Design – Supermarket FreshChoice BV case

Authors: Roko Mladinić, Daniella Namuli, Simeon Markov, Zed Minabowan & Kristiano Mizher

Introduction:

Commented [MR1]: @Mladinić,Roko R. me

The design phase builds on everything that was established during the analysis. With the requirements, priorities, and user stories now clearly defined, this phase focuses on shaping practical solutions that meet the needs of FreshChoice. The goal is to translate the earlier findings into structured designs that show how the system will function, how users will interact with it, and how the technical components will work together.

In this document, the ideas from the analysis are turned into visual layouts, system structures, data models, and workflow descriptions. These designs will guide development and ensure that everyone involved understands what is being built before any coding begins.

By the end of the design phase, there will be a complete blueprint that prepares the project for the realization phase where the actual implementation takes place.

Design Guide (UI/UX Principles)

This section is about the initial design plan (incorporating layout principles, accessibility, etc.)

- **User-Centric**
  The design focuses on customers who want a simple way to order groceries and staff who need fast access to inventory and schedules. Each page supports the main tasks of the user viewing it.

- **Visual Hierarchy**
  important information will be highlighted with the FreshChoice color style. Layout, spacing and text size should guide users through the page naturally.

- **Simplicity and Clarity**
  Pages will be clean and easy to understand. Only the information needed for the current page will be shown.

- **Consistency**
  All colors, fonts and buttons will have the same style as the FreshChoice

website. This makes the whole system logical and easy to use.

- **Appropriate Visualization Selection**
  Bar or line charts will be used when they make data easier to understand. For example, managers could see visuals that help compare stock or track changes.

- **Real-Time Data Presentation and Contextual Feedback**
  Users will get instant feedback when saving or editing something. Updated stock levels and clear messages will help to avoid any confusion.

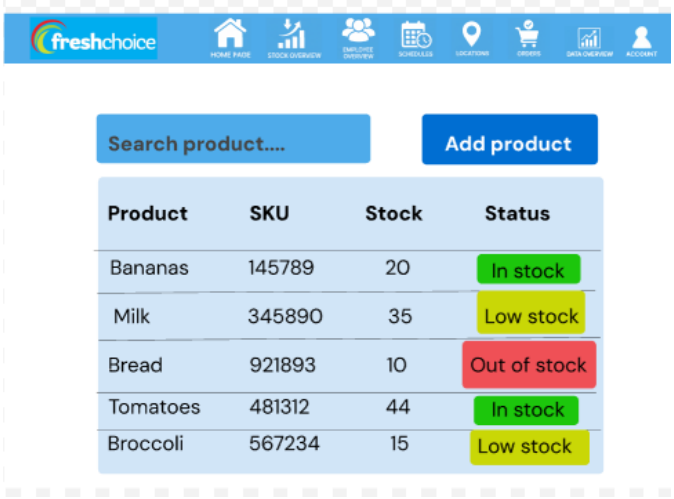
- **Big As Numbers (BANs)**
  The admin dashboard should include important data like stock totals or active orders in an easy to read feature. These numbers give managers a quick overview of the store at a glance.

Commented [ZM2]: Me

Wireframing

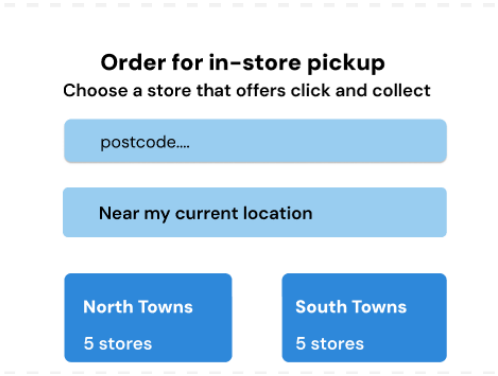
Stock Overview

The Stocks page is designed to show the availability of the products in the supermarket system. It helps managers see how many items are still in stock and whether products need restocking. The stocks page displays key fields such as product name, SKU, current stock amount and their status**.** The page uses a simple table layout for easy reading and a smooth workflow. This design ensures employees can understand stock levels at a glance and make fast decisions.



