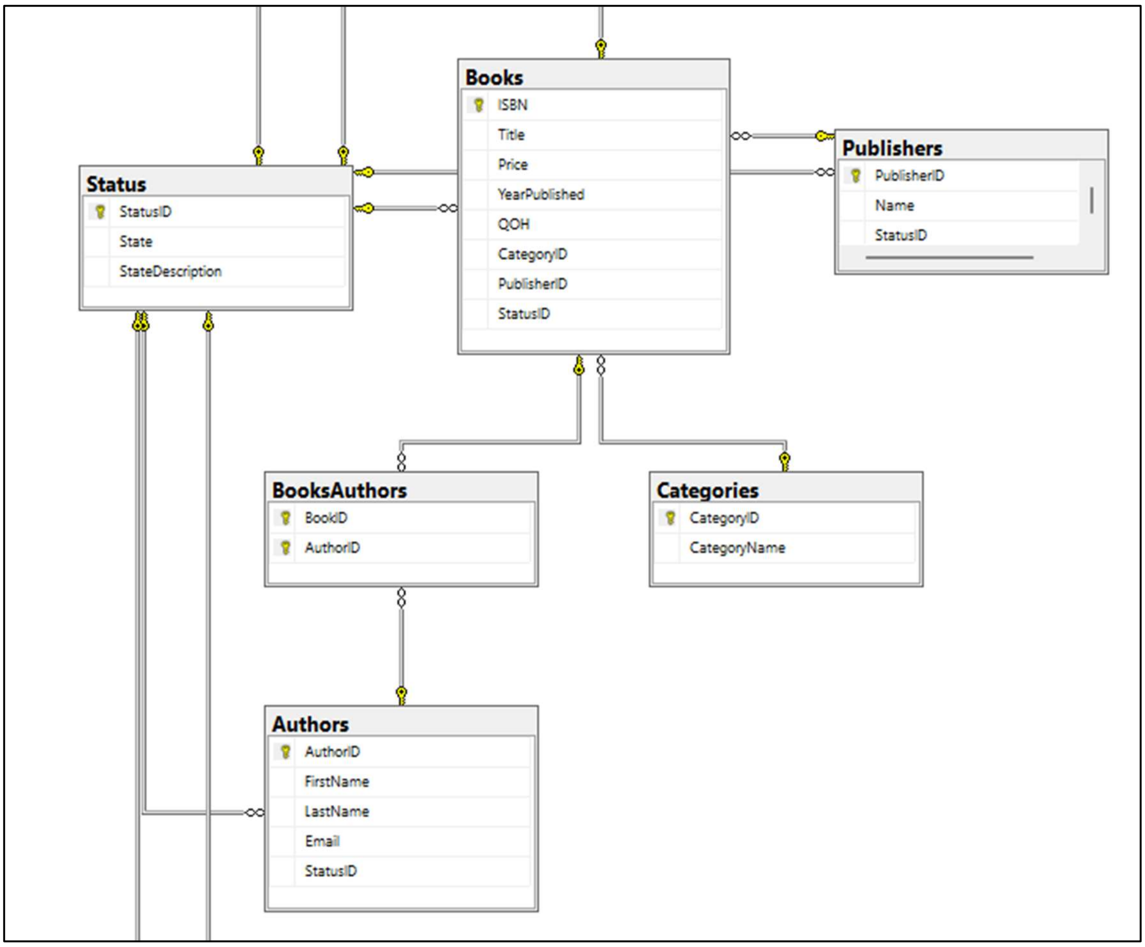
11. OrderDetails Table

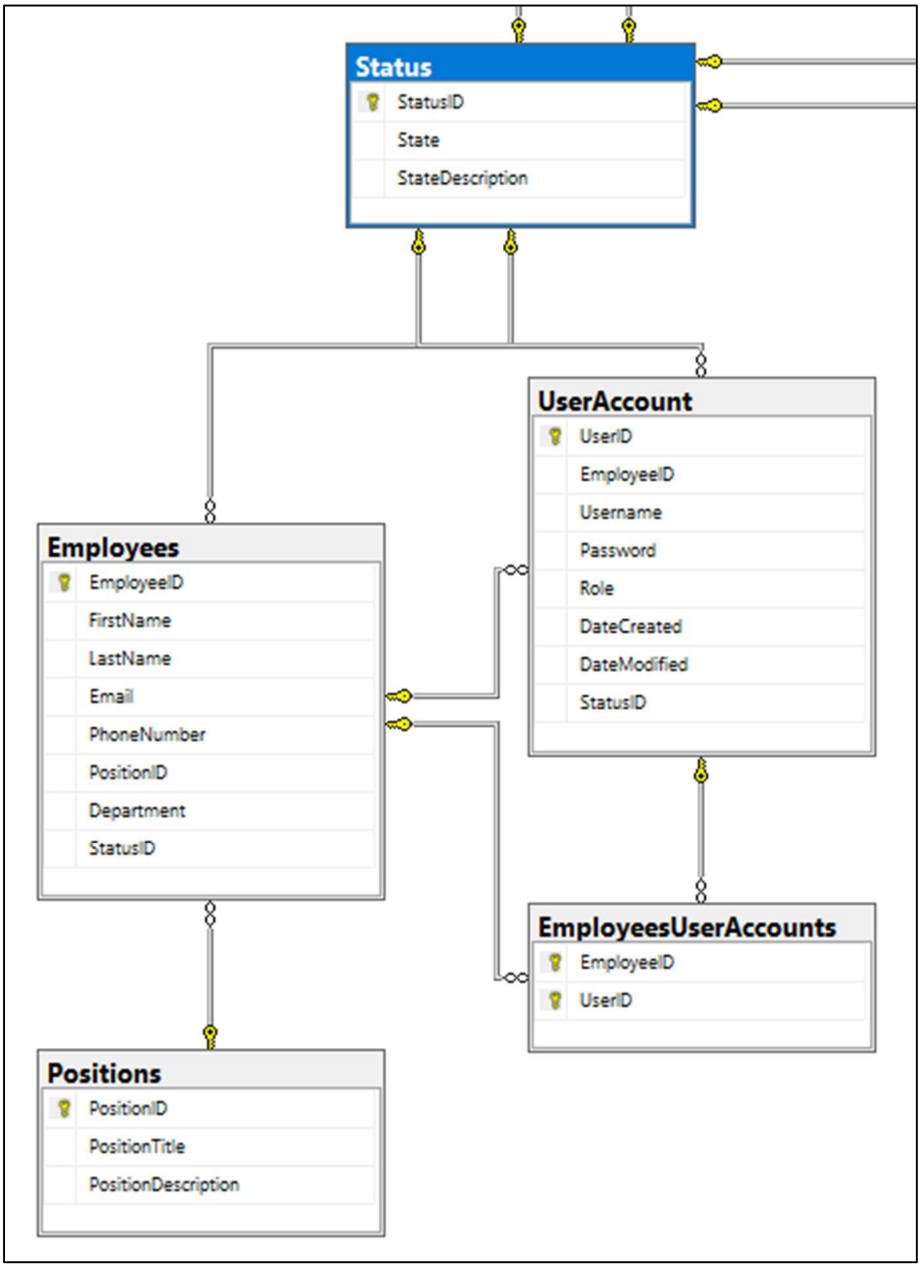
Field Name	Data Type	Design Notes	Example
OrderDetailID	INT	Primary key, auto-increments	1
OrderID	INT	Foreign key to Orders table	1

