**Software Design Specification**

for

# Supermarket Management System (SMS)

Version 1.0

Prepared by  
MHD HADI SAID ALKADDOUR  
June 14, 2025

# Table of Contents

ii. Table of Contents   
1. Introduction   
2. System Architecture   
 2.1 High-Level Architecture Diagram   
 2.2 Architecture Narrative   
3. GUI Components with Layouts and Navigation   
4. System Components   
 4.1 Class Diagram   
 4.2 Use Case Sequence Diagrams   
5. Design Decisions and Tradeoffs   
6. Appendices

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Mhd Hadi Said Alkaddour | 14/06/2025 | Initial Creation | 1.0 |

# 1. Introduction

## 1.1 Purpose of Document

This Software Design Specification (SDS) outlines the design architecture and system components of the Supermarket Management System (SMS), a desktop-based application developed in C# using Windows Forms. This document serves as a guide for developers during implementation and testing phases.

## 1.2 Scope of the Development Project

The SMS project includes modules such as Inventory Management, Point of Sale (POS), Reporting, Profile Management, and Admin Dashboard. It uses SQLite for local data storage and DevExpress for UI enhancement.

## 1.3 Definitions, Acronyms, and Abbreviations

POS: Point of Sale  
UI: User Interface  
DB: Database  
CRUD: Create, Read, Update, Delete  
PDF: Portable Document Format

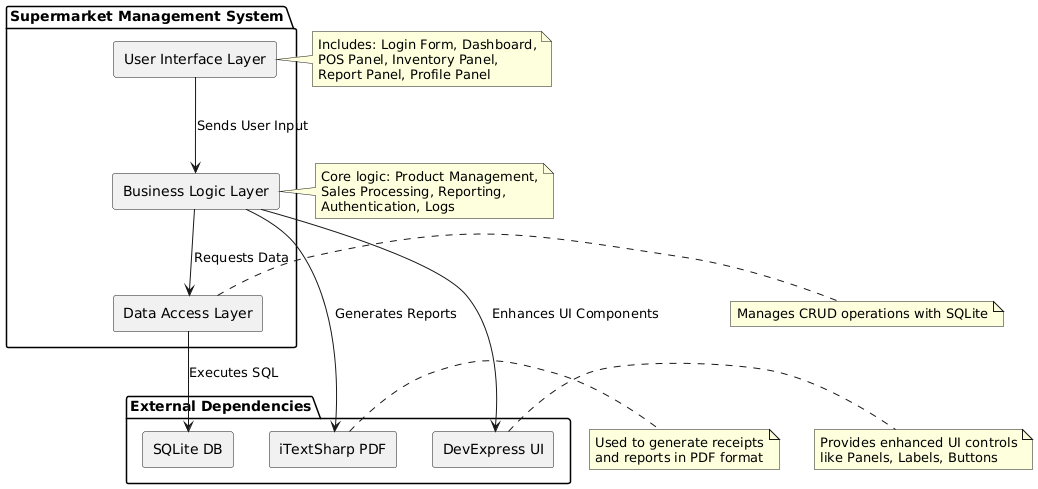
## 1.4 Overview of Document

This SDS focuses on the system architecture, user interface layout, class diagram, use case sequence diagrams, and design decisions made during development. It provides a technical blueprint to support consistent implementation and future maintenance.

# 2. System Architecture

## 2.1 High-Level Architecture Diagram

A visual representation of the system layers is shown below:



## 2.2 Architecture Narrative

The system follows a layered architecture pattern consisting of:  
- \*\*Presentation Layer\*\*: Windows Forms UI built with DevExpress controls.  
- \*\*Business Logic Layer\*\*: Core logic including inventory management, POS operations, reporting, and authentication.  
- \*\*Data Access Layer\*\*: SQLite database access layer handling CRUD operations.  
  
The system does not rely on external APIs or network services, operating entirely locally. The UI communicates directly with business logic classes, which interact with the database via SQLite commands.

