**Expense Tracker Mobile App - Detailed Implementation Plan**

**Phase 1: Project Setup & Foundation (Weeks 1-3)**

**Week 1: Environment Setup & Project Initialization ✅ COMPLETED**

* ✅ Set up development environment (VSCode, dependencies, extensions)
* ✅ Initialize React project with TypeScript
* ✅ Configure build system (webpack/vite, babel)
* ✅ Set up linting and formatting (ESLint, Prettier)
* ✅ Configure Firebase project
  + ✅ Create Firebase project
  + ✅ Set up Firebase Authentication
  + ✅ Configure Firebase storage rules

**Week 2: UI Foundation & Authentication ✅ COMPLETED**

* ✅ Design system implementation
  + ✅ Create color palette constants
  + ✅ Implement typography components
  + ✅ Create button component variants
  + ✅ Implement form input components
  + ✅ Create card and container components
* ✅ Authentication screens
  + ✅ Create login screen
  + ✅ Create registration screen
  + ✅ Create password reset screen
  + ✅ Build phone verification components
* ✅ Authentication logic
  + ✅ Implement Firebase auth hooks
  + ✅ Set up protected routes
  + ✅ Create authentication context
  + ✅ Implement session persistence

**Week 3: Core Navigation & Data Structures ⚠️ IN PROGRESS**

* ✅ Main navigation structure
  + ✅ Create app shell with bottom navigation
  + ✅ Implement screen transitions
  + ✅ Build header components
  + ✅ Implement navigation context/state
* ⚠️ Data models & storage
  + ✅ Define Transaction interface
  + ✅ Create Category data structure
  + ⚠️ Implement storage utility functions (50% complete)
  + ❌ Create data validation schemas
* ⚠️ Basic state management
  + ✅ Create UserContext for profile data
  + ⚠️ Implement TransactionContext (70% complete)
  + ❌ Build settings state management

**Phase 2: Core Features Development (Weeks 4-8)**

**Week 4-5: Profile & SMS Integration ⚠️ IN PROGRESS**

* ✅ Complete user profile
  + ✅ Create profile editing screen
  + ✅ Implement image upload for profile pictures
  + ✅ Build preference settings
  + ✅ Create currency selection component
* ⚠️ SMS provider selection
  + ✅ Create SMS provider selection UI
  + ⚠️ Implement provider connection logic (40% complete)
  + ❌ Build provider management screen
* ❌ SMS reading capability
  + ❌ Research platform-specific SMS access APIs
  + ❌ Implement SMS permission requests
  + ❌ Create SMS reading service
  + ❌ Build SMS history viewer

**Week 6-7: Transaction Processing ⚠️ IN PROGRESS**

* ✅ SMS parsing engine
  + ✅ Create SMS message schema
  + ✅ Implement regex patterns for common banks
  + ✅ Build transaction extraction logic
  + ✅ Create validation for parsed data
* ⚠️ Transaction management
  + ✅ Implement transaction creation
  + ⚠️ Create transaction listing component (80% complete)
  + ⚠️ Build transaction detail view (60% complete)
  + ❌ Implement transaction editing
  + ❌ Build transaction deletion with confirmation
* ⚠️ Manual entry interface
  + ✅ Create form for manual transaction entry
  + ⚠️ Implement category selection (60% complete)
  + ❌ Build receipt photo capture
  + ❌ Implement recurring transaction logic

**Week 8: Basic Reporting ⚠️ MINIMAL PROGRESS**

* ⚠️ Dashboard development
  + ✅ Create dashboard layout
  + ⚠️ Implement balance summary card (70% complete)
  + ⚠️ Build recent transactions list (50% complete)
  + ❌ Create quick actions menu
* ⚠️ Basic charts
  + ✅ Set up charts library integration
  + ⚠️ Create expense pie chart (30% complete)
  + ❌ Implement income vs expense bar chart
  + ❌ Build spending trend line chart
* ❌ Data aggregation
  + ❌ Create data aggregation utilities
  + ❌ Implement category-based spending summaries
  + ❌ Build monthly comparison logic
  + ❌ Create budget vs actual calculations

**Phase 3: Advanced Features (Weeks 9-12)**

**Week 9-10: Enhanced Reporting & Multi-currency ❌ NOT STARTED**

* ❌ Custom dashboard
  + ❌ Create dashboard widget components
  + ❌ Implement widget configuration
  + ❌ Build dashboard layout customization
  + ❌ Create widget persistence logic
* ❌ Advanced reports
  + ❌ Build custom date range selector
  + ❌ Implement detailed category breakdown
  + ❌ Create exportable reports
  + ❌ Build comparison reports (month-to-month, year-to-year)
* ❌ Multi-currency support
  + ❌ Implement currency conversion service
  + ❌ Create multi-currency transaction handling
  + ❌ Build currency preference settings
  + ❌ Implement base currency calculations

