

STOCKD — PROJECT CASE STUDY

01. Project Overview

Stockd is a stock management application designed to help businesses manage their products, inventory, orders and customers in one place.

The project began as an idea for a simple stock management system and developed into a complete cross-platform application, with both web and Android functionality.

The aim was to create an application that was practical and easy to use while also having a distinctive visual identity.

[SCREENSHOT — Stockd Dashboard]

02. The Problem

Small businesses can often rely on spreadsheets, handwritten records or multiple disconnected systems to manage their stock and orders.

This can make it difficult to quickly understand what products are available, which products are running low, what orders need attention and how the business is performing.

I wanted to create a single application that brought these functions together in a clear and organised way.



03. Project Goals

The main goals of Stockd were to:

- Make inventory management simple and accessible
- Allow users to create and manage products
- Track stock quantities and low-stock levels
- Manage customer information
- Create and manage orders
- Provide an overview of business information
- Support different currencies
- Generate PDF reports
- Allow products to be deactivated
- Provide notifications
- Create a consistent and customisable interface
- Make the application available across web and Android

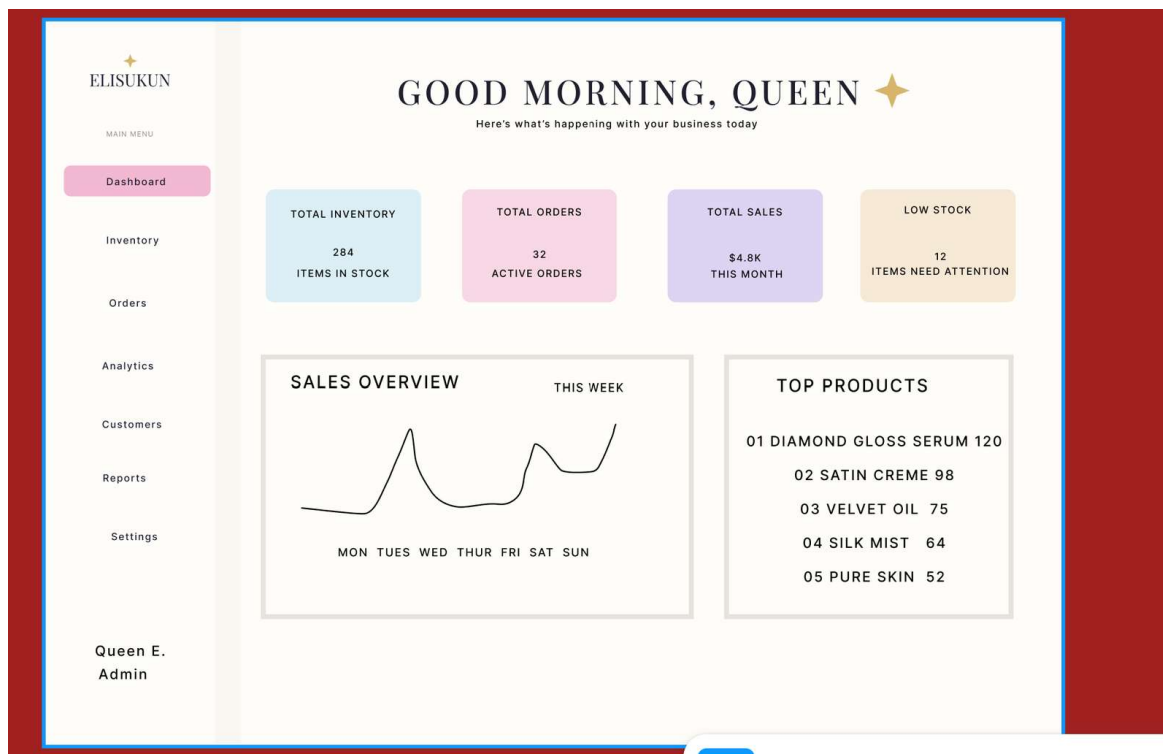
04. Planning

I planned Stockd around the main workflow of a small business:

Products → Inventory → Orders → Customers → Business Overview

This helped me decide which features were most important and how they should be connected.

The main areas of the application were organised into a simple navigation system so users could move between different parts of their business without having to search through complicated menus.



05. Design Direction

The visual identity of Stockd developed around a modern Y2K-inspired editorial style.

I experimented with pastel blues, lavender, purple, pink, white and gold-inspired accents to create an interface that felt modern and distinctive without becoming difficult to use.

The design uses rounded cards, soft gradients and clear visual hierarchy to make information easier to understand.

The final style was intended to feel nostalgic and creative while still being suitable for a professional business application.