# 3. GUI Components with Layouts and Navigation

## 3.1 User Interface Issues

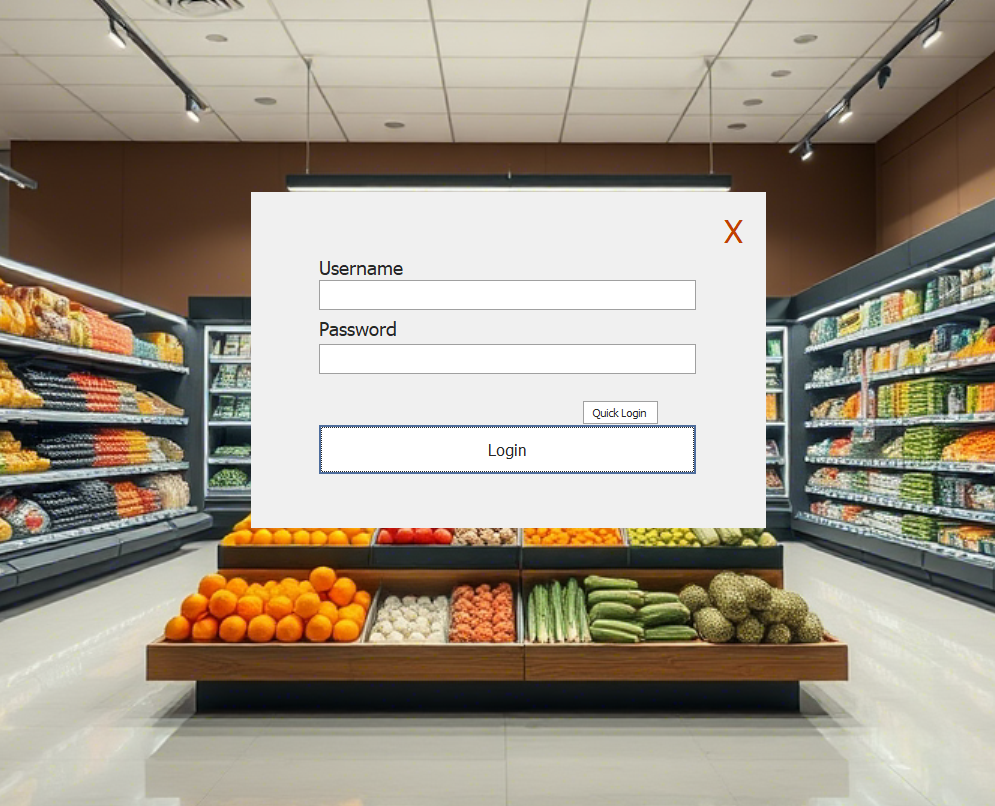
The primary users are supermarket staff who require an intuitive and efficient interface for daily operations. All screens are designed to be navigable without training. Buttons and actions are clearly labeled. The system supports keyboard input and mouse interaction.

