

# # InvoiceFlow - Product Specification Document

## ## 1. Product Overview

**\*\*Product Name:\*\*** InvoiceFlow

**\*\*Version:\*\*** 1.0.0+1

**\*\*Platform:\*\*** Flutter (Cross-platform mobile application)

**\*\*Target Platforms:\*\*** Android, iOS, Web

**\*\*Development Framework:\*\*** Flutter SDK ^3.6.0, Dart

**\*\*Backend:\*\*** Firebase/Firestore (Cloud-first architecture)

**\*\*Authentication:\*\*** Firebase Authentication with Google Sign-In

### ### 1.1 Product Vision

InvoiceFlow is a comprehensive mobile invoice management application designed for small to medium businesses and freelancers. It provides end-to-end invoice lifecycle management, real-time inventory tracking, customer relationship management, and advanced business analytics with cloud synchronization and multi-device support.

### ### 1.2 Key Value Propositions

- **\*\*Complete Business Management\*\*:** Invoice, inventory, customer, and analytics in one unified platform
- **\*\*Real-time Cloud Sync\*\*:** Firebase/Firestore backend with instant multi-device synchronization
- **\*\*Advanced Authentication\*\*:** Firebase Auth with email/password and Google Sign-In
- **\*\*Intelligent Analytics\*\*:** Real-time business insights with interactive charts and KPIs
- **\*\*Professional Design\*\*:** Sophisticated Material 3 theme with navy blue and sage green palette
- **\*\*Cross-Platform Excellence\*\*:** Single Flutter codebase for Android, iOS, and Web

## **## 2. Technical Architecture**

### **### 2.1 Technology Stack**

- **Frontend Framework**: Flutter 3.6.0+ with Material Design 3
- **Programming Language**: Dart
- **Backend**: Firebase/Firestore (NoSQL cloud database)
- **Authentication**: Firebase Authentication
- **Cloud Storage**: Firebase Storage
- **Analytics**: Firebase Analytics
- **Messaging**: Firebase Cloud Messaging
- **State Management**: Provider pattern with ChangeNotifier
- **UI Framework**: Material Design 3 with custom theming
- **Typography**: Google Fonts (Inter, JetBrains Mono, Poppins)
- **Charts**: FL Chart for data visualization
- **Responsive Design**: Sizer package for adaptive layouts

### **### 2.2 Architecture Patterns**

- **Clean Architecture**: Layered separation of concerns
- **Repository Pattern**: Data access abstraction via Firebase services
- **Service Layer**: Business logic encapsulation with dedicated services
- **Provider Pattern**: State management and dependency injection
- **Multi-tenant Architecture**: User-scoped Firestore collections
- **Event-Driven Architecture**: Real-time updates via streams and notifications

### **### 2.3 Firebase/Firestore Data Structure**

...

```
users/{uid}/  
  └─ customers/{customerId}  
    |  └─ name: String  
    |  └─ phoneNumber: String  
    |  └─ createdAt: Timestamp  
    |  └─ updatedAt: Timestamp  
    └─ invoices/{invoiceId}  
      |  └─ invoiceNumber: String  
      |  └─ clientName: String  
      |  └─ customerPhone: String?  
      |  └─ customerId: String?  
      |  └─ date: Timestamp  
      |  └─ revenue: double  
      |  └─ status: String  
      |  └─ invoiceType: String  
      |  └─ amountPaid: double  
      |  └─ items: Array<InvoiceItem>  
      |  └─ createdAt: Timestamp  
      |  └─ updatedAt: Timestamp  
      |  └─ [audit fields]  
    └─ inventory_items/{itemId}  
      |  └─ sku: String  
      |  └─ name: String  
      |  └─ unit: String
```

```
|   |--- current_stock: double  
|   |--- reorder_point: double  
|   |--- avg_cost: double  
|   |--- category: String  
|   └── last_updated: Timestamp  
└── stock_movements/{movementId}  
    |--- itemId: String  
    |--- type: String  
    |--- quantity: double  
    |--- unitCost: double  
    |--- sourceRefType: String  
    |--- sourceRefId: String  
    |--- createdAt: Timestamp  
    └── [reversal fields]  
...  
...
```

