

# Nevus App - Technical Documentation

---

Margherita Cignoni

---

## Introduction

The role of self-skin examination is considered very important for melanoma prevention, and many works recommend to keep track in time of existing moles in time using photos [^1][^2][^3] [^1]: Goodson, Agnessa Gadeliya, and Douglas Grossman. "Strategies for early melanoma detection: Approaches to the patient with nevi." Journal of the American Academy of Dermatology 60.5 (2009): 719-735. [^2] Weinstock, Martin A., et al. "Melanoma early detection with thorough skin self-examination: the "Check It Out" randomized trial." American Journal of Preventive Medicine 32.6 (2007): 517-524.

A comprehensive Flutter application designed to help people track and monitor their moles over time. The app allows users to take photos, identify individual moles, create a history of changes for each mole, and locate mole positions on a human figure for comprehensive tracking.

## Features

- **Photo Management:** Take and store photos in a private gallery accessible only through the app
- **Mole Tracking:** Identify and track individual moles with detailed annotations
- **Campaign System:** Organize photos by date into campaigns for systematic monitoring
- **Spot Annotations:** Mark and annotate specific moles on photos with size and position data
- **User Management:** Support for multiple user accounts with secure local storage
- **Fitzpatrick Scale Integration:** Built-in skin type assessment for personalized risk evaluation
- **Export Functionality:** Export mole data and photos for medical consultations
- **Metadata Extraction:** Automatic extraction of photo metadata including capture date

## Architecture

### Data Models

- **Photo:** Core data structure containing image path, capture date, body region, and spot annotations
- **Mole:** Individual mole entities with unique IDs, descriptions, and tracking history
- **Campaign:** Time-based groupings of photos for systematic monitoring
- **Spot:** Annotations on photos that mark mole locations with size and position data
- **User:** User account information with secure local storage

### Directory Structure

```
lib/
├── main.dart           # Application entry point
├── models/             # Data models
│   ├── campaign.dart
│   ├── mole.dart
│   └── photo.dart
```

```
|   ├── spot.dart
|   └── user.dart
|   └── screens/                                # UI screens
|       ├── auth_screen.dart
|       ├── camera_screen.dart
|       ├── campaigns_screen.dart
|       ├── campaign_detail_screen.dart
|       ├── edit_account_screen.dart
|       ├── fitzpatrick_info_screen.dart
|       ├── home_screen.dart
|       ├── mole_detail_screen.dart
|       ├── mole_list_screen.dart
|       ├── photo_gallery_screen.dart
|       └── single_photo_screen.dart
|   └── services/                              # Business logic services
|       ├── campaign_service.dart
|       ├── export_service.dart
|       └── photo_metadata_service.dart
|   └── storage/                               # Data persistence
|       ├── campaign_storage.dart
|       ├── photo_storage.dart
|       └── user_storage.dart
|   └── utils/                                 # Utility functions
|       └── dialog_utils.dart
|   └── widgets/                              # Reusable UI components
|       ├── app_bar_title.dart
|       ├── interactive_photo_viewer.dart
|       ├── mark_mode_controls.dart
|       ├── photo_grid_item.dart
|       ├── photo_info_panel.dart
|       └── spot_widget.dart
```

## Technical Stack

- **Framework:** Flutter 3.8.1+
- **Language:** Dart
- **Storage:** Local JSON files with structured data organization
- **Camera:** Native camera integration with metadata extraction
- **Image Processing:** Support for EXIF data extraction and photo manipulation
- **File Management:** Secure local file system storage

## Key Dependencies

- **camera:** Device camera access and photo capture
- **image\_picker:** Alternative photo selection methods
- **path\_provider:** File system path management
- **file\_picker:** File selection capabilities
- **share\_plus:** Data export and sharing functionality
- **exif:** Photo metadata extraction
- **intl:** Date formatting and internationalization

- **archive**: Data compression for exports

## Data Model

Each user has a personal folder named after the user where all data is stored. When a user doesn't log in, the account will be managed as a guest user and the data will be stored in a folder named "guest". The image data inside each user folder is organized in campaign folders (all photos relative to one campaign are stored in the same campaign folder). All data is stored in JSON files that are loaded and saved when entering or exiting a screen. The JSON files are stored directly inside the user folder.

```
/app_documents/  
└─ users/  
    └─ guest/  
        └─ photos.json  
        └─ campaigns.json  
        └─ moles.json  
        └─ campaigns/  
            └─ campaign_001/  
                └─ photo1.jpg  
                └─ photo2.jpg  
            └─ campaign_002/  
└─ john_doe/  
    └─ photos.json  
    └─ campaigns.json  
    └─ moles.json  
    └─ campaigns/  
        └─ campaign_001/
```

### Photo

Photos are linked to only one campaign. Each image can contain more than one mole and has a list of spots that identify each mole and its position. Each photo contains a description that identifies which region of the body the picture represents.