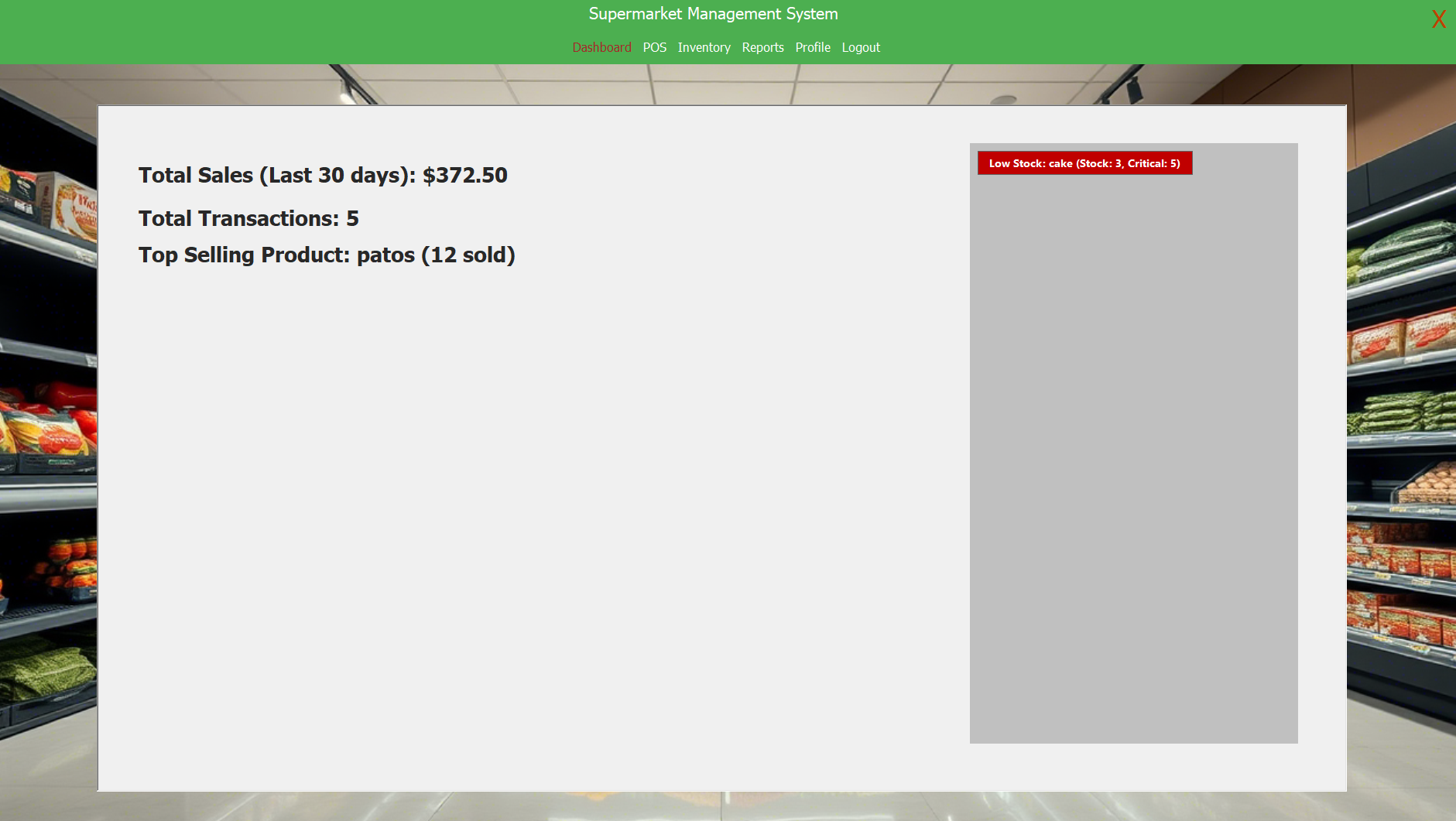
## 3.2 List of User Interface Screens

- SS1 - Login Screen   
- SS2 - Dashboard   
- SS3 - Inventory Management   
- SS4 - Point of Sale (POS)   
- SS5 - Sales Reports   
- SS6 - Profile Management