06. User Interface

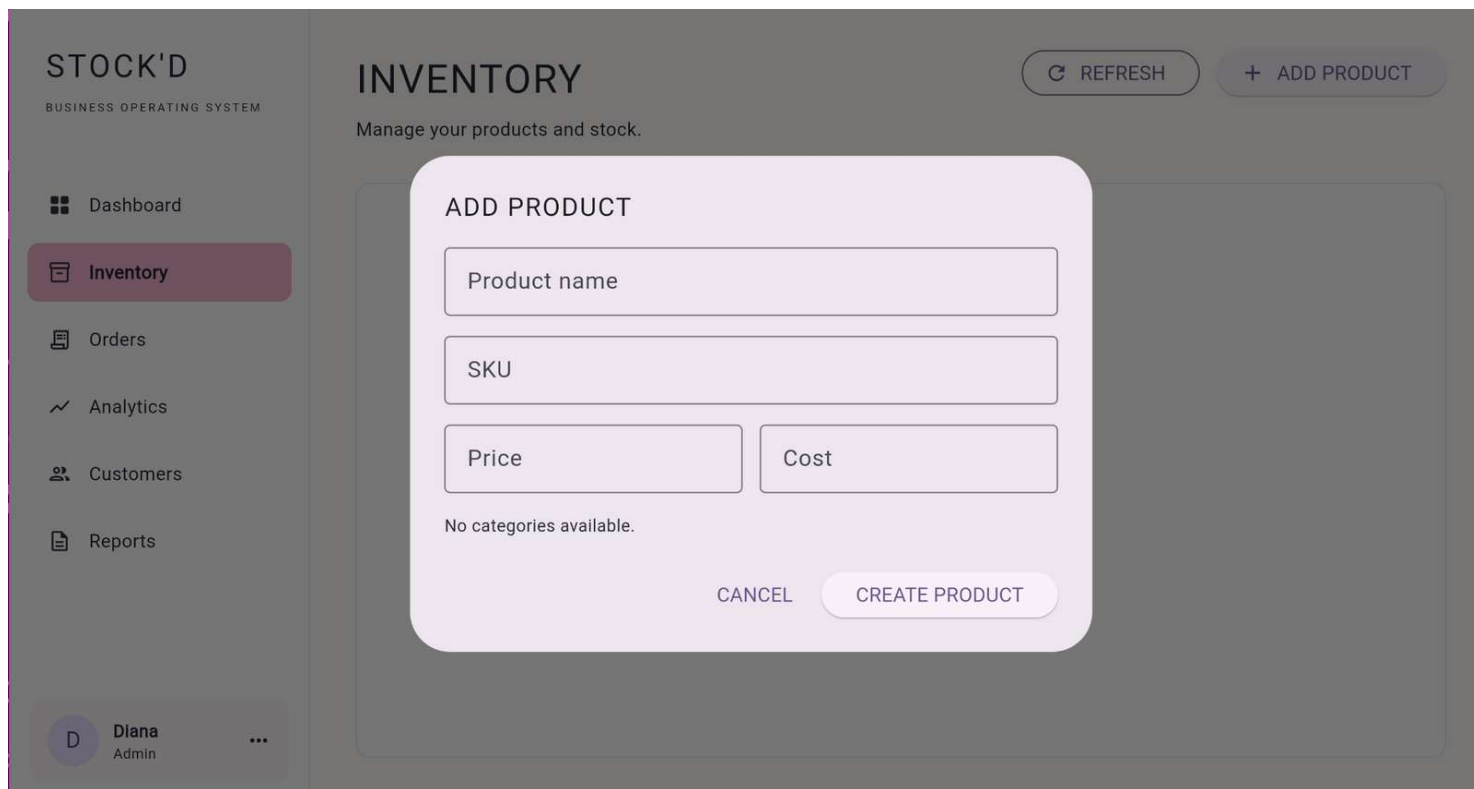
The dashboard became the central point of the application.

It was designed to give users an immediate overview of their business and provide quick access to important information.

The inventory screen was designed to make stock information easy to scan. It includes information such as active products, inactive products, low-stock items and products with no stock.

The orders screen connects customers, products and inventory so that business records can be managed together.

The customer section allows businesses to keep customer information alongside their order history.



07. Key Features

Inventory Management

Users can create products, manage stock quantities and set low-stock thresholds.

Products can also be deactivated when they are no longer being used while keeping their records within the system.

Orders

Users can create and manage orders using information from their products and customers.

This keeps order information connected to the rest of the business data.

Customers

Customer information can be stored and managed within Stockd, allowing businesses to keep their customer records organised.

Currency

Stockd includes currency selection and exchange-rate functionality, allowing the application to work with different currencies.

PDF Reports

Users can generate PDF reports containing business information, allowing important data to be taken outside of the application.

Notifications

Notifications help users stay aware of important information within their business, such as stock-related updates.

08. Technical Development

Stockd was developed as a cross-platform application using Flutter for the front end and C#/.NET for the backend.

The application uses a structure where the Flutter application communicates with a .NET API, which then communicates with the SQL Server database.

Flutter Application → .NET API → SQL Server Database

The backend handles areas such as authentication, businesses, products, inventory, customers, orders and currency functionality.

Entity Framework was used to work with the SQL Server database.

09. Authentication and Data

Authentication was implemented so that users can securely log into their account and access information associated with their business.

The application stores authentication information and uses the business ID to retrieve the correct data from the backend.

This means that different businesses can use the same application while keeping their information separated.

STOCK'D

BUSINESS OPERATING SYSTEM

Email

dimor43@moretti.com

Password

.....