Locations

The Locations page displays the towns areas with Fresh Choice supermarkets where the customers can pick up the products they ordered online. Customers can also insert their postcode and find the nearest Fresh Choice supermarket.
The page uses a clean, consistent layout that matches the rest of the system.



Tech stack decision

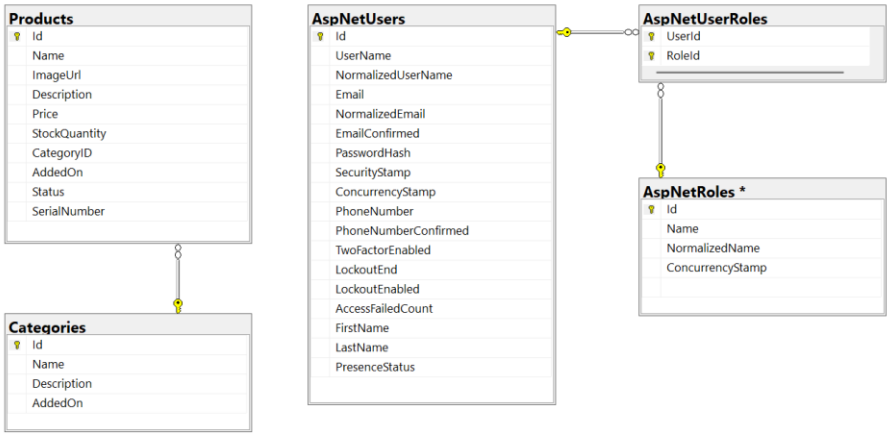
Commented [MS3]: @Markov,Simeon S. Task: Validate techstack decisions

- **Frontend:**
  1. **Bootstrap:**
     - Mobile-First Design: Designed to be responsive, components adjust to fit device screen size.
     - Pre-build components: Pre-build components allow building clean design faster than traditional CSS, coming up with components like navbar, modals, cards, etc.
     - Customizable: Can be overwritten by CSS.
     - Browser compatibility: Compatible with all modern browsers.
  2. **CSS:** Rich custom styling on components.
  3. **HTML:** Backbone of building documents structure layout.
- **Backend:**
  1. **Blazor (Blazor Server):** Recommended for Admin Dashboards. Used for Internal business tools, admin dashboards, real-time collaboration, applications with stable network connections. Its benefits are Fast initial load, smaller client footprint, server-side security, direct database access, minimal client resources.
  2. **JavaScript:** Used for Interacting with browser APIs not exposed to .NET, integrating JavaScript libraries and handling bootstrap interlop interactivity.
  3. **SQL Server:** It has fast queries procession, support for complex queries, and it's scalable, offering great ORM capabilities (This project uses the Entity Framework).

Database design

Commented [MS4]: @Markov,Simeon S. Task: Figure out the database design

As a RDBMS, SQL Server Management Studio was chosen for its compatibility with SQL Server and Microsoft ecosystem. It is performance-oriented and production ready graded.



Overview:

Entities:

Users: Users authentication table, serving the purpose of storing authenticated user alongside their profile data (name, email, password...).

Roles: Stores all the available roles within the application (e.g. manager, employee).

UserRoles: A pivot table for the many-to-many relationship for users and roles.

Products: The table contains all the necessary data for each individual product (e.g. name, image, price) and an established many-to-one relationship between **Categories** table (one category can have many products).

Category: The table stores data about each individual created category (e.g. name, description).

Tables description:

'Users'

Column	Type	Constraint	Description
Id	nvarchar(450)	NOT NULL, PRIMARY KEY	Unique identifier for each user record
UserName	nvarchar(256)	NULLABLE	Stores username record
NormalizedUserName	nvarchar(256)	NULLABLE	Stores normalized version of username
Email	nvarchar(256)	UNIQUE, NOT NULL	Stores emails records
NormalizedEmail	nvarchar(256)	NULLABLE	Stores normalized version of email
EmailConfirmed	bit	NOT NULL	Used for post-email confirmation, stores true/false
PasswordHash	nvarchar(MAX)	NOT NULL, UNIQUE	Stores the password hashed
SecurityStamp	nvarchar(MAX)	NOT NULL, UNIQUE	Stores a created security stamp for each user.
PhoneNumber	nvarchar(MAX)	NULLABLE, UNIQUE	Stores tel. of each user
PhoneNumberConfirmed	bit	NOT NULL	Stores true/false for confirmation
TwoFactorEnabled	bit	NOT NULL	Flag column for checking if 2fa is enabled
FirstName	nvarchar(MAX)	NOT NULL	Stores user's first name
LastName	nvarchar(MAX)	NOT NULL	Stores user's last name
PresenceStatus	nvarchar(MAX)	NULLABLE	Stores employee's presence status