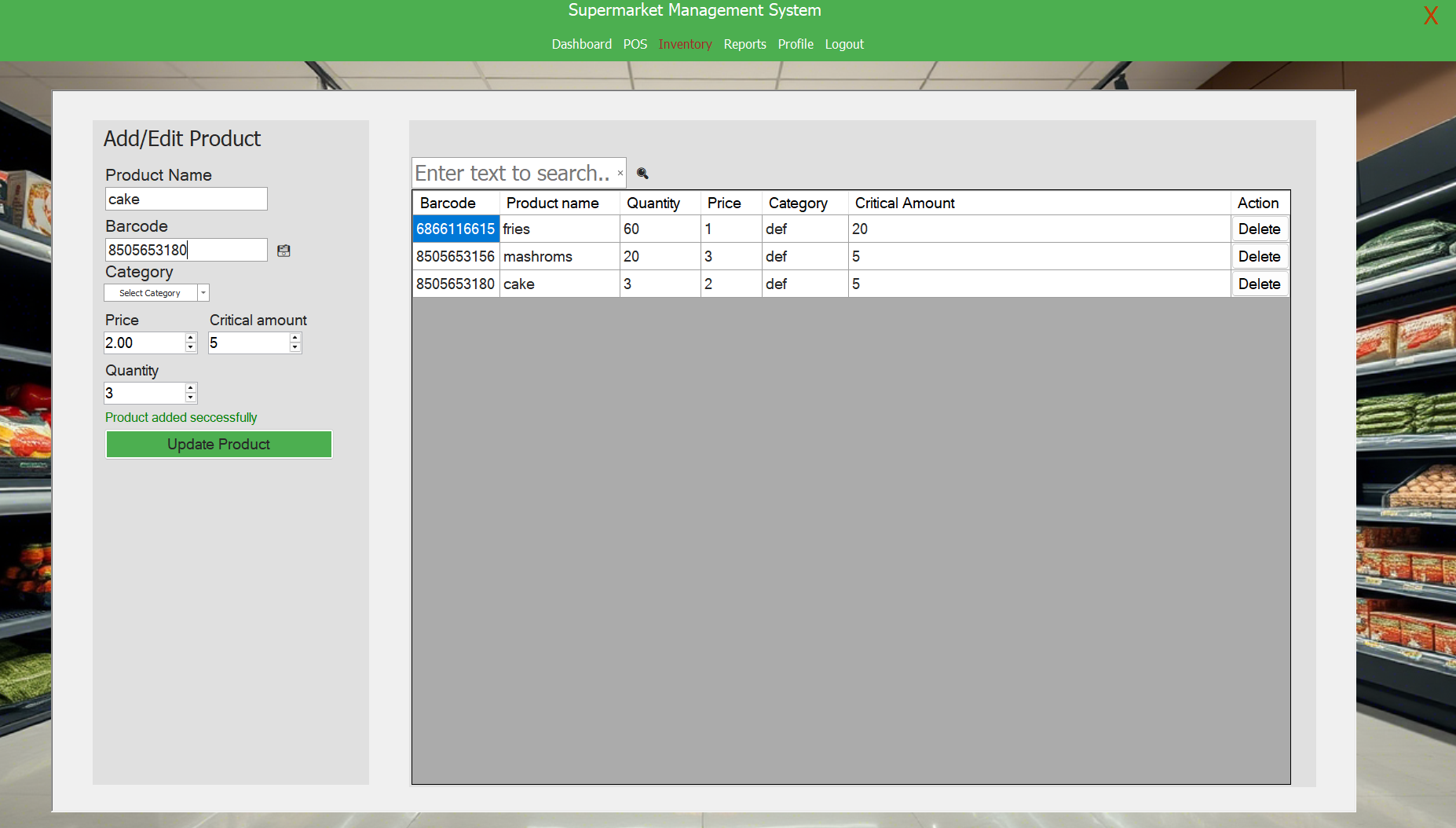
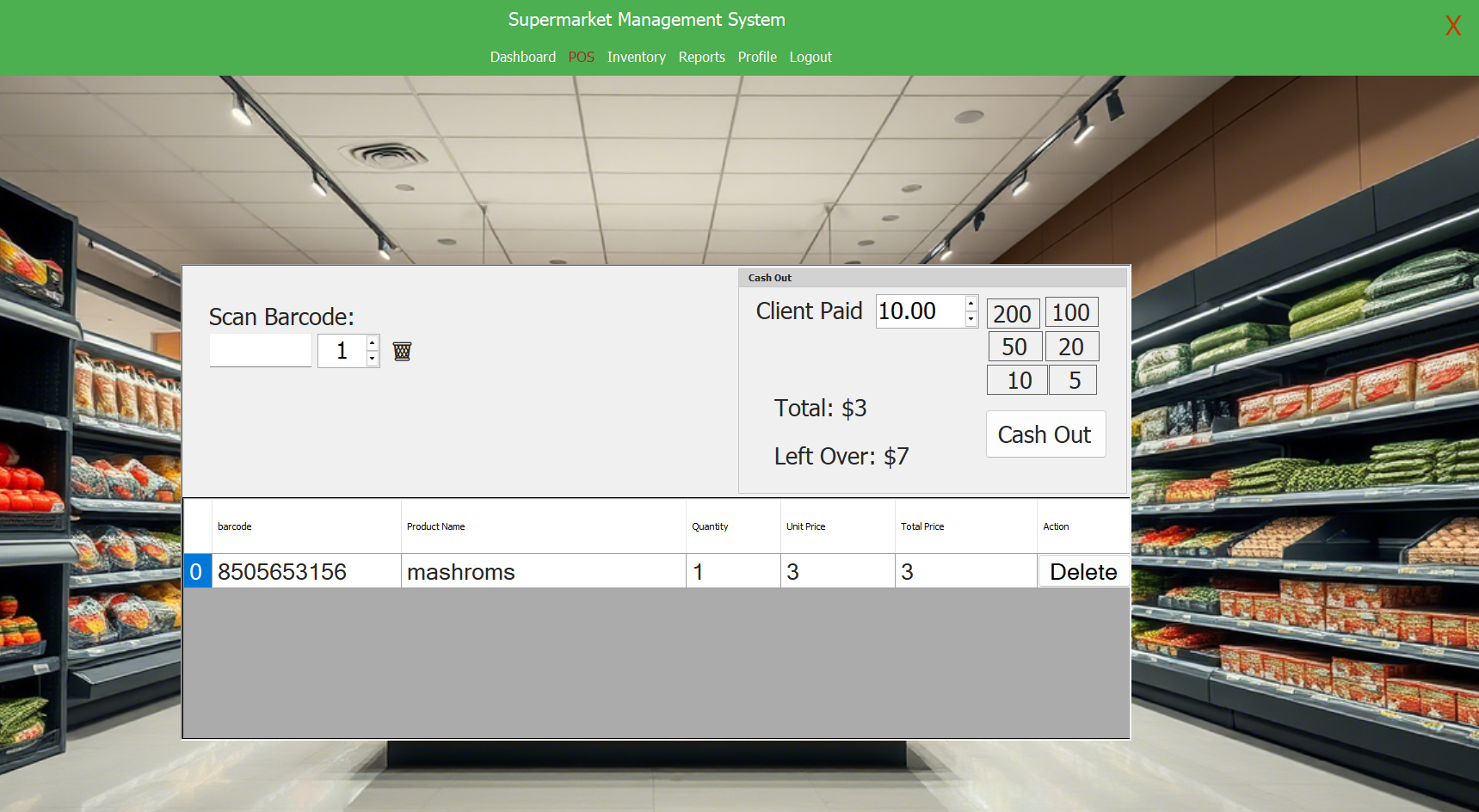