#### ### 2.4 Security & Privacy

- **Firebase Authentication**: Secure user authentication with email/password and Google OAuth
- **Multi-tenant Data Isolation**: User-scoped collections prevent cross-user data access
- **Firestore Security Rules**: Server-side security enforcement
- **Data Encryption**: Automatic encryption in transit and at rest
- **Input Validation**: Comprehensive form validation and sanitization
- **Error Handling**: Graceful error management with user-friendly messages

#### ## 3. Core Features & Functionality

### **### 3.1 Authentication System**

#### **#### 3.1.1 Authentication Methods**

- **Email/Password Authentication**: Traditional sign-up and sign-in
- **Google Sign-In**: OAuth integration with Google accounts
- **Password Recovery**: Email-based password reset functionality
- **Session Management**: Automatic session handling with Firebase Auth

#### **#### 3.1.2 User Experience**

- **AuthWrapper**: Automatic routing based on authentication state
- **Loading States**: Smooth loading indicators during auth operations
- **Error Handling**: User-friendly error messages for auth failures
- **Auto-Navigation**: Seamless transition between auth and app screens

### **### 3.2 Invoice Management**

#### **#### 3.2.1 Invoice Creation & Types**

- **Sales Invoices**: Customer billing with line items and pricing
- **Purchase Invoices**: Vendor billing and expense tracking
- **Draft System**: Save invoices as drafts before posting
- **Line Item Management**: Multiple products per invoice with individual pricing
- **Automatic Numbering**: Sequential invoice number generation

#### **#### 3.2.2 Invoice Lifecycle Management**

- **Status Workflow**: Draft → Posted → Paid → Cancelled
- **Payment Tracking**: Partial and full payment recording
- **Payment Methods**: Multiple payment method support

- **Follow-up System**: Automated and manual reminder scheduling
- **Modification Tracking**: Complete audit trail for invoice changes

#### **#### 3.2.3 Advanced Invoice Operations**

- **Invoice Cancellation**: Validated cancellation with inventory reversal
- **Admin Override**: Force operations with audit trail and reason tracking
- **Bulk Operations**: Mass invoice processing capabilities
- **Export Functions**: PDF generation and sharing
- **WhatsApp Integration**: Direct payment reminders via WhatsApp

### **## 3.3 Customer Management**

#### **#### 3.3.1 Customer Profiles (Firestore-based)**

- **Contact Information**: Name, phone, email, address
- **Purchase History**: Complete transaction timeline from Firestore
- **Outstanding Balances**: Real-time payment tracking and aging
- **Customer Analytics**: Spending patterns and behavior analysis

#### **#### 3.3.2 Customer Operations**

- **User-scoped Data**: Each user's customers isolated in Firestore
- **Real-time Updates**: Instant synchronization across devices
- **Phone-based Lookup**: Efficient customer search and duplicate prevention
- **Customer Insights**: Revenue contribution and purchase frequency analysis

### **## 3.4 Inventory Management System**

#### **#### 3.4.1 Real-time Inventory Tracking**

- **Automatic Stock Updates**: Invoice posting triggers inventory changes

- **Stock Movement Audit**: Complete traceability of all stock changes
- **Movement Types**: IN, OUT, ADJUSTMENT, RETURN\_IN, RETURN\_OUT, REVERSAL\_OUT
- **Real-time Calculations**: Live stock computation from movement history

#### #### 3.4.2 Advanced Inventory Features

- **Cancellation Validation**: Prevent negative stock scenarios with dependency checking
- **Admin Override**: Force operations with comprehensive audit trail
- **Low Stock Alerts**: Automatic notifications when items reach reorder points
- **Reorder Management**: Intelligent reorder quantity suggestions
- **Barcode Support**: Product identification and scanning capabilities

