

Phone Verification Authentication - Implementation Summary

Implementation Complete

This document summarizes the phone verification authentication system implemented for the Yardım Yolda Flutter app.

Files Created

Domain Layer (Models)

1. **lib/features/auth/domain/models/user_profile.dart**
 - UserProfile model matching the database schema
 - JSON serialization/deserialization
 - copyWith method for immutability
2. **lib/features/auth/domain/models/auth_state.dart**
 - AuthState model for managing authentication status
 - Helper methods: isAuthenticated, isNewUser
 - Factory constructors for different states

Data Layer (Repository)

1. **lib/features/auth/data/repositories/auth_repository.dart**
 - sendOTP() - Send verification code via SMS
 - verifyOTP() - Verify the OTP code
 - getUserProfile() - Fetch user profile from database
 - createUserProfile() - Create new user profile
 - updateUserProfile() - Update existing profile
 - signOut() - Sign out user
 - Turkish error messages

State Management (Riverpod)

1. **lib/features/auth/providers/auth_state_provider.dart**
 - AuthStateNotifier for managing auth state
 - Listens to Supabase auth changes
 - Providers:
 - authStateProvider - Main auth state
 - currentUserProvider - Current user
 - currentUserProfileProvider - User profile
 - isAuthenticatedProvider - Auth status check

Presentation Layer (Screens)

1. **lib/features/auth/presentation/screens/auth_phone_screen.dart**
 - Phone number entry with Turkish formatting
 - Real-time validation
 - Auto-formatting (5XX XXX XX XX)

- Converts to international format (+90...)
- Loading states and error handling

2. **lib/features/auth/presentation/screens/otp_verify_screen.dart**

- 6-digit OTP input with auto-focus
- 60-second resend timer
- Auto-verify when complete
- Error handling with retry
- Masked phone number display

3. **lib/features/auth/presentation/screens/onboarding_screen.dart**

- 3-page onboarding flow
- Welcome page
- Features showcase
- Name entry and profile creation
- Skip functionality

Configuration & Routing

1. **lib/routes/app_router.dart** (Updated)

- New routes: /auth/phone, /auth/verify, /onboarding
- Auth-aware navigation
- Automatic redirects based on auth state
- Protection for authenticated routes

2. **lib/features/auth/presentation/pages/splash_page.dart** (Updated)

- Integrated with auth state provider
- Smart navigation based on user status

3. **lib/features/dashboard/presentation/pages/dashboard_page.dart** (Updated)

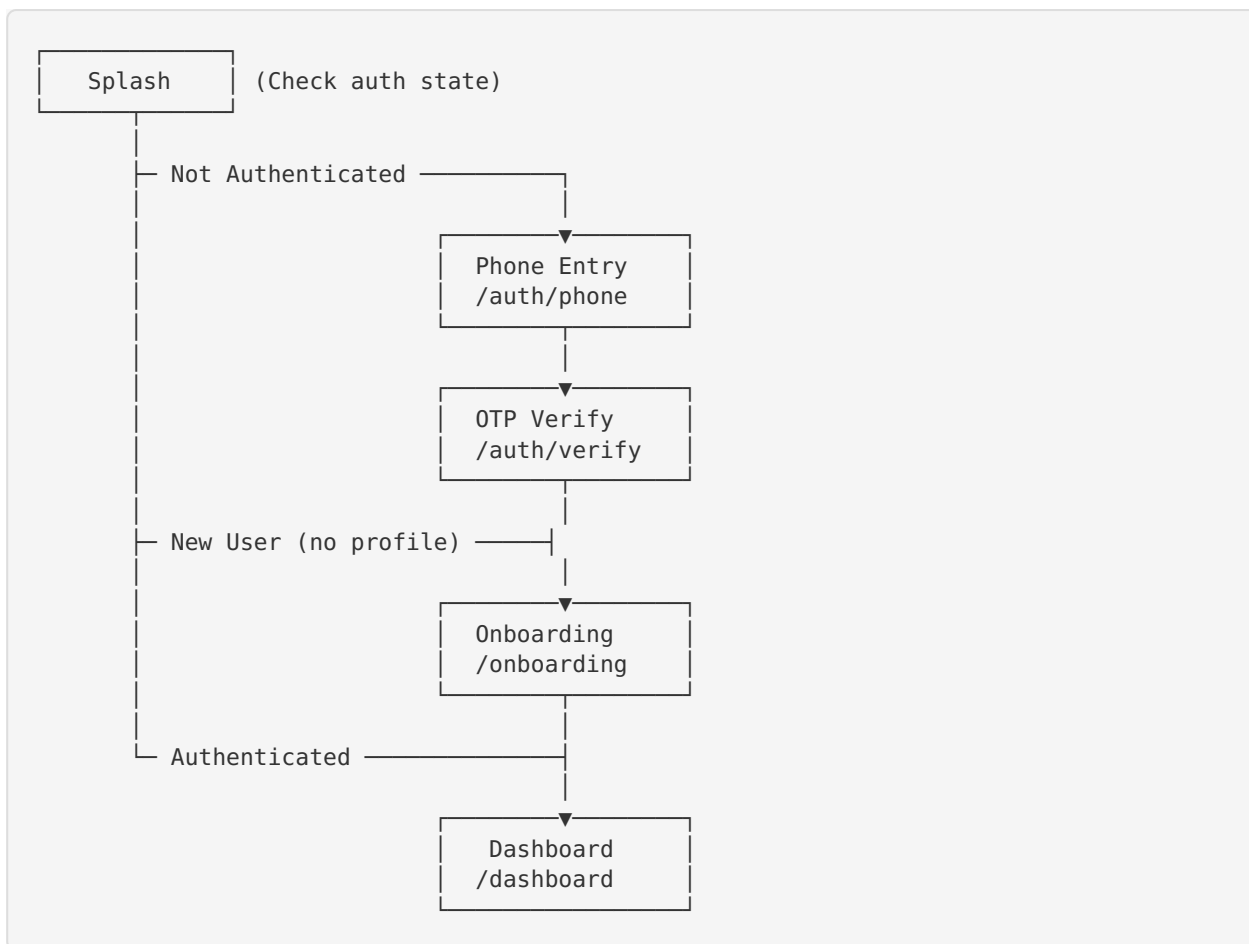
- Display user profile information
- Sign out functionality using auth provider

