VC++ .NET MSSQL Exercise: Inventory Management System

Task:
Create a C++/CLI Windows Forms application that interfaces with a Microsoft SQL Server database to manage an inventory system. The application should perform the following operations:

1. Connect to a MSSQL database using ADO.NET.
2. Implement CRUD (Create, Read, Update, Delete) operations for inventory items.
3. Display inventory data in a DataGridView.
4. Implement a search function to filter items by various criteria.
5. Generate a simple report of low stock items.

Requirements:

1. Use C++/CLI with .NET Framework 4.7 or later and Windows Forms for the GUI.
2. Create a MSSQL database with at least two tables: Items and Categories.
3. Implement the following features:
    a. Add new items to the inventory
    b. Update existing item details
    c. Delete items from the inventory
    d. Display all items in a DataGridView
    e. Search items by name, category, or stock level
    f. Generate a report of items with stock below a specified threshold
4. Use parameterized queries to prevent SQL injection.
5. Implement proper exception handling and display user-friendly error messages.
6. Use transactions where appropriate (e.g., when updating multiple related records).
7. Implement at least one stored procedure and call it from the application.

Database Schema:

```
CREATE TABLE Categories (
    CategoryID INT PRIMARY KEY IDENTITY(1,1),
    CategoryName NVARCHAR(50) NOT NULL
);

CREATE TABLE Items (
    ItemID INT PRIMARY KEY IDENTITY(1,1),
    ItemName NVARCHAR(100) NOT NULL,
    CategoryID INT FOREIGN KEY REFERENCES Categories(CategoryID),
    Quantity INT NOT NULL,
    UnitPrice DECIMAL(10, 2) NOT NULL,
    LastUpdated DATETIME NOT NULL DEFAULT GETDATE()
);
```

Bonus:

1. Implement data validation on the client-side before sending updates to the database.
2. Add a feature to export the low stock report to a CSV file.
3. Implement basic user authentication and role-based access control.
4. Use asynchronous operations for database queries to improve UI responsiveness.