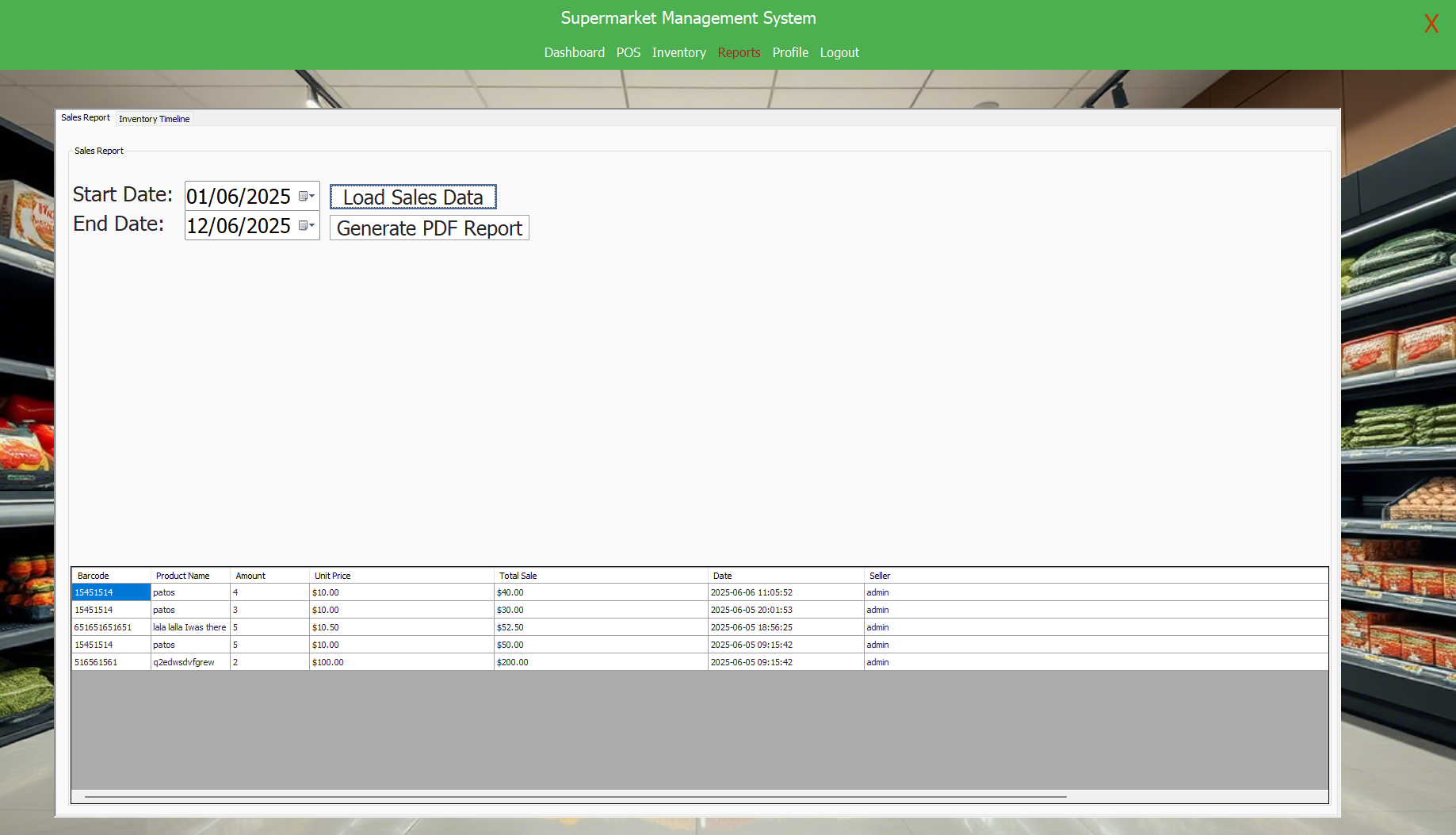