BookID	VARCHAR(13)	Foreign key to Books table	'978-3-16-148410-0'
Quantity	INT	Number of copies of the book ordered	2
Price	DECIMAL(10, 2)	Price at which each book was sold	49.99

Database Diagram







Design of Application Domain Classes

The application domain classes in the Multitier Hi-Tech Distribution Management System are centralized within the Hi-TechLibrary project. This centralization ensures that the core business logic and data models are consistently applied across the system. The domain classes strictly model the business requirements, which is crucial for maintaining data integrity and enforcing business rules consistently throughout the application.

Using Entity Framework for inventory and orders management allows for a high level of abstraction in database access. This ORM simplifies the management of complex data interactions and reduces the need for extensive boilerplate code related to data operations. By separating these domain classes from the graphical user interface modules, the system adheres to a clean architecture principle, keeping business logic distinct from presentation concerns. This separation facilitates independent development and testing of each component, optimizing performance and maintainability.

Design of Data Access Classes

The data access classes in the Hi-TechLibrary project are designed to handle both connected and disconnected modes of database interaction, depending on the requirements of different system components. The connected mode is used for employee management, utilizing ADO.NET to ensure immediate consistency and real-time data processing, which are essential for operations that involve sensitive data manipulation.

Conversely, the disconnected mode, employed for customer management, utilizes batch processing to accommodate scenarios where network connectivity may be inconsistent. This method allows for local data caching and batch updates, which can be synchronized with the database once connectivity is restored or at scheduled intervals, enhancing the system's usability under varied network conditions.

The use of Entity Framework for managing more complex entities such as inventory and orders facilitates easier data manipulation and reduces the risk of SQL-related errors. It automates many aspects of database interaction, including query generation and schema updates, allowing developers to focus more on business logic rather than data handling specifics.

Overall, the design of the data access classes ensures efficient, secure, and reliable database interactions, providing a solid foundation for the system's data management needs. This structure supports easy updates and scalability, enabling future enhancements with minimal impact on existing functionalities.

Project Implementation

Problems Faced During Implementation

Here's a summary of the main problems faced during the implementation:

1. Handling Null Reference Exceptions

One of the recurrent issues encountered was related to null reference exceptions. This problem was particularly prevalent when dealing with GUI elements and data manipulation, highlighting the importance of thorough null checks and proper initialization of objects before use. Ensuring that all variables and objects were correctly instantiated before accessing their members was crucial in resolving these exceptions.

2. Database Connectivity and CRUD Operations

Ensuring reliable and efficient database connectivity posed challenges, especially when implementing CRUD (Create, Read, Update, Delete) operations. Managing connections, executing SQL commands, and handling data retrieval and updates required meticulous attention to detail to prevent data inconsistency and ensure the integrity of transactions.

3. Data Conversion and Parsing Errors

Errors related to data conversion and parsing were encountered, particularly when converting text input into numerical data types or parsing enum values. These issues underscored the need for robust data validation and error handling mechanisms to prevent runtime exceptions and ensure data integrity.

4. User Input Validation

Validating user input to ensure it met the expected format and constraints was another challenge. This was crucial for maintaining data integrity and preventing SQL injection attacks. Implementing comprehensive validation logic, including regular expressions and custom validation rules, was essential for mitigating these issues.