#### #### 3.4.3 Inventory Analytics

- **Stock Valuation**: Real-time inventory value calculation
- **Movement Analysis**: Fast/slow moving item identification
- **Age Analysis**: Inventory holding period tracking
- **Performance Metrics**: Turnover rates and efficiency indicators

### ## 3.5 Analytics & Reporting (Real-time)

#### #### 3.5.1 Financial Analytics

- **Revenue Analysis**: Real-time sales performance with correct line-item calculations
- **Revenue Formula**: Revenue = SUM(quantity × selling price) for sales invoice line items
- **Payment Analytics**: Collection efficiency and outstanding receivables
- **Profit Analysis**: Gross profit calculation and trend analysis

#### #### 3.5.2 Interactive Dashboard

- **7-Card Main Screen**: Overview KPIs, Revenue, Items & Sales, Inventory, Due Reminders, Analytics Table, Charts
- **Date Range Filtering**: Last 7/30/90 days, custom date ranges
- **Real-time KPIs**: Live business metrics with trend indicators
- **Visual Charts**: Bar charts, pie charts, line graphs with real data

#### #### 3.5.3 Advanced Analytics Features

- **Customer Analytics**: Customer lifetime value and segmentation
- **Item Performance**: Top-selling items and category analysis
- **Due Reminders**: Customer-wise and item-wise outstanding tracking
- **Inventory Health**: Stock distribution and movement analysis

### ## 4. User Interface Design

#### ### 4.1 Design System

##### #### 4.1.1 Material 3 Theme

- **Primary Colors**: Deep Navy Blue (#1B3A57), Serene Blue (#4A90B8)
- **Secondary Colors**: Sage Green (#5D8A72), Sophisticated Gold (#B8860B)
- **Background Colors**: Soft Off-white (#FAFAFC), Pure White (#FFFFFF)
- **Status Colors**: Muted Red (#B85450), Success Green, Warning Gold

##### #### 4.1.2 Typography System

- **Primary Font**: Inter (UI elements, headings)
- **Monospace Font**: JetBrains Mono (financial data, invoice numbers)
- **Display Font**: Poppins (welcome screens, branding)
- **Font Weights**: Light (300) to Bold (700) with proper hierarchy

#### **#### 4.1.3 Component Design**

- **Rounded Aesthetics**: 24px border radius for modern look
- **Sophisticated Shadows**: Color-tinted shadows with proper elevation
- **Button Variants**: Elevated, outlined, and text buttons with consistent styling
- **Input Fields**: Filled style with rounded borders and proper validation

### **### 4.2 Navigation Architecture**

#### **#### 4.2.1 Bottom Navigation**

- **Home Dashboard**: Business overview and quick actions
- **Invoices**: Invoice list and management
- **Analytics**: Business insights and reporting (7-card layout)
- **Customers**: Customer management and profiles
- **Profile**: User settings and app configuration

#### **#### 4.2.2 Screen Hierarchy**

- **Authentication Flow**: Login → Register → Forgot Password
- **Main Screens**: Primary navigation destinations with bottom nav
- **Detail Screens**: Entity-specific information and actions
- **Modal Screens**: Contextual actions and forms
- **Full-Screen Analytics**: Comprehensive analytics with date filtering

### **### 4.3 Responsive Design**

- **Adaptive Layouts**: Screen size optimization using Sizer package
- **Flexible Components**: Responsive sizing with percentage-based dimensions
- **Overflow Handling**: Proper text truncation and scrollable containers

