تحليل مشروع الأول برو المحاسبي

تقرير تحليل شامل للهيكل والكود والوظائف

المشروع: FirstProAccounting

الإصدار: 2.0.0 (Build 1.0.10)

المنصة: Android - Flutter/Dart

تاريخ التحليل: مايو 2026

المحلل: Z.ai - تحليل آلي

نظرة عامة على المشروع

معلومات أساسية وتعريف بالمشروع وتقنياته المستخدمة

مشروع "الأول برو المحاسبي" هو تطبيق محاسبي متكامل مخصص لمنصة أندرويد، تم تطويره باستخدام إطار عمل Flutter ولغة Dart. يستهدف التطبيق أصحاب المتاجر والمؤسسات الصغيرة والمتوسطة في اليمن والمنطقة العربية، حيث يوفر واجهة عربية كاملة مع دعم الاتجاه من اليمين لليسار (RTL). يتميز التطبيق بتصميم حديث يعتمد على نظام Material 3 مع سمة لونية تعتمد على التدرجات الزرقاء البنفسجية والذهبية.

60	23,443	14
ملف Dart في المكتبة الأساسية	سطر كود إجمالي الكود المصدري	جدول قاعدة بيانات في SQLite

التعريف بالمشروع

الأول برو المحاسبي هو نظام محاسبي شامل يعمل بدون اتصال بالإنترنت (Offline-First)، حيث يعتمد على قاعدة بيانات SQLite محلية لتخزين جميع البيانات. يوفر التطبيق إدارة كاملة للمبيعات والمشتريات، والعملاء والموردين، والمنتجات والمخزون، والصناديق والبنوك، والمصروفات والموظفين، مع دعم متعدد العملات (الريال اليمني، الريال السعودي، الدولار الأمريكي) ونظام مزدوج المحاسبة مع دليل الحسابات وقيد اليومية.

التقنيات المستخدمة

- Flutter 3.x
- Dart ^3.11.5
- SQLite (sqflite)
- Material 3
- Provider
- RTL Arabic
- Cairo Font
- SVG Icons
- Phosphor Icons

التبعيات الرئيسية

الحزمة	الإصدار	الغرض
sqflite	^2.4.1	قاعدة بيانات SQLite المحلية
provider	^6.1.4	إدارة الحالة المركزية

الحزمة	الإصدار	الغرض
flutter_svg	^2.0.17	عرض الرسوميات المتجهة SVG
phosphor_flutter	^2.1.0	مكتبة أيقونات حديثة
intl	^0.20.2	تنسيق التواريخ والأرقام
uuid	^4.5.1	توليد معرفات فريدة للفواتير
image_picker	^1.0.4	التقاط واختيار الصور
path_provider	^2.1.1	مسارات تخزين الملفات

البنية البرمجية

تنظيم المشروع والبنية المعمارية وأنماط التصميم المستخدمة

طبقة العرض (UI Layer)
الشاشات، الويدجتس، التنقل، السقالة الرئيسية - 30+ ملف Dart

⬇

طبقة الأعمال (Core Layer)
الثوابت، السمات، المنسقون، نظام التصميم، الإضافات

⬇

طبقة البيانات (Data Layer)
نماذج البيانات، مساعد قاعدة البيانات، عمليات CRUD

هيكل المجلدات

المسار	الوصف	عدد الملفات
/lib/core/constants	الثوابت والتكوينات	1
/lib/core/theme	السمات والألوان ونظام التصميم	3
/lib/core/utils	أدوات تنسيق العملات والتواريخ	2
/lib/core/extensions	إضافات السياق	1
/lib/data/datasources	مساعد قاعدة البيانات	1
/lib/data/models	نماذج البيانات	10
/lib/ui/navigation	التوجيه والسقالة الرئيسية	2
/lib/ui/screens	شاشات التطبيق	25+ شاشة
/lib/ui/widgets	مكونات واجهة المستخدم	9

نمط البنية المعمارية

يتبع المشروع نمطاً معمارياً بسيطاً قريباً من نمط MVP (Model-View-Presenter) بدون طبقة عرض تقديمي رسمية. يتم التعامل المباشر مع قاعدة البيانات من الشاشات عبر Singleton Pattern في DatabaseHelper. لا يوجد فصل صارم بين منطق الأعمال وطبقة العرض، حيث تحتوي الشاشات على منطق الأعمال مدمجاً فيها مباشرة. هذا النمط مناسب للتطبيقات الصغيرة لكنه قد يصعب الصيانة مع نمو المشروع.