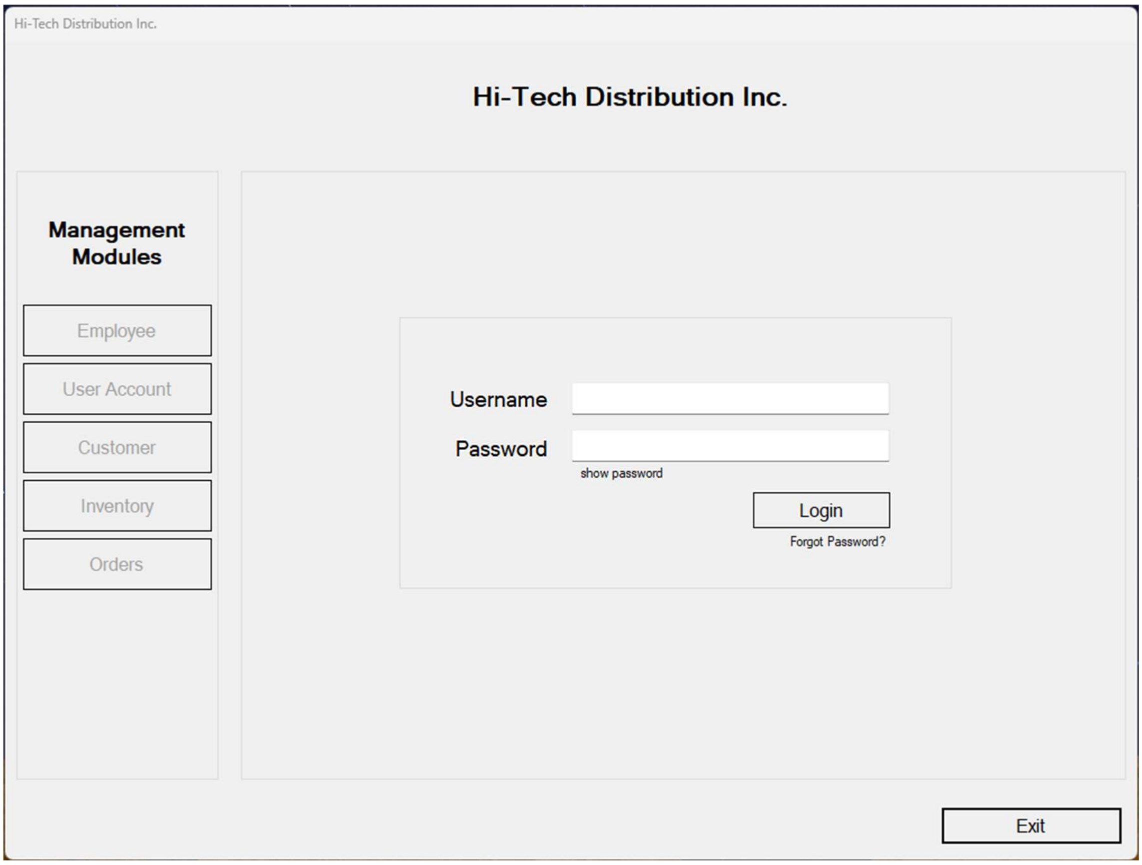
5. Dynamic UI Updates Based on Data Changes

Dynamically updating the user interface in response to data changes presented difficulties, especially in scenarios where user actions in one part of the application needed to reflect immediately in another. Ensuring a seamless and reactive user experience required careful design of event handlers and state management.

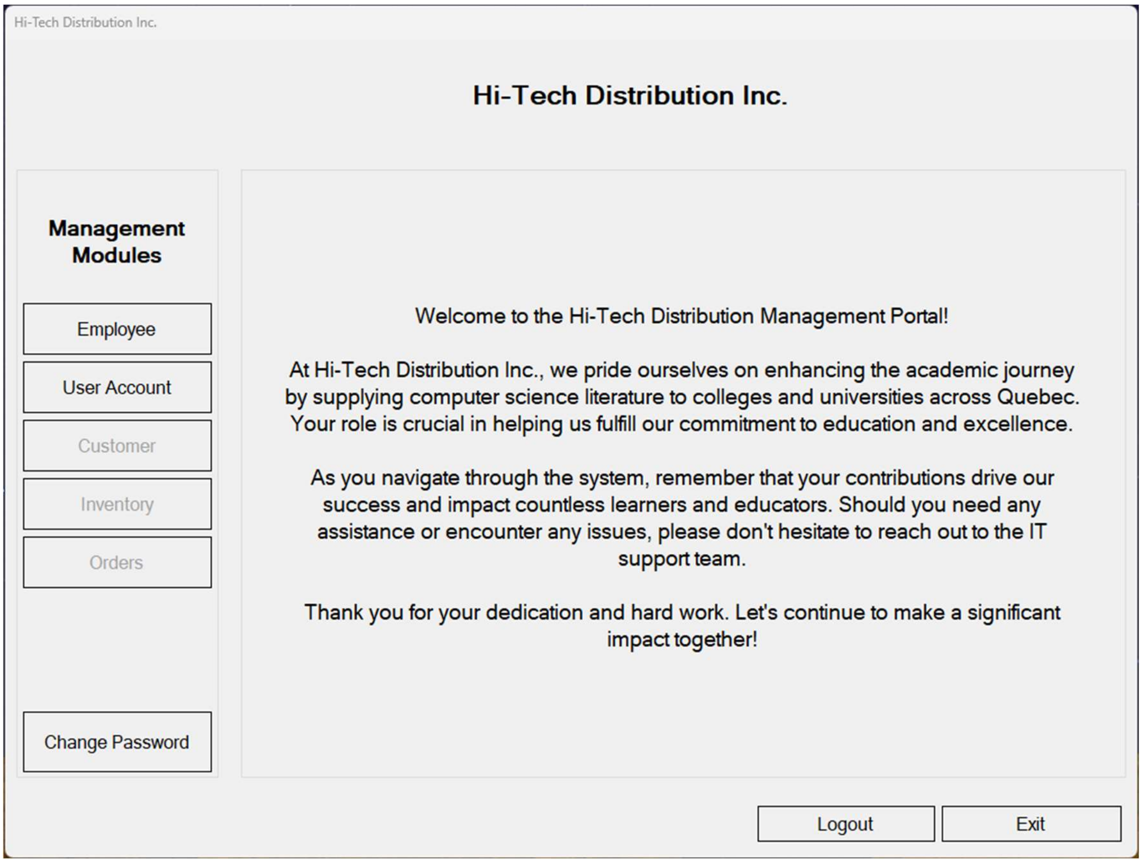
6. Enum Handling in GUI Elements

Integrating enums with GUI elements, such as ComboBoxes, for roles or statuses, and ensuring their correct display and selection was challenging. This involved mapping enum values to user-friendly strings and vice versa, which required implementing utility methods for enum description handling.

Project Testing



All buttons are disabled before login, except for Login and Exit.