Roles:

Column	Type	Constraint	Description
Id	nvarchar(450)	NOT NULL, PRIMARY KEY	Stores unique records for each role created
Name	nvarchar(256)	NULLABLE	Stores the name of the role
NormalizedName	nvarchar(256)	NULLABLE	Normalized version of the name

UserRoles:

Column	Type	Constraint	Description
UserId	nvarchar(450)	NOT NULL	References user's id from the Users table
RoleId	nvarchar(450)	NOT NULL	References roles's id from the Roles table

Products:

Column	Type	Constraint	Description
Id	int	PRIMARY KEY	Identifies each product record
Name	nvarchar(MAX)	NOT NULL	Stores product's name
ImageUrl	nvarchar(MAX)	NOT NULL	Stores the image URL of the product
Description	nvarchar(MAX)	NOT NULL	Stores product's description
Price	decimal(18, 2)	NOT NULL	Stores product's price
StockQuantity	int	NOT NULL	Stores product's quantity
CategoryID	int	FOREIGN KEY	References a category from table Categories
AddedOn	datetime2(7)	NOOT NULL	Stores the date the product was added
Status	nvarchar(MAX)	NOT NULL	Stores the product's status (e.g. available, not in stock)

SerialNumber	nvarchar(MAX)	NOT NULL	Stores the product's serial number

Categories:

Column	Type	Constraint	Description
Id	int	PRIMARY KEY	Uniquely identifies each category record
Name	nvarchar(MAX)	NOT NULL	Stores category name
Description	nvarchar(MAX)	NOT NULL	Stores category's description
AddedOn	datetime2(7)	NOT NULL	Stores the date when the category was added

References

ANIRUDDHA ADAK (2014). Bootstrap: The Essential Front-End Framework. https://dev.to/aniruddhaadak/bootstrap-the-essential-front-end-framework-515f

Davor Bursać (2025). Blazor Server vs Blazor WebAssembly: Choose the right one for your project. https://wearenotch.com/blog/blazor-server-vs-blazor-webassembly/

Cheyenne Sokkappa (2025). Blazor Server vs. WebAssembly: Which Works Best For You? https://www.gapvelocity.ai/blog/blazor-server-vs-webassembly

Telerik. When to Use Blazor WebAssembly vs. Blazor Server? https://www.telerik.com/faqs/blazor/blazor-webassembly-vs-server

Microsoft Documentation (2025). ASP.NET Core Blazor JavaScript interoperability (JS interop). https://learn.microsoft.com/en-us/aspnet/core/blazor/javascript-interoperability/?view=aspnetcore-9.0

Saravanan G (2025). Pros and Cons of Using JavaScript Interop in Blazor. https://www.syncfusion.com/blogs/post/pros-cons-javascript-interop-blazor

Siva Kumar Raju Bhupathiraju (2025). Key Features and Innovations in SQL Server 2025: Advancing Performance, Security, and AI Integration. https://www.ijsat.org/papers/2025/1/2493.pdf

Abhishek Nag (2025). How to Build a RESTful API with ASP.NET Core, C# for Real-World Applications? https://embarkingonvoyage.com/blog/technologies/restful-api-with-asp-net-core/

Imtiaj Hossain (2025). Why SQL Server 2025 is a game changer for modern data platforms. https://novadba.com/sql-server-2025-game-changer/

AI Transparency: Perplexity AI for in-depth research, articles summarization.

Conclusion:

Our design document gave showed our plan for how our FreshChoice website will look and work. We turned the ideas from the analysis into layouts, UI choices, system structures and a database design that fit the needs of our project. With this design finished, we now have everything we need to start building the actual website in the realization phase.