نمط Singleton لقاعدة البيانات

يستخدم DatabaseHelper نمط Singleton لضمان وجود نسخة واحدة فقط من اتصال قاعدة البيانات طوال دورة حياة التطبيق. هذا يمنع تسرب الذاكرة ويضمن الاتساق في العمليات، مع استخدام آلية Lazy Initialization لعدم فتح قاعدة البيانات إلا عند الحاجة الفعلية.

- **نظام التنقل**

يعتمد التطبيق على نظام تنقل بسيط يتكون من: شريط تنقل سفلي مخصص (5 علامات تبويب: الرئيسية، العملاء، الفواتير، التقارير، المزيد)، وقائمة جانبية (Drawer) تحتوي على اختصارات لجميع الأقسام، وتوجيه مسمى (Named Routes) عبر AppRouter مع أكثر من 20 مساراً مسجلاً. يستخدم IndexedStack للحفاظ على حالة الشاشات عند التبديل بين التبويبات.

قاعدة البيانات

المخطط الكامل لجداول SQLite والعلاقات والفهارس مع آلية الترحيل

يستخدم التطبيق قاعدة بيانات SQLite محلية بالإصدار 8 مع نظام ترحيل متدرج يضمن تحديث المخطط عند كل ترقية. تشمل قاعدة البيانات 14 جدولاً رئيسياً مع 19 فهارساً لتحسين الأداء، وبيانات أولية مزروعة للعملات والحسابات الافتراضية.

accounts (الحسابات)	12 حقل
id PK	INTEGER AUTOINCREMENT
name_ar, name_en	TEXT
account_code	TEXT NOT NULL
account_type	TEXT (ASSET/LIABILITY/COST/REVENUE/EXPENSE)
currency	'TEXT DEFAULT 'YER
balance, debt_ceiling	REAL
parent_id FK	INTEGER → accounts.id

invoices (الفواتير)	22 حقل
id PK	TEXT (UUID)
type	TEXT (sale/purchase/sale_return/purchase_return)
payment_mechanism	TEXT (cash/credit)
payment_method	TEXT (cash/check/transfer/bank/ewallet)
currency, exchange_rate	TEXT, REAL
customer_id, supplier_id FK	INTEGER

products (المنتجات)	25 حقل
id PK	INTEGER AUTOINCREMENT
item_code, barcode	TEXT
cost_price, sell_price, wholesale_price	REAL
current_stock, min_stock	REAL
category_id, supplier_id, warehouse_id FK	INTEGER

آلية الترحيل (Migration)

يدعم التطبيق 8 إصدارات من قاعدة البيانات مع نظام ترحيل تدريجي. كل إصدار يضيف جداول أو أعمدة جديدة مع معالجة الأخطاء عبر try-catch لمنع تعطل التطبيق. يتم زرع بيانات افتراضية تلقائياً عند الإنشاء الأول، بما في ذلك 3 عملات (YER, SAR, USD) و36 حساباً محاسبياً (12 حساباً لكل عملة).

جداول إضافية

الجدول	الوصف	الحقول الرئيسية
customers	العملاء	name, phone, balance, currency, credit_limit
suppliers	الموردين	name, phone, balance, balance_type, currency
cash_boxes	الصناديق والبنوك	name, type, balance, linked_account_id
invoice_items	عناصر الفاتورة	invoice_id, product_id, quantity, unit_price
transactions	القيود المحاسبية	account_id, debit, credit, journal_id
expenses	المصروفات	title, amount, category, operation_type
employees	الموظفين	name, job_title, balance, account_id
currencies	العملات	code, exchange_rate, is_default
categories	التصنيفات	name, parent_id, icon, color
warehouses	المستودعات	name, location, is_active
users	المستخدمين (الكاشير)	username, role, is_active

الوظائف والميزات

تحليل شامل للوحدات الوظيفية وشاشات التطبيق

نظام الفواتير
إنشاء فواتير البيع والشراء مع دعم المرتجعات، خصومات، ضرائب، شحن، و6 طرق دفع تشمل المحافظ الإلكترونية والتحويلات المصرفية