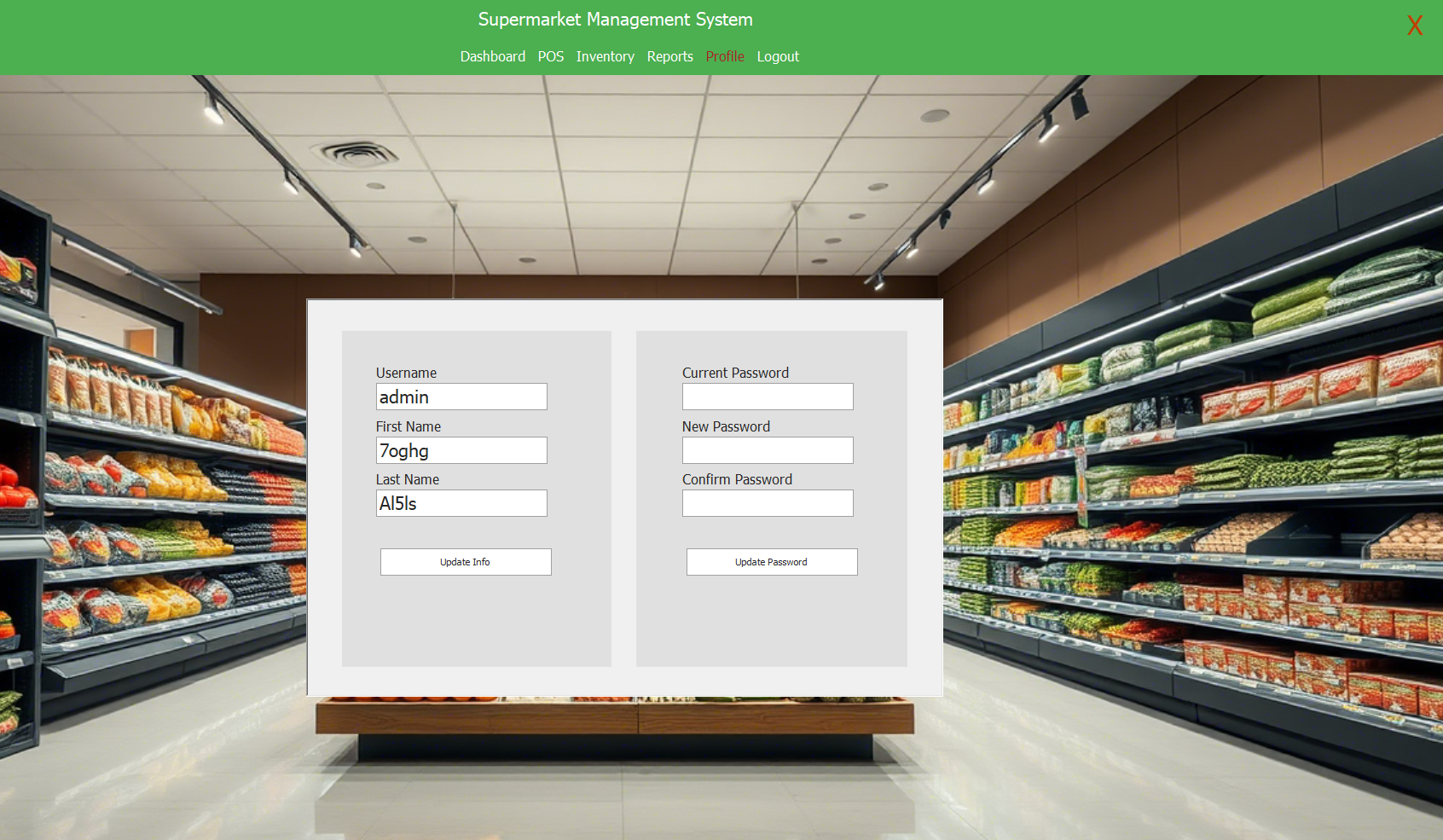
## 3.3 User Interface Screens