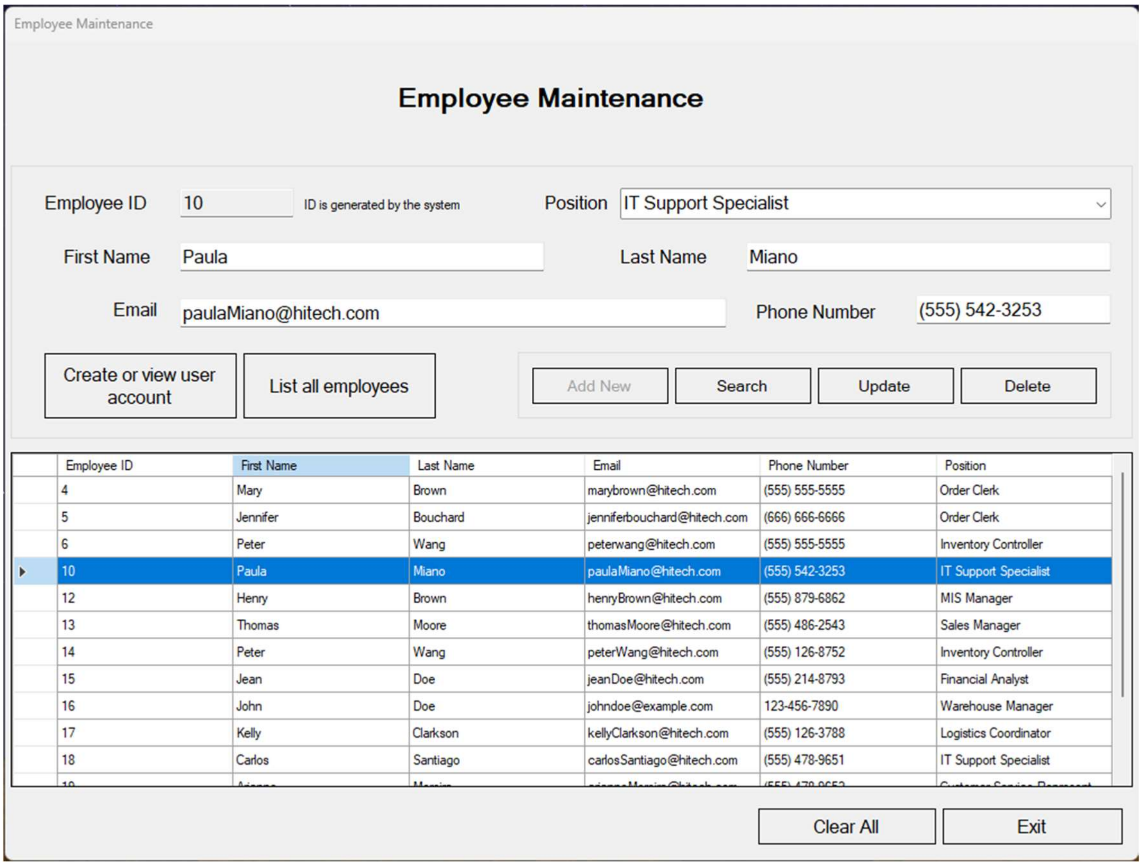
Welcome to the Hi-Tech Distribution Management Portal!

At Hi-Tech Distribution Inc., we pride ourselves on enhancing the academic journey by supplying computer science literature to colleges and universities across Quebec. Your role is crucial in helping us fulfill our commitment to education and excellence.

As you navigate through the system, remember that your contributions drive our success and impact countless learners and educators. Should you need any assistance or encounter any issues, please don't hesitate to reach out to the IT support team.

Thank you for your dedication and hard work. Let's continue to make a significant impact together!

Login in enables only the modules to which the user is allowed access.

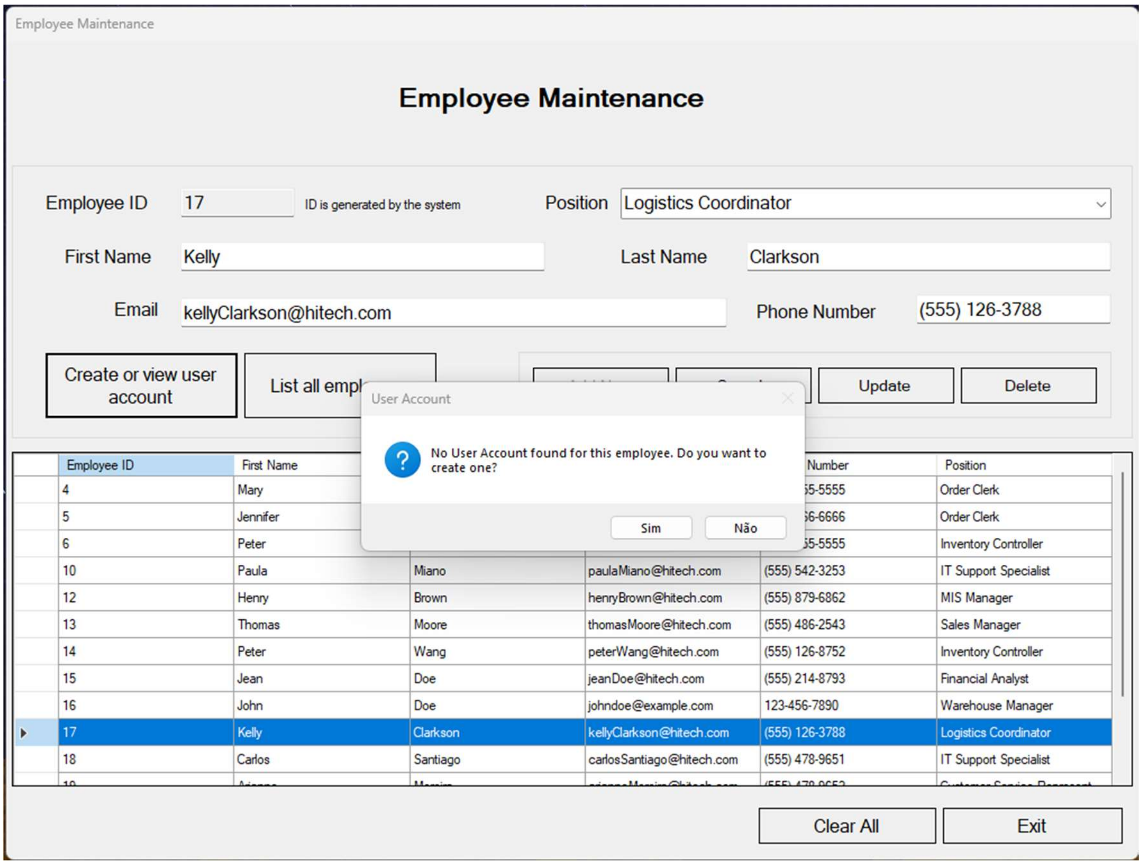


Employee Maintenance

Employee ID	10 (ID is generated by the system)	Position	IT Support Specialist
First Name	Paula	Last Name	Miano
Email	paulaMiano@hitech.com	Phone Number	(555) 542-3253