نقطة البيع (POS)
واجهة بيع سريعة مع بحث الباركود، سلة مشتريات تفاعلية، ودعم الورديات (Shifts) لإدارة نوبات العمل

إدارة المنتجات
إدارة كاملة للأصناف مع باركود، كود صنف، أسعار متعددة (تجزئة/جملة/جملة خاصة)، تصنيفات، وتتبع مخزون

العملاء والموردون
إدارة العملاء والموردين مع أرصدة، حدود ائتمانية، عملات متعددة، وتتبع الديون

الصناديق والبنوك
إدارة صناديق النقدية والحسابات البنكية مع ربطها بالحسابات المحاسبية وتتبع الأرصدة

المصروفات
تسجيل المصروفات مع تصنيفات، حسابات مصروفات، عمليات صرف واستلام، ومرفقات

دليل الحسابات
شجرة حسابات محاسبية كاملة (أصول، خصوم، تكاليف، إيرادات، مصاريف) مع قيود يومية مزدوجة

تعدد العملات
دعم 3 عملات (ريال يمني، سعودي، دولار) مع أسعار صرف وحسابات محاسبية مستقلة لكل عملة

نظام القيود المحاسبية (Double-Entry)

يتميز التطبيق بتطبيق نظام القيد المزدوج تلقائياً عند حفظ الفواتير. كل عملية بيع أو شراء تولد قيوداً محاسبية تلقائية في الحسابات المناسبة (المبيعات، المشتريات، العملاء، الموردين، الصناديق) حسب نوع الفاتورة وآلية الدفع، مع دعم عمولات النقل كقيود إضافية مستقلة. كل قيد يحصل على journal_id موحد لربط الأطراف المدينة والدائنة.

الشاشات الرئيسية (25+ شاشة)

الفئة	الشاشات
الرئيسية	لوحة التحكم، الإحصائيات
المبيعات	نقطة البيع، إنشاء فاتورة بيع، إنشاء فاتورة شراء

الفئة	الشاشات
الإدارة	المنتجات، العملاء، الموردين، المستودعات، الموظفين
المالية	الصناديق، المصروفات، العملات، دليل الحسابات
التقارير	التقارير، التدقيق المحاسبي، ميزان المراجعة
النظام	الإعدادات، الدعم الفني

تصميم واجهة المستخدم

نظام السمات والألوان والمكونات وتجربة المستخدم

نظام التصميم (Design System)

يعتمد التطبيق على نظام تصميم شامل مُعرَّف في ملف design_system.dart يتضمن مسافات موحدة (4-64 بكسل)، أنظمة حركة (قصير 200ms، متوسط 350ms، طويل 500ms)، وأبعاد ثابتة للبطاقات والأزرار والحقول. يتبع التطبيق معايير Material 3 مع تخصيص كامل لجميع المكونات بما يشمل AppBar، Cards، Buttons، Inputs، NavigationBar، Dialogs، BottomSheets، Tabs، وغيرها.

لوحة الألوان

الدور	اللون	الكود
اللون الرئيسي	أزرق بنفسجي عميق	#2633C5
اللون الثانوي	ذهبي/كهرماني	#F1B440
اللون المميز الأزرق	أزرق فاتح	#87A0E5
اللون المميز الوردي	وردي	#F56E98
النجاح	أخضر	#43A047
الخطأ	أحمر	#E53935
خلفية الوضع الداكن	أزرق داكن جداً	#0F0F1A
سطح الوضع الداكن	أزرق رمادي داكن	#1A1A2E

دعم RTL واللغة العربية

يوفر التطبيق دعماً كاملاً للاتجاه من اليمين لليسار عبر Directionality.rtl على مستوى التطبيق بالكامل. يستخدم خط Cairo العربي المتغير (Variable Font) بجميع الأوزان، مع دعم اللغتين العربية والإنجليزية عبر MaterialApp localization delegates. جميع النصوص في الواجهة باللغة العربية مع دعم الاسماء ثنائية اللغة في قاعدة البيانات.