10. Challenges

One of the biggest challenges during development was moving the application from a local development environment towards a real deployed application.

I encountered several technical issues involving:

- .NET publishing
- SQL Server connections
- Entity Framework migrations
- Authentication
- API deployment
- Hosting configuration
- Flutter and API communication
- Android builds

Debugging these problems required testing different parts of the system individually to identify where the issue was occurring.

This was an important part of the project because it helped me understand the difference between creating something that works locally and creating something that can actually be deployed and used by other people.



This page isn't working

stockd-api.runasp.net is currently unable to handle this request.

HTTP ERROR 500

Reload

11. Iteration

Stockd changed significantly throughout development.

The early versions focused mainly on making the core functionality work. Once the main features were implemented, I focused more on improving the visual design, navigation and overall user experience.

The development process became:

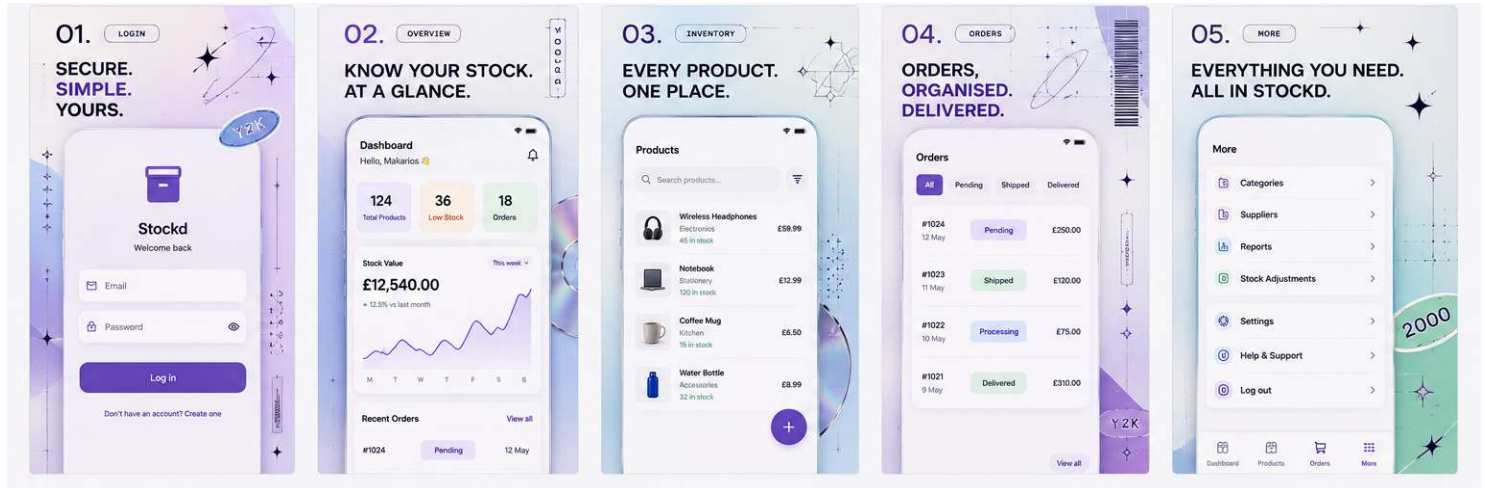
Functionality → Usability → Visual Design → Consistency → Deployment

I repeatedly revisited screens and improved them after seeing how they worked as part of the complete application.

12. Final Product

The final version of Stockd brings together products, inventory, orders, customers, business information, currency management, reports and notifications into one application.

The application is designed to give small businesses a central place to manage their stock and related business information.



13. What I Learned

Developing Stockd taught me that creating an application involves much more than designing individual screens.

Throughout the project I developed experience in:

- Flutter development
- C# and .NET
- API development
- SQL Server
- Entity Framework
- Authentication
- Database design
- Front-end and back-end integration
- Application deployment
- UI/UX design
- Debugging and problem solving

One of the most valuable things I learned was how all of these areas connect.

The design needs to work with the functionality, the application needs to communicate with the backend, the backend needs to communicate with the database, and the complete system needs to work outside of the development environment.

14. Reflection

Stockd started as an idea for a stock management application and developed into a working cross-platform product.

The project allowed me to experience the full development process, from the initial idea and visual identity through to interface design, programming, database integration, authentication and deployment.

It also taught me how important iteration is. Features that initially worked technically often needed to be redesigned or improved to make the overall experience clearer and easier to use.

The project has given me a much stronger understanding of how a real software product is created from start to finish.

15. Future Development

Although Stockd has reached a working stage, there are still opportunities for future development.

Potential future improvements include:

- More advanced business analytics
- Expanded reporting
- Additional customisation
- More detailed stock insights
- Improved collaboration between business users
- Additional business management features
- Further platform optimisation

16. Final Outcome

Stockd developed from a simple concept into a complete application combining design, software development, databases and deployment.

The project demonstrates my ability to take an idea, plan its features, design the user experience, build the application and solve technical problems throughout the development process.

Stockd — From concept, to design, to development, to deployment.

