# EzBuy Project Documentation

## 1. Introduction

EzBuy is a mobile shopping application built with Flutter. The main idea behind the app is simple: give users a clean, easy, and pleasant way to browse products and make purchases directly from their phones. Instead of relying on heavy or complicated e-commerce platforms, EzBuy focuses on being lightweight, intuitive, and fast.

To power the app, Firebase Firestore is used as the database and Firebase Authentication handles secure sign-ins and account management. Users can explore products, look at detailed descriptions, add items to their cart, save favorites, and go through a straightforward checkout process.

This project represents hands-on experience with Flutter development, Firebase integration, mobile UI/UX, and proper code organization using the BLoC state management pattern.

## 2. Project Objectives

EzBuy aims to achieve the following goals:

• Build a fully functioning, user-friendly mobile shopping app.

• Use Firebase Authentication and Firestore to support secure login and real-time data storage.

• Follow clean architecture principles with modular and reusable code.

• Include essential e-commerce features: product browsing, cart management, favorites, checkout, and order history.

• Apply BLoC for smooth and predictable state management.

• Keep the interface simple, attractive, and easy to navigate.

## 3. Problem Statement

Many small businesses and individual sellers want to take advantage of mobile shopping platforms, but building a custom app is usually expensive and complicated. Existing solutions can be overly complex or not tailored to their needs.

EzBuy attempts to bridge that gap. The app offers a straightforward, ready-to-use structure that can easily be adapted or expanded. It keeps things simple for users while still being scalable and backed by reliable cloud services.

## 4. Requirements Gathering

### 4.1 Stakeholder Analysis

• End Users: A smooth shopping experience, secure login, and a clean interface.

• Developers: A modular codebase that is easy to maintain and improve.

• Firebase Services: Secure authentication and scalable cloud data handling.

### 4.2 User Stories & Use Cases

User Stories:

• “I want to browse products quickly so I can find items I like.”

• “I want to add things to my cart so I can buy them later.”

• “I want to save my favorite products so I don’t lose track of them.”

Use Case Example – Login:

The user enters their email and password. Firebase checks the credentials and, if valid, the user is taken to the home screen.

### 4.3 Functional Requirements

• User authentication (sign up, log in, password reset)

• Product browsing and product detail viewing

• Cart management (add, remove, update quantity)

• Favorites management

• Checkout and order history storage

• Profile editing and password updates

### 4.4 Non-Functional Requirements

• Fast loading times

• Secure authentication and data storage

• Simple and intuitive UI

• Real-time updates using Firestore streams

• Scalability supported by Firebase

5. How the Application Works (User Flow)

1. Users sign up or log in.

2. The home screen displays available products.

3. Users browse and open product details.

4. Items can be added to favorites or cart.

5. The cart allows quantity updates and item removal.

6. Checkout confirms the order and stores it in history.

7. Purchase history displays past orders.

8. Users can manage their profile and password.

6. System Overview

EzBuy is organized into screens, services, models, and BLoC components:

• Screens manage UI and interactions.

• Services handle Firestore communication.

• Models structure the data.

• BLoC manages app state.

• Firebase powers authentication and storage.

7. System Analysis & Design

7.1 Use Case Diagram (Description)

The main user can log in, browse products, view details, add to cart, add to favorites, check out, and update their profile.

7.2 Database Design & Data Modeling

Firestore Structure:

users/userId/cart/

users/userId/favorites/

users/userId/orders/

products/productId/

ER Model Summary:

• A user can have multiple cart items.

• A user can save many favorites.

• A user can complete multiple orders.

• Orders can contain several products.

7.3 Data Flow Diagram (Description)

High-level flow:

User → App → Firebase → App → User

Cart example:

The user adds an item, the app sends it to Firestore, Firestore updates the data, and the UI refreshes instantly.

7.5 UI/UX Design

The design follows Material Design principles:

• Home page uses a product grid.

• Product page highlights images and descriptions.

• Cart page shows selected items and totals.

• Profile page allows editing user information.

8. Technology Stack

• Flutter & Dart

• Firebase Authentication

• Firestore Database

• Firebase Storage

• BLoC for state management

• Material Design UI

9. Firebase Integration

Firebase Authentication:

Handles login, signup, and password updates.

Firestore:

Stores product data, favorites, cart items, and order history. Each user’s data is isolated and secure.

10. Implementation Details

Authentication:

• Handled using FirebaseAuth with proper validation.

Product Handling:

• Products displayed in real time from Firestore.

Cart System:

• Each user has a personal cart with quantities and totals.

Favorites:

• Stored independently for each user.

Checkout:

• Orders saved to purchase history and cart cleared afterward.

State Management (BLoC):

Ensures predictable state transitions.

Profile Management:

Users can update personal info and change their password.

11. Challenges and Solutions

• Managing UI states → Solved with BLoC.

• Keeping data updated → Solved using Firestore streams.

• Ensuring authentication safety → Handled through FirebaseAuth.

• Maintaining clean structure → Achieved with modular architecture.

12. Future Improvements

• Add real payment gateway

• Add order tracking system

• Enhance animations and transitions

13. Conclusion

EzBuy is a practical and polished mobile e-commerce application built with Flutter and powered by Firebase. It includes all essential features—authentication, product browsing, favorites, cart, checkout, and order history—while maintaining clean structure and scalability. This project serves as a strong foundation for future growth and additional features.