المكونات المخصصة (Widgets)

المكون	الوصف	الاستخدام
CustomBottomBar	شريط تنقل سفلي مخصص مع زر إنشاء مركزي	التنقل الرئيسي
StatCard	بطاقة إحصائية مع أيقونة وعدد وعنوان	لوحة التحكم
BarChartWidget	رسم بياني شريطي مخصص	الإحصائيات
AnimatedEntry	حركة دخول متحركة للعناصر	القوائم والبطاقات
EmptyState	حالة فارغة مع أيقونة ونص	القوائم الفارغة
InvoiceItemCard	بطاقة عنصر فاتورة	إنشاء الفواتير
CartItemTile	عنصر سلة المشتريات	نقطة البيع
TransactionTile	عنصر قيد محاسبي	دفتر الأستاذ
QuickActionButton	زر إجراء سريع	لوحة التحكم

الوضع الداكن (Dark Mode)

يدعم التطبيق الوضع الفاتح والداكن مع تبديل تلقائي حسب إعدادات النظام. يتميز الوضع الداكن بخلفية أزرق داكن (#0F0F1A) وسطح (#1A1A2E) مع تدرجات ألوان متوافقة. جميع المكونات مُصممة بشكل متجاوب مع كلا الوضعين عبر isDark checks.

نقاط القوة والضعف

تقييم شامل لنقاط القوة والضعف والتحديات في المشروع

✓ نقاط القوة

- ✓ تصميم UI/UX حديث ومتجاوب مع Material 3 ونظام ألوان احترافي
- ✓ دعم كامل للغة العربية واتجاه RTL على مستوى التطبيق
- ✓ نظام محاسبي مزدوج (Double-Entry) مع قيود يومية تلقائية
- ✓ دعم تعدد العملات مع حسابات محاسبية مستقلة لكل عملة
- ✓ عمل بدون إنترنت (Offline-First) مع قاعدة بيانات محلية
- ✓ نظام ترحيل قاعدة بيانات متدرج مع معالجة الأخطاء
- ✓ بنية ملفات منظمة ومنطقية مع فصل واضح للطبقات
- ✓ دعم الوضع الداكن والفاتح مع تبديل تلقائي
- ✓ نماذج بيانات (Models) محكمة مع toMap/fromMap/copyWith
- ✓ فهارس قاعدة بيانات محسنة للأداء (19 فهرس)

! نقاط الضعف

- ! عدم وجود طبقة منطق أعمال منفصلة (BLoC/Repository)
- ! التعامل المباشر مع قاعدة البيانات من الشاشات (SQL في UI)
- ! غياب اختبارات وحدة واختبارات تكامل شاملة
- ! عدم استخدام Provider فعلياً لإدارة الحالة رغم إضافته كتبعية
- ! ملفات متعددة بنسخ مكررة (ملفات _current.dart)
- ! عدم وجود نظام مصادقة/تسجيل دخول فعلي رغم جدول users
- ! غياب النسخ الاحتياطي والاستعادة للبيانات
- ! عدم وجود مزامنة سحابية أو مشاركة بيانات بين الأجهزة
- ! README بسيط جداً دون توثيق تقني للمشروع
- ! ملفات مهارة (/skills) غير ضرورية مُضمَّنة في المستودع

ملفات مكررة ومستودع غير نظيف

يحتوي المستودع على عدد كبير من الملفات المكررة بأسماء مثل _current.dart و pubspec_current.yaml، بالإضافة إلى مجلد /skills يحتوي على أكثر من 30 مهارة (ppt, pdf, xlsx, charts, وغيرها) لا علاقة لها بالمشروع الأساسي. كما توجد ملفات APK في مجلد /download وصور في /upload مما يزيد حجم المستودع بشكل غير مبرر. يُنصح بشدة بتنظيف المستودع وإضافة ملف .gitignore مناسب.

مخاوف أمنية

يحتوي جدول users على عمود password_hash لكن لا يوجد نظام مصادقة فعلي في التطبيق. جميع العمليات تتم بدون تسجيل دخول أو صلاحيات، مما يعني أن أي شخص يستخدم التطبيق يمكنه الوصول لجميع البيانات وإجراء أي عملية. كما أن التطبيق لا يشفر قاعدة البيانات المحلية، والبيانات الحساسة مثل أرصدة العملاء والموردين محفوظة بنص عادي في SQLite.