Documentation

1. **lib/features/auth/README.md**

- Comprehensive authentication flow documentation
- Architecture overview
- Usage examples
- Database schema

Authentication Flow



UI Features

Phone Entry Screen





- ☒ Turkish phone number formatting
- ☒ Real-time validation
- ☒ Auto-formatting as user types
- ☒ Clear instructions in Turkish
- ☒ Loading indicator during API call
- ☒ Error messages via Snackbar

OTP Verification Screen

- ☒ 6 separate input boxes
- ☒ Auto-focus on next field
- ☒ Auto-verify when complete
- ☒ 60-second resend countdown
- ☒ Masked phone number display
- ☒ Loading state with Turkish text





Onboarding Screen

- ☒ 3-page introduction
- ☒ Feature highlights







-  Name collection
-  Skip button
-  Page indicators
-  Profile creation

Security & Error Handling

Security Features

-  Supabase Auth with PKCE flow
-  Secure OTP verification
-  Session management
-  Protected routes

Error Handling

-  Network errors
-  Invalid OTP codes
-  Expired codes
-  Rate limiting
-  Profile creation failures
-  All errors in Turkish

Database Integration

User Profile Creation

When a user verifies their phone number:

1. System checks if profile exists in `user_profiles` table
2. If not exists, creates new record with:
 - id: `auth.user().id`
 - phone: verified phone number
 - role: 'customer' (default)
 - created_at, updated_at: timestamps
3. If exists, just redirects to dashboard

Profile Updates

After onboarding, the profile is updated with:

- full_name: from user input
- Other fields can be added later

Localization

All text is in Turkish:

- Screen titles and labels
- Button text
- Error messages
- Validation messages
- Success notifications
- Loading indicators

Dependencies Used

- flutter_riverpod: State management
- supabase_flutter: Authentication and database
- go_router: Navigation and routing
- flutter (built-in): UI components and formatters

Testing Checklist

Before deploying, test:

- [] Phone number formatting
- [] OTP sending
- [] OTP verification
- [] Profile creation for new users
- [] Profile retrieval for existing users
- [] Onboarding flow
- [] Navigation between screens
- [] Sign out functionality
- [] Error scenarios (invalid OTP, expired code, etc.)
- [] Resend OTP functionality

Next Steps

To use this implementation:

1. Setup Supabase

- Enable Phone Auth in Supabase Dashboard
- Configure phone provider (Twilio, MessageBird, etc.)
- Set up rate limiting

2. Create Database Table

```
sql
CREATE TABLE user_profiles (
  id UUID PRIMARY KEY REFERENCES auth.users(id),
  phone TEXT NOT NULL,
  full_name TEXT,
  role TEXT NOT NULL DEFAULT 'customer',
  avatar_url TEXT,
  created_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
  updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW()
);
```

3. Configure Environment

- Add SUPABASE_URL to .env
- Add SUPABASE_ANON_KEY to .env

4. Run the App

```
bash
flutter pub get
flutter run
```



Code Quality

- ☒ Clean architecture (domain/data/presentation)
- ☒ Separation of concerns
- ☒ Proper error handling
- ☒ Loading states
- ☒ Type safety
- ☒ Code comments
- ☒ Null safety
- ☒ Immutable state
- ☒ Reactive programming



Key Achievements

1. **Complete Phone OTP Flow:** From phone entry to dashboard
2. **Automatic Profile Management:** Creates profiles for new users
3. **Turkish Language Support:** All UI in Turkish
4. **Smart Navigation:** Auth-aware routing with automatic redirects
5. **User Experience:** Loading states, error handling, validation
6. **State Management:** Clean Riverpod implementation
7. **Security:** Supabase Auth with proper session management
8. **Documentation:** Comprehensive README and comments



Support

For issues or questions about this implementation:

1. Check the README in lib/features/auth/
2. Review Supabase documentation
3. Check Riverpod documentation for state management

Implementation Date: October 22, 2025

Status: ☒ Complete and Ready for Testing