**Week 11: Performance Optimization ❌ NOT STARTED**

* ❌ Code optimization
  + ❌ Implement code splitting
  + ❌ Optimize re-renders with memoization
  + ❌ Refactor heavy computations
  + ❌ Add virtualization for long lists
* ❌ Data management
  + ❌ Implement pagination for large data sets
  + ❌ Create efficient filtering mechanisms
  + ❌ Build data caching strategy
  + ❌ Optimize state updates
* ✅ Initial security review
  + ✅ Implement basic error handling
  + ✅ Create input validation
  + ❌ Review data storage security
  + ❌ Implement sensitive data protection

**Week 12: Advanced Integrations ❌ NOT STARTED**

* ❌ Currency conversion service
  + ❌ Research and select API provider
  + ❌ Implement API client
  + ❌ Create rate caching mechanism
  + ❌ Build conversion UI
* ❌ Enhanced SMS parsing
  + ❌ Extend support for additional banks
  + ❌ Implement learning from user corrections
  + ❌ Create confidence scoring for parsed transactions
  + ❌ Build parsing rule editor
* ❌ Notification system
  + ❌ Create transaction alert system
  + ❌ Implement budget threshold notifications
  + ❌ Build scheduled report reminders
  + ❌ Create custom notification preferences

**Phase 4: Testing & Refinement (Weeks 13-16)**

**Week 13: Comprehensive Testing ❌ NOT STARTED**

* ❌ Unit testing
  + ❌ Set up testing framework
  + ❌ Write tests for core utilities
  + ❌ Test SMS parsing logic
  + ❌ Validate calculation functions
* ❌ Integration testing
  + ❌ Test user flows
  + ❌ Validate form submissions
  + ❌ Test data persistence
  + ❌ Verify navigation patterns
* ❌ Performance testing
  + ❌ Test with large datasets
  + ❌ Measure and optimize render times
  + ❌ Test on low-end devices
  + ❌ Optimize battery usage

**Week 14: User Acceptance Testing ❌ NOT STARTED**

* ❌ Beta testing setup
  + ❌ Create beta distribution pipeline
  + ❌ Set up feedback collection mechanism
  + ❌ Write beta tester guidelines
  + ❌ Prepare test scenarios
* ❌ User testing
  + ❌ Recruit beta testers
  + ❌ Collect and categorize feedback
  + ❌ Prioritize issues and enhancements
  + ❌ Conduct usability interviews
* ❌ Initial bug fixes
  + ❌ Address critical bugs
  + ❌ Fix UI/UX issues
  + ❌ Resolve performance bottlenecks
  + ❌ Patch security vulnerabilities

**Week 15: Platform Optimization ❌ NOT STARTED**

* ❌ iOS optimization
  + ❌ Implement iOS-specific UI adjustments
  + ❌ Optimize for different iPhone models
  + ❌ Address iOS-specific permissions
  + ❌ Test and fix iOS-specific issues
* ❌ Android optimization
  + ❌ Implement Android-specific UI adjustments
  + ❌ Optimize for diverse Android devices
  + ❌ Address Android-specific permissions
  + ❌ Test and fix Android fragmentation issues
* ❌ Cross-platform consistency
  + ❌ Ensure feature parity across platforms
  + ❌ Normalize UI behavior
  + ❌ Create platform detection utilities
  + ❌ Implement graceful fallbacks

**Week 16: Deployment Preparation ❌ NOT STARTED**

* ❌ Final polishing
  + ❌ Create app splash screen and icons
  + ❌ Polish animations and transitions
  + ❌ Finalize copy and text
  + ❌ Conduct final UX review
* ❌ App store preparation
  + ❌ Create app store screenshots
  + ❌ Write compelling app descriptions
  + ❌ Prepare privacy policy
  + ❌ Create promotional materials
* ❌ Launch infrastructure
  + ❌ Scale backend services
  + ❌ Set up monitoring and alerts
  + ❌ Prepare for user onboarding
  + ❌ Create support documentation

**Current Status Summary:**

* **Phase 1**: ~85% complete (Foundation largely in place, finalizing data structures)
* **Phase 2**: ~35% complete (Core features partially implemented, focusing on transaction processing)
* **Phase 3**: ~5% complete (Only basic error handling implemented)
* **Phase 4**: 0% complete (Not started)