تحليل جودة الكود

المعيار	التقييم	الملاحظات
تنظيم الملفات	جيد	هيكل منطقي بفصل طبقات واضح
قابلية القراءة	جيد	تعليقات عربية واضحة، أسماء معبرة
نماذج البيانات	جيد	نماذج محكمة مع تحويلات كاملة
إدارة الحالة	ضعيف	لا يوجد نمط إدارة حالة فعلي
الاختبارات	ضعيف	اختبار ويدجت واحد فقط
معالجة الأخطاء	متوسط	try-catch في الترحيل فقط
الأمان	ضعيف	لا مصادقة، لا تشفير
التوثيق	ضعيف	README أساسي جداً

التوصيات

توصيات عملية لتحسين المشروع على المستويات المعمارية والأمنية والوظيفية

1 تبني نمط BLoC/Cubit لإدارة الحالة — عالي

استبدال التعامل المباشر مع قاعدة البيانات من الشاشات بنمط BLoC أو Cubit مع طبقة Repository وسيطة. هذا يفصل منطق الأعمال عن العرض، يسهل الاختبار، ويحسن قابلية الصيانة. كل شاشة يجب أن يكون لها BLoC خاص يتعامل مع الحالات (loading, success, error) بطريقة موحدة.

2 تنفيذ نظام مصادقة وتفويض — عالي

تفعيل جدول users الموجود بنظام تسجيل دخول حقيقي مع تشفير كلمات المرور (bcrypt)، وإدارة صلاحيات (admin, manager, cashier) تحدد ما يمكن لكل مستخدم الوصول إليه. إضافة قفل التطبيق (App Lock) بالبصمة أو PIN كما هو مخطط في الملفات المرجعية.

3 إضافة نظام نسخ احتياطي واستعادة — عالي

تطوير آلية تصدير واستيراد قاعدة البيانات كاملة بتنسيق JSON أو ملف مشفر. دعم النسخ الاحتياطي التلقائي المحلي والتخزين السحابي (Google Drive/OneDrive) لمنع فقدان البيانات. هذا أمر حاسم لتطبيق محاسبي حيث البيانات المالية لا يمكن تعويضها.

4 تنظيف المستودع وإدارة الملفات — متوسط

إزالة مجلد skills/ وملفات APK والصور والمستندات البحثية غير المتعلقة بالمشروع. حذف ملفات current.dart_ المكررة. تحديث .gitignore لمنع إدراج الملفات الثنائية والكاش. هذا يقلل حجم المستودع من عشرات الميجابايت إلى أقل من 1 ميجابايت ويحسن تجربة التطوير.

5 كتابة اختبارات شاملة — متوسط

إضافة اختبارات وحدة للنماذج ومنطق الأعمال، واختبارات تكامل لعمليات قاعدة البيانات، واختبارات ويدجت للشاشات الرئيسية. التركيز أولاً على اختبار نظام القيود المحاسبية لضمان صحة الأرصدة، واختبار عمليات الفواتير لضمان اتساق البيانات.

6 تحسين نظام العملات — متوسط

إضافة تحديث تلقائي لأسعار الصرف من مصادر خارجية، ودعم عمليات التحويل بين العملات مع تسجيل فرق الصرف كإيراد/مصروف محاسبي. حالياً أسعار الصرف ثابتة ويجب تحديثها يدوياً، مما يسبب عدم دقة في القيود المحاسبية متعددة العملات.

7 إضافة التقارير والطباعة منخفض

تطوير تقارير مالية متقدمة: الميزانية العمومية، قائمة الدخل، التدفقات النقدية، وتقرير الأرباح والخسائر. إضافة إمكانية تصدير التقارير كملفات PDF وطباعة الفواتير عبر طابعات حرارية (thermal printers) شائعة في المتاجر اليمنية.

خارطة طريق مقترحة

المرحلة الأولى (1-2 شهر): تبني BLoC + نظام مصادقة + نسخ احتياطي. المرحلة الثانية (2-3 أشهر): تنظيف الكود + اختبارات + تحسين العملات. المرحلة الثالثة (3-4 أشهر): تقارير مالية + طباعة + مزامنة سحابية اختيارية. كل مرحلة يجب أن تُختتم بإصدار مستقر مع اختبارات كاملة.