Employee ID	First Name	Last Name	Email	Phone Number	Position
4	Mary	Brown	marybrown@hitech.com	(555) 555-5555	Order Clerk
5	Jennifer	Bouchard	jenniferbouchard@hitech.com	(666) 666-6666	Order Clerk
6	Peter	Wang	peterwang@hitech.com	(555) 555-5555	Inventory Controller
10	Paula	Miano	paulaMiano@hitech.com	(555) 542-3253	IT Support Specialist
12	Henry	Brown	henryBrown@hitech.com	(555) 879-6862	MIS Manager
13	Thomas	Moore	thomasMoore@hitech.com	(555) 486-2543	Sales Manager
14	Peter	Wang	peterWang@hitech.com	(555) 126-8752	Inventory Controller
15	Jean	Doe	jeanDoe@hitech.com	(555) 214-8793	Financial Analyst
16	John	Doe	johndoe@example.com	123-456-7890	Warehouse Manager
17	Kelly	Clarkson	kellyClarkson@hitech.com	(555) 126-3788	Logistics Coordinator
18	Carlos	Santiago	carlosSantiago@hitech.com	(555) 478-9651	IT Support Specialist

All employees are shown in a list and clicking on one fills out the employee information. Searches by ID, name or email can also be done by filling the information in the corresponding text box and clicking search. Update and delete functionalities are also implemented in this form.



Employee Maintenance

Employee Maintenance

Employee ID	17	Position	Logistics Coordinator
First Name	Kelly	Last Name	Clarkson
Email	kellyClarkson@hitech.com	Phone Number	(555) 126-3788

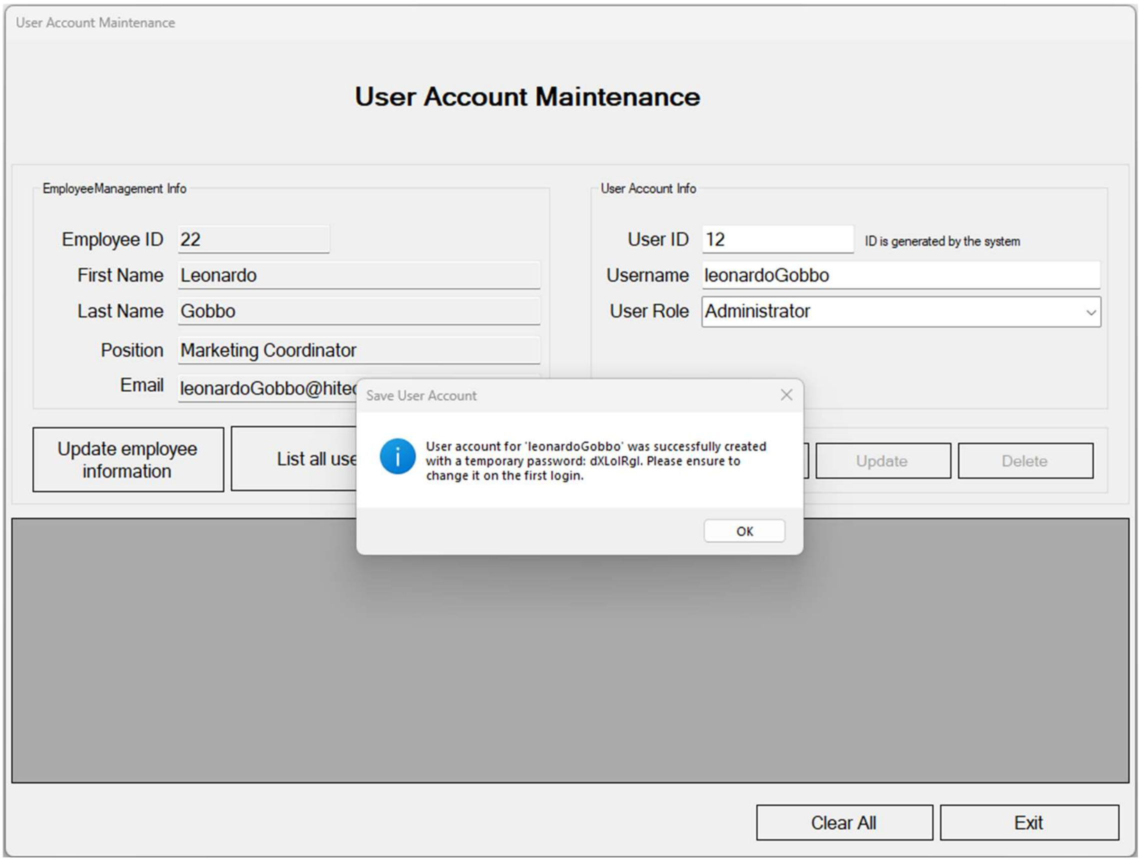
ID is generated by the system

User Account

No User Account found for this employee. Do you want to create one?

Employee ID	First Name			Number	Position
4	Mary			55-5555	Order Clerk
5	Jennifer			56-6666	Order Clerk
6	Peter			55-5555	Inventory Controller
10	Paula	Miano	paulaMiano@hitech.com	(555) 542-3253	IT Support Specialist
12	Henry	Brown	henryBrown@hitech.com	(555) 879-6862	MIS Manager
13	Thomas	Moore	thomasMoore@hitech.com	(555) 486-2543	Sales Manager
14	Peter	Wang	peterWang@hitech.com	(555) 126-8752	Inventory Controller
15	Jean	Doe	jeanDoe@hitech.com	(555) 214-8793	Financial Analyst
16	John	Doe	johndoe@example.com	123-456-7890	Warehouse Manager
17	Kelly	Clarkson	kellyClarkson@hitech.com	(555) 126-3788	Logistics Coordinator
18	Carlos	Santiago	carlosSantiago@hitech.com	(555) 478-9651	IT Support Specialist

Clicking on the button that links to the user account checks if an account already exists. If not, it opens the form in creation mode. If it does, opens the form in update mode. The employee information is sent to the user account form in read only mode.