SS1: Login Screen – Username and password entry

  
SS2: Dashboard – Navigation bar for all modules

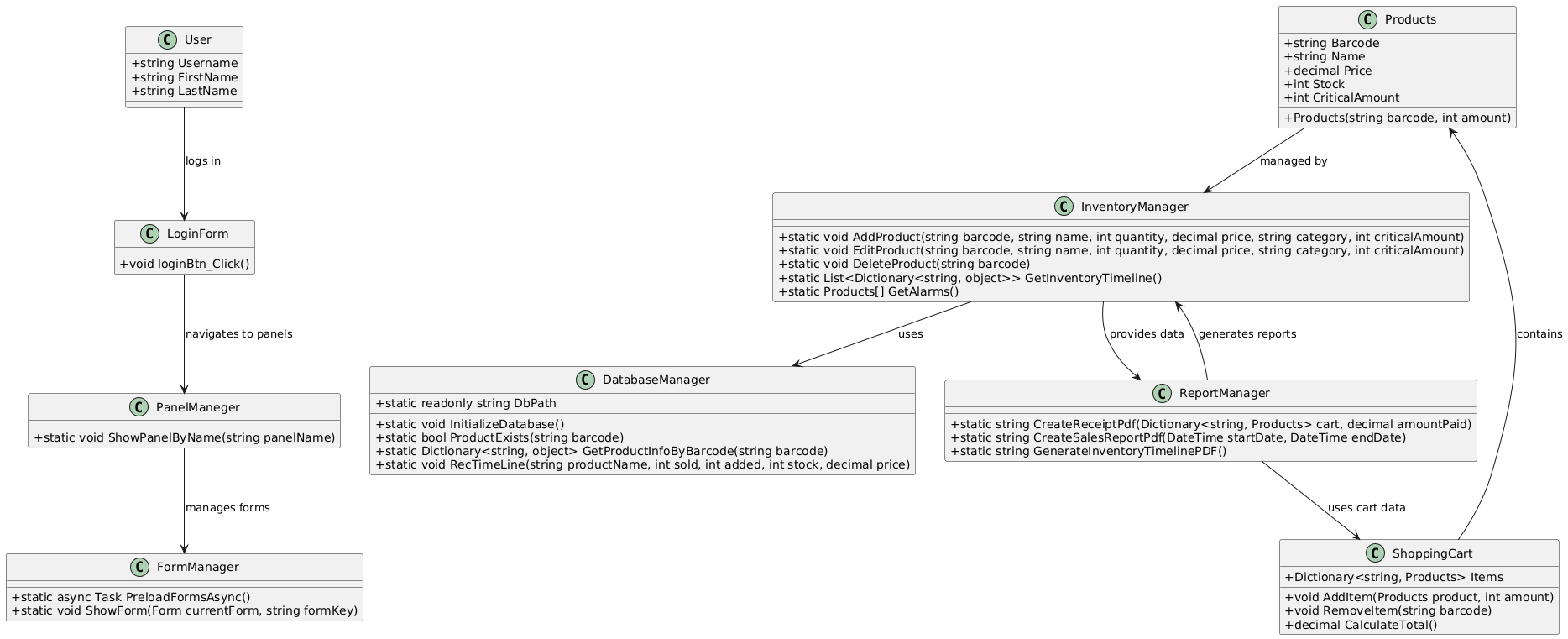


SS3: Inventory Management – Product table with search, edit, delete options

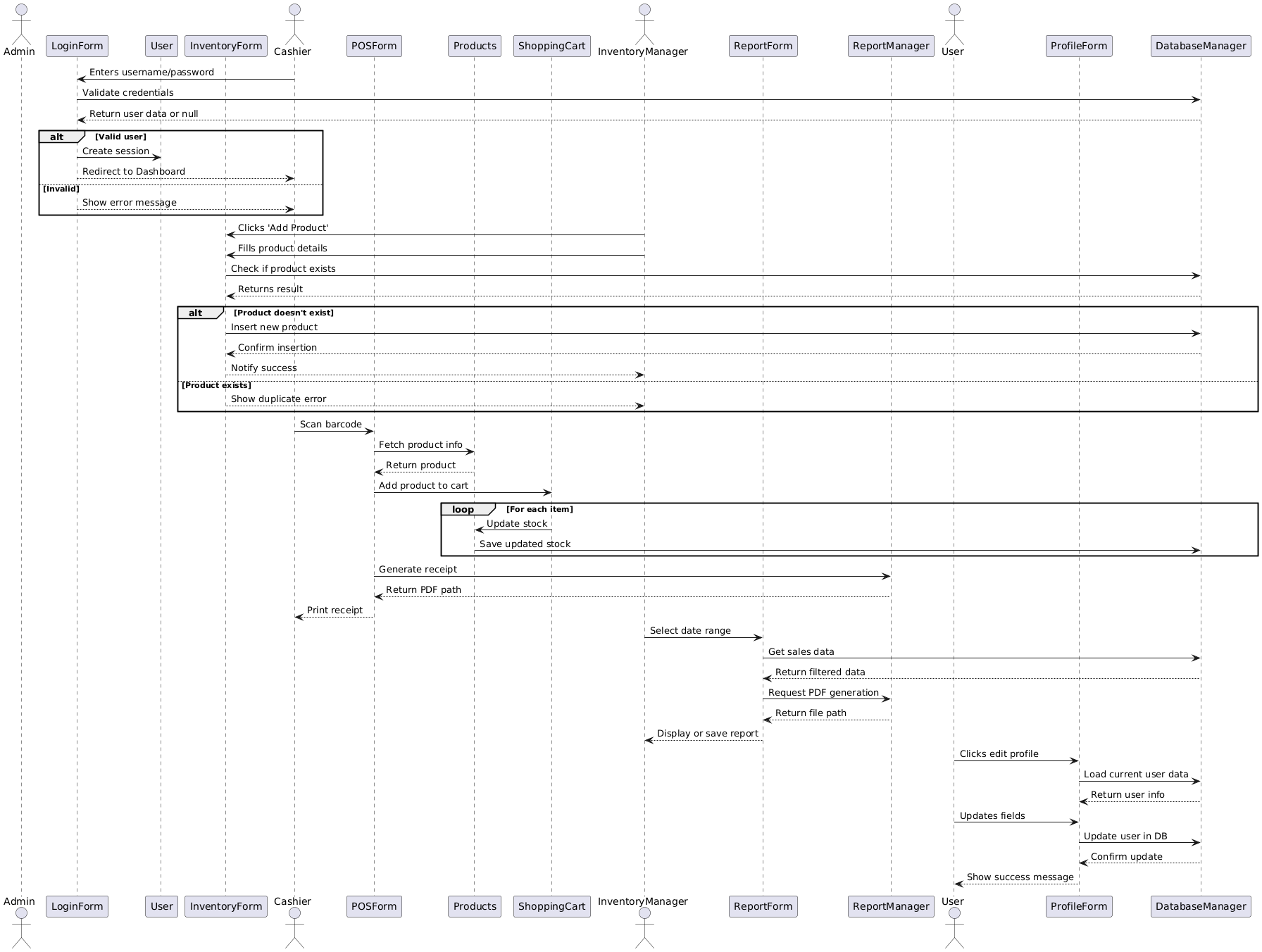
  
SS4: POS – Barcode input, cart display, checkout button   
SS5: Sales Reports – Date filters, sales table, PDF export   
SS6: Profile Management – Edit name, update password 