- **\*\*Touch Targets\*\***: Minimum 44px touch areas for accessibility

## ## 5. Business Logic & Rules

### ### 5.1 Revenue Calculation (Corrected Implementation)

- **\*\*Accurate Formula\*\***: Revenue = SUM(quantity × selling price) for sales invoice line items
- **\*\*Analytics Consistency\*\***: All revenue metrics use line item calculations, not invoice totals
- **\*\*Multi-currency Ready\*\***: Prepared for future currency handling
- **\*\*Real-time Updates\*\***: Revenue calculations update instantly with data changes

### ### 5.2 Inventory Business Rules

- **\*\*Stock Validation\*\***: Comprehensive validation to prevent negative stock
- **\*\*Movement Audit\*\***: Complete traceability with reversal capabilities
- **\*\*Cancellation Safety\*\***: Validate dependent documents before allowing cancellations
- **\*\*Admin Override\*\***: Allow force operations with mandatory audit trail and reasoning

### ### 5.3 Multi-tenant Security

- **\*\*User Isolation\*\***: All data strictly scoped to authenticated user UID
- **\*\*Firestore Rules\*\***: Server-side security rules enforce user-based access
- **\*\*Authentication Required\*\***: All operations require valid Firebase authentication
- **\*\*Data Integrity\*\***: Atomic operations and proper error handling

## ## 6. Services Architecture

### ### 6.1 Core Services

- **\*\*AuthService\*\***: Firebase Authentication management

- **FirestoreService**: Core invoice and customer CRUD operations
- **InventoryFirestoreService**: Inventory and stock movement operations
- **InvoiceService**: Business logic layer with CSV migration support
- **CustomerService**: Customer management with WhatsApp integration
- **AnalyticsService**: Real-time analytics with correct revenue calculations

### ### 6.2 Notification Services

- **NotificationService**: Local push notifications
- **InventoryNotificationService**: Real-time inventory update streams
- **BackgroundService**: Daily reminder scheduling
- **EventService**: App-wide event broadcasting

### ### 6.3 Utility Services

- **StockMapService**: Inventory update coordination
- **BulkInvoiceImport**: CSV data migration utilities
- **HapticFeedbackUtil**: User interaction feedback

## ## 7. Performance Requirements

### ### 7.1 Response Time Requirements

- **App Launch**: < 3 seconds with Firebase initialization
- **Authentication**: < 2 seconds for sign-in operations
- **Data Loading**: < 2 seconds for Firestore queries
- **Real-time Updates**: Instant synchronization via Firestore listeners
- **Analytics Generation**: < 3 seconds for complex calculations

### **### 7.2 Scalability Requirements**

- **Data Volume**: Unlimited with Firestore's horizontal scaling
- **Concurrent Users**: Supported by Firebase infrastructure
- **Real-time Sync**: Automatic across multiple devices and platforms
- **Global Distribution**: Firebase's global CDN and edge locations

## **## 8. Testing & Quality Assurance**

### **### 8.1 Testing Strategy**

- **Unit Testing**: Core business logic validation
- **Widget Testing**: UI component testing with flutter\_test
- **Integration Testing**: End-to-end workflow validation
- **Performance Testing**: Load testing with Firebase
- **Authentication Testing**: Comprehensive auth flow testing

### **### 8.2 Quality Metrics**

- **Code Coverage**: Comprehensive test coverage for critical paths
- **Performance Benchmarks**: Defined response time targets
- **User Experience**: Usability testing and feedback integration
- **Error Tracking**: Firebase Crashlytics integration
- **Analytics**: Firebase Analytics for user behavior insights

## **## 10. Data Protection**

- **GDPR Compliance**: User data portability and deletion capabilities
- **Privacy Policy**: Comprehensive privacy policy implementation

- **Data Minimization**: Collect only necessary business data
- **Audit Trail**: Complete operation logging for compliance requirements

### **### 10.2 Security Measures**

- **Firebase Security Rules**: Server-side access control
- **Input Sanitization**: Comprehensive validation and sanitization
- **Error Handling**: Secure error messages without data leakage
- **Session Management**: Automatic session timeout and renewal