User Account Maintenance

EmployeeManagement Info

Employee ID: 22
First Name: Leonardo
Last Name: Gobbo
Position: Marketing Coordinator
Email: leonardoGobbo@hitec

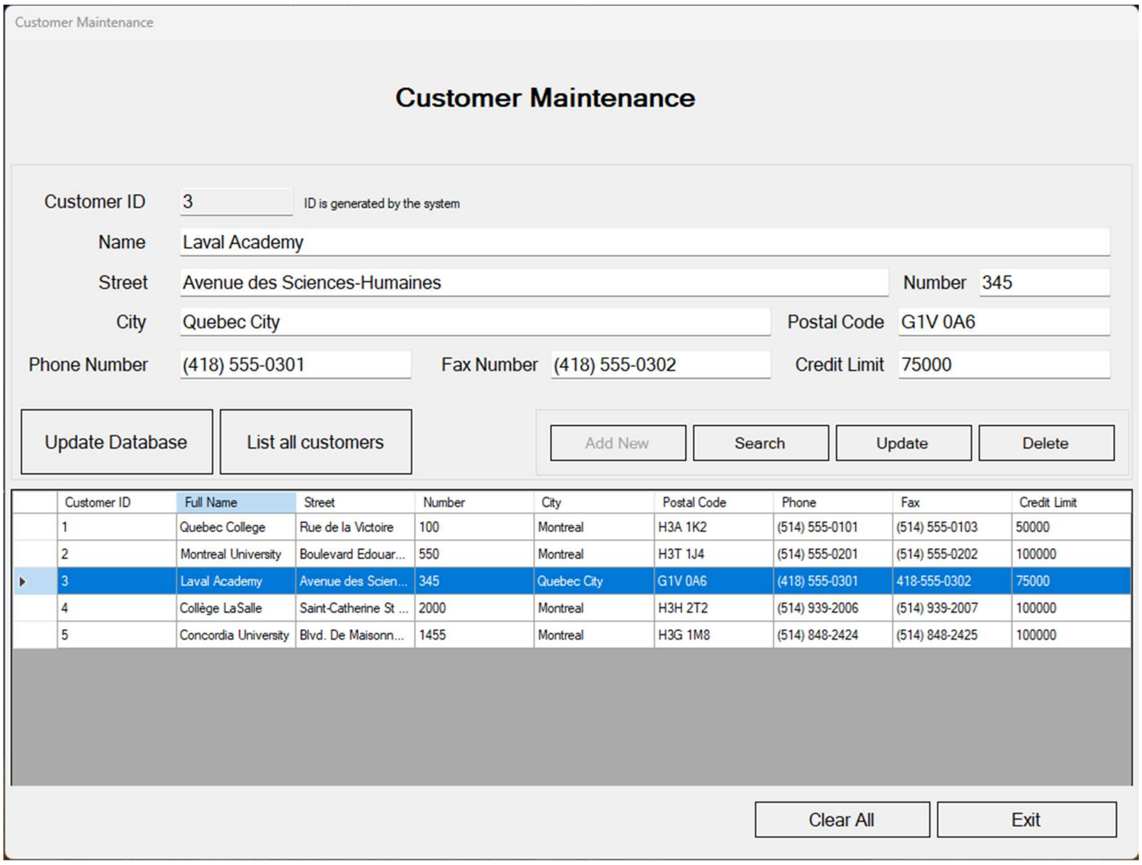
User Account Info

User ID: 12 — ID is generated by the system
Username: leonardoGobbo
User Role: Administrator

Save User Account

User account for 'leonardoGobbo' was successfully created with a temporary password: dXLoIRgl. Please ensure to change it on the first login.

When a user account is created, a temporary password is generated, and the user must change it on the first login.