# 4. System Components

## 4.1 Class Diagram



## 4.2 Use Case Sequence Diagrams

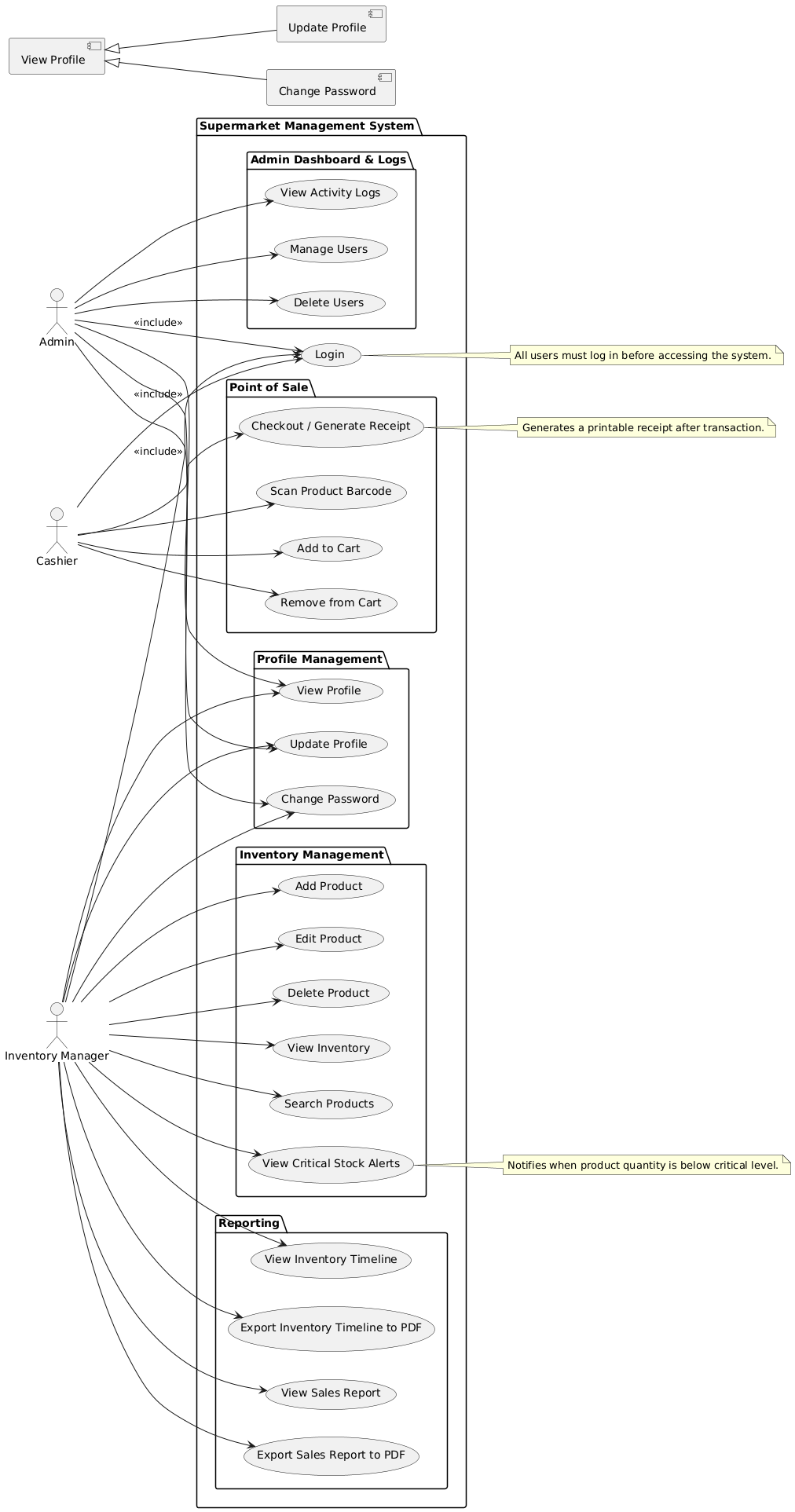


# 5. Design Decisions and Tradeoffs

Key design decisions include:  
- \*\*Local-only Data Storage\*\*: SQLite was chosen over cloud databases for simplicity and offline usage.  
- \*\*C# and Windows Forms\*\*: Selected for ease of integration with existing libraries and fast development cycle.  
- \*\*DevExpress UI Library\*\*: Used for advanced UI features but increases deployment complexity.  
- \*\*No External Dependencies\*\*: Avoided internet connectivity to ensure reliability in offline environments.

# 6. Appendices

## Appendix A: Use Case Diagrams



## Appendix B: Data Dictionary

Detailed definitions of data entities used in the system.

User:  
- id (int, PK)  
- username (string)  
- firstname (string)  
- lastname (string)  
- password (string)  
  
Product:  
- barcode (string, PK)  
- product\_name (string)  
- quantity (int)  
- price (decimal)  
- category (string)  
- critical\_amount (int)

## Appendix C: List of Inputs and Outputs

Inputs come from user interaction: typing into fields, clicking buttons, scanning barcodes.

Outputs include screen updates, receipt printing, and PDF reports.