### Spot

Spots are annotations on a specific photo that highlight moles. Each spot contains information about the mole it identifies, such as an ID, its position, and its size.

### Campaign

Campaigns are linked to a date and represent the group of photos taken at a specific moment (day, week) to track your moles.

### Mole

Every mole has a unique ID, a short string that describes it, a long description, and an automatic method that retrieves all the photos where it appears.

## User

User accounts store personal information and preferences, with support for multiple accounts on the same device.

## User Interface Screens

### Authentication Screen (auth\_screen)

The authentication screen allows users to access the app by entering their username and password in the login form. Users without an account can tap the "plus" button in the top right corner to navigate to the account creation page where they can set up their first account or add additional ones.

### **Edit an Account (edit\_account\_screen)**

This screen enables users to create their first account, add new accounts, or modify existing accounts. The app supports multiple user accounts on the same device, ensuring that all personal data remains strictly within the app without being shared with external platforms. Users can access this screen through the "plus" button on the login screen or by selecting "edit account" from the menu. The form requests personal information such as gender and age, along with username and password credentials. Users can save changes or create new accounts using the button at the bottom of the page.

### Home Screen (home\_screen)

The home screen displays a human figure where users will be able to position their mole photos in future updates. This feature will allow users to create a visual map of their moles, making tracking more intuitive and comprehensive. The screen also provides quick access to key features through two main buttons: one for the camera to take new photos and another for the gallery where all photos are stored. Users can view their most recent campaigns directly on this screen, with dedicated buttons to access the complete campaign list and mole list.

### Campaign List (campaigns\_screen)

This screen presents users with a comprehensive list of all campaigns they have created throughout their use of the app.

### Camera Screen (camera\_screen)

The camera screen allows users to capture photos that will automatically appear in their gallery. After taking a photo, users receive a "Picture saved!" confirmation message along with a button to navigate directly to the gallery screen. If the app encounters an issue while capturing an image, users will see an "Error taking picture" message.

### Campaign Detail Screen (campaign\_detail\_screen)

This screen displays all images from a specific campaign, excluding any that users have deleted. Users can tap on individual photos to view detailed information such as the capture date and photo name. While photo names are initially blank by default, users can customize them using the "Edit name" button.

### **Single Photo Screen (single\_photo\_screen)**

The single photo screen provides users with detailed information about individual photos, including the capture date and all existing annotations. Users can add new spots or edit existing ones to highlight and track specific moles on this screen.

### Mole List Screen (mole\_list\_screen)

This screen shows users all their tracked moles, displaying representative images alongside key characteristics that help distinguish one mole from another.

### Mole Detail Screen (mole\_detail\_screen)

The mole detail screen focuses on a single mole, providing users with comprehensive information and all annotations related to that specific mole.

## Privacy & Security

- All data is stored locally on the device
- No cloud storage or external data transmission
- Private photo gallery separate from device's main photo library
- Support for guest accounts for anonymous usage

## Data Organization

```
/app_documents/  
└─ users/  
    └─ guest/  
        ├── photos.json  
        ├── campaigns.json  
        ├── moles.json  
        └─ campaigns/  
            ├── campaign_001/  
            │   ├── photo1.jpg  
            │   └─ photo2.jpg  
            └─ campaign_002/  
    └─ [username]/  
        ├── photos.json  
        ├── campaigns.json  
        ├── moles.json  
        └─ campaigns/  
            └─ campaign_001/
```

## Getting Started

### Prerequisites

- Flutter SDK 3.8.1 or higher
- Dart SDK compatible with Flutter version
- Android Studio / Xcode for mobile deployment
- Device with camera capability

## Installation

1. Clone the repository
2. Run `flutter pub get` to install dependencies
3. Configure platform-specific settings if needed
4. Run `flutter run` to launch the application

## Building

- **Android:** `flutter build apk` or `flutter build appbundle`
- **iOS:** `flutter build ios`
- **Other platforms:** See Flutter documentation for specific build commands

## Medical Disclaimer

This application is designed for tracking and documentation purposes only. It does not provide medical diagnosis or replace professional medical advice. Users should:

- Consult healthcare providers for medical concerns
- Follow dermatologist recommendations for mole monitoring
- Use this app as a supplementary tool, not a medical device
- Seek immediate medical attention for concerning changes in moles