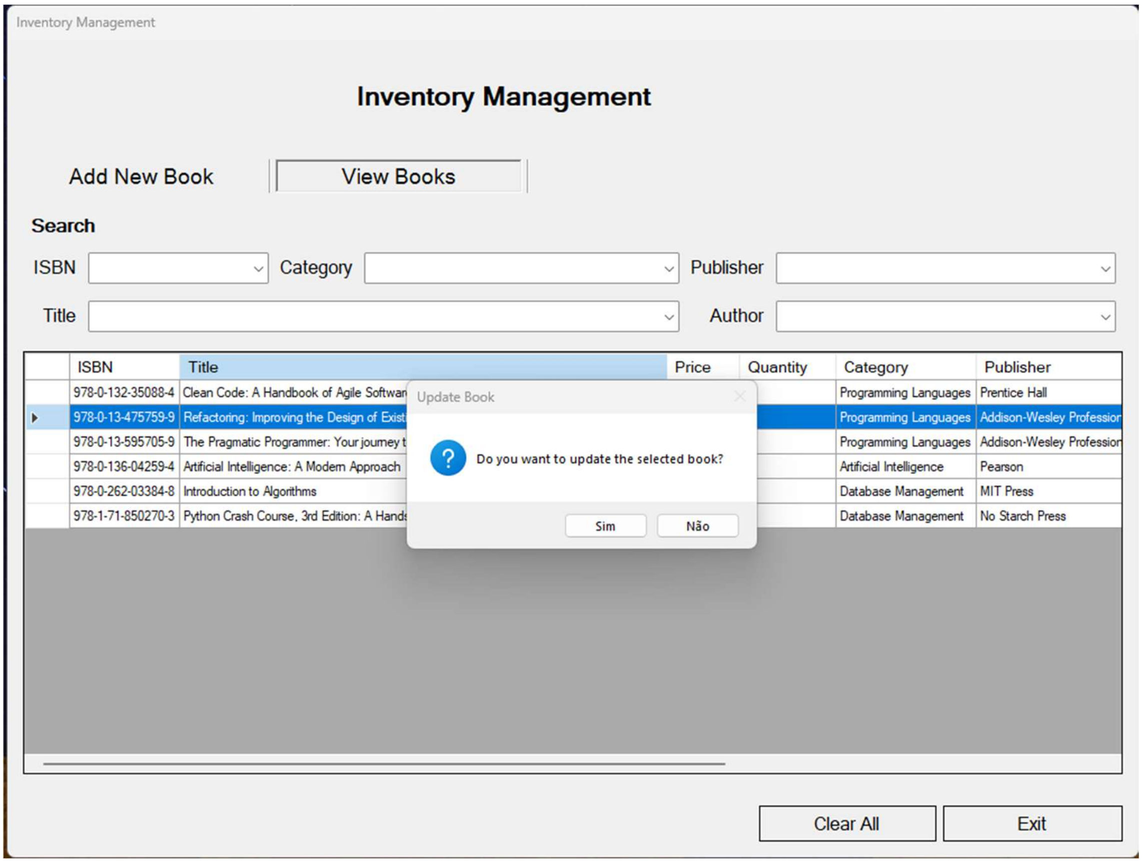


Customer Maintenance

Customer ID: 3 — ID is generated by the system
Name: Laval Academy
Street: Avenue des Sciences-Humaines
Number: 345
City: Quebec City
Postal Code: G1V 0A6
Phone Number: (418) 555-0301
Fax Number: (418) 555-0302
Credit Limit: 75000

	Customer ID	Full Name	Street	Number	City	Postal Code	Phone	Fax	Credit Limit
	1	Quebec College	Rue de la Victoire	100	Montreal	H3A 1K2	(514) 555-0101	(514) 555-0103	50000
	2	Montreal University	Boulevard Edouar...	550	Montreal	H3T 1J4	(514) 555-0201	(514) 555-0202	100000
▶	3	Laval Academy	Avenue des Scien...	345	Quebec City	G1V 0A6	(418) 555-0301	418-555-0302	75000
	4	Collège LaSalle	Saint-Catherine St ...	2000	Montreal	H3H 2T2	(514) 939-2006	(514) 939-2007	100000
	5	Concordia University	Blvd. De Maisonn...	1455	Montreal	H3G 1M8	(514) 848-2424	(514) 848-2425	100000

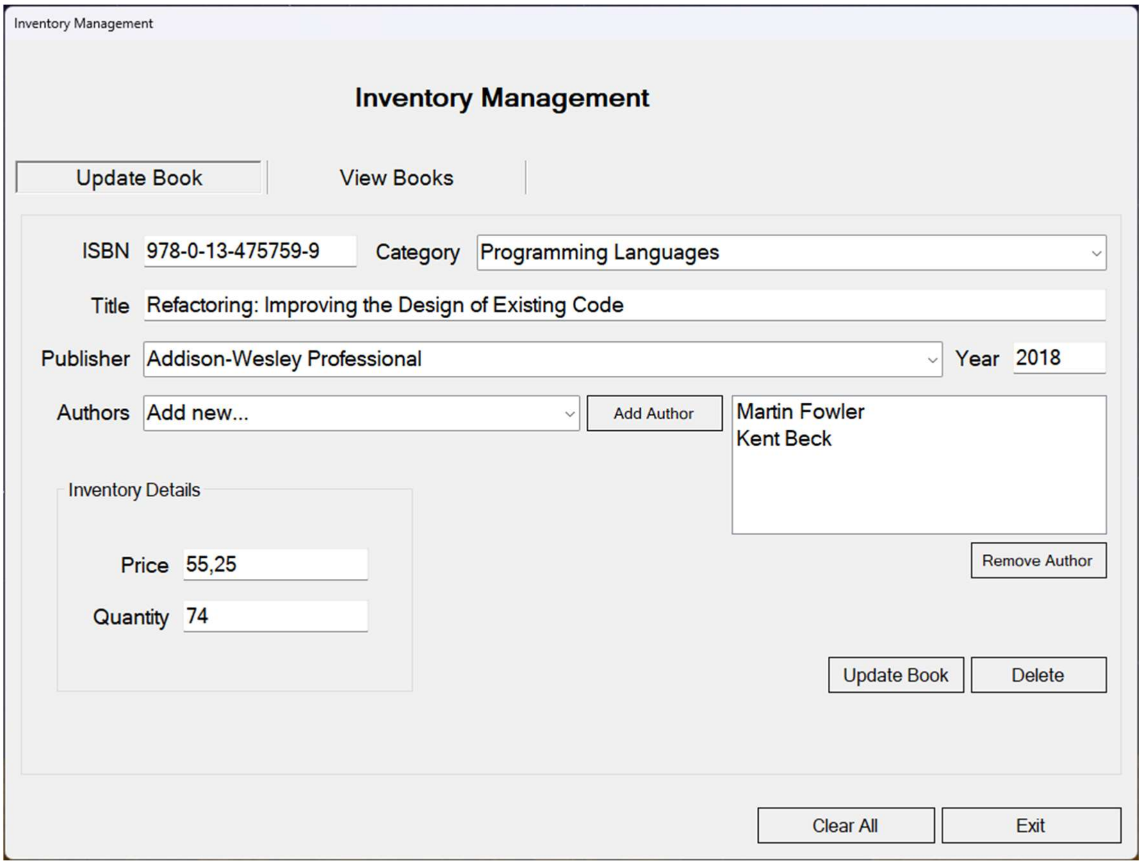
Customer Maintenance form has the same functionality as the previous ones. There is a button to update the database, as changes are made on the data set first (and can be seen in the data grid view) before being committed to the database.



Do you want to update the selected book?

ISBN	Title	Price	Quantity	Category	Publisher
978-0-132-35088-4	Clean Code: A Handbook of Agile Softwar			Programming Languages	Prentice Hall
978-0-13-475759-9	Refactoring: Improving the Design of Exist			Programming Languages	Addison-Wesley Professior
978-0-13-595705-9	The Pragmatic Programmer: Your journey t			Programming Languages	Addison-Wesley Professior
978-0-136-04259-4	Artificial Intelligence: A Modern Approach			Artificial Intelligence	Pearson
978-0-262-03384-8	Introduction to Algorithms			Database Management	MIT Press
978-1-71-850270-3	Python Crash Course, 3rd Edition: A Hands			Database Management	No Starch Press

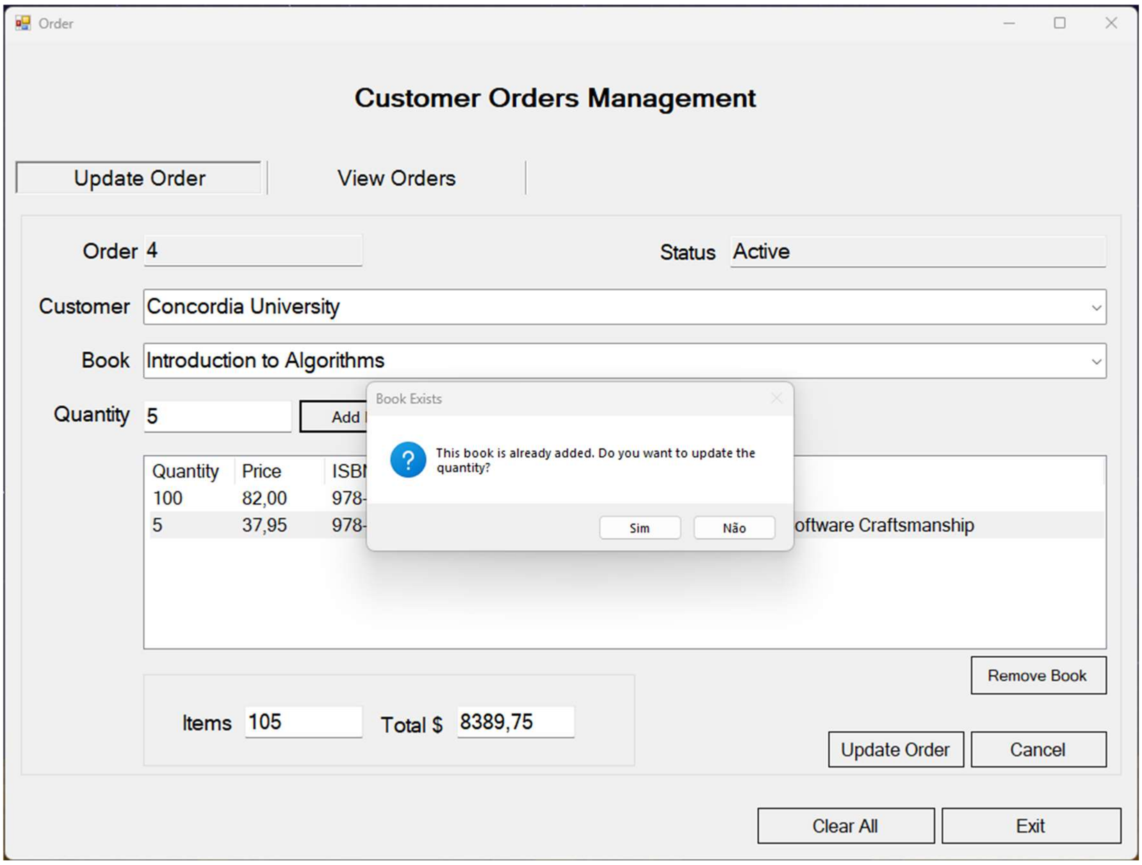
On Inventory Management the functionalities are divided in two tabs, one for searching and another one for adding and updating. If you click on a book in the search tab the system asks if you want to update it. Pressing yes redirects the user to the add or update tab.



ISBN 978-0-13-475759-9
Category Programming Languages
Title Refactoring: Improving the Design of Existing Code
Publisher Addison-Wesley Professional
Year 2018
Authors Add new...
Martin Fowler
Kent Beck

Inventory Details
Price 55,25
Quantity 74

The add new or update tab allows the user to insert or change a book's details, including adding a new publisher or a new author to the database.



Customer Orders Management

Order 4

Status Active

Customer Concordia University

Book Introduction to Algorithms

Quantity 5

Quantity	Price	ISBI	
100	82,00	978-	
5	37,95	978-	oftware Craftsmanship

Book Exists

This book is already added. Do you want to update the quantity?

Sim Não

Items 105 Total $ 8389,75

Orders management works similarly to inventory management. When the user tries to add the same book to the order the system asks if he wants to update the quantity instead.

Conclusion

The development of the Multitier Hi-Tech Distribution Management System has been a significant part of my educational journey as a programming student, providing a practical framework for applying theoretical knowledge to real-world software development challenges. The project has underscored the importance of a well-thought-out design in achieving a scalable and maintainable software architecture.

While the system has successfully integrated multiple modules catering to different functionalities such as order processing, inventory management, and customer and employee management, there were several functionalities that could not be completed due to time constraints. Notably, the integration for checking available inventory quantities before placing orders, and constraints to prevent the deletion of books included in orders or employees linked to user accounts were not implemented. These features are crucial for ensuring the operational integrity of the system and would be the primary focus of future development efforts.

Implementing these functionalities would not only refine the system's usability and reliability but also enhance its data integrity and error handling capabilities. Such improvements are essential for a fully operational commercial system and represent the next steps in this project's lifecycle.

Reflecting on this project, it has been immensely valuable for my growth as a programmer. It challenged me to implement and manage a complex system architecture, deepening my understanding of both front-end and back-end development practices. Moreover, it emphasized the importance of rigorous testing and thoughtful design in building software that is both robust and adaptable.

The skills and insights gained from this project are invaluable, laying a solid foundation for my future endeavors in software development. As I move forward, I will carry forward the lessons learned from this project, particularly the importance of comprehensive planning and the integration of various software components into a cohesive system. This project has not only enhanced my technical skills but also enriched my problem-solving abilities, preparing me for more complex challenges in the tech industry.