# **User Guide: Camera Gallery App**

## **Overview**

This mobile application allows users to:

* Take photos using front or back camera
* Save photos to a local gallery
* Navigate between Home, Camera, Gallery, and Profile screens

## **Camera Screen Features**

### **Taking a Photo**

1. Open the Camera screen
2. Point camera at desired subject
3. Tap "Take Picture" button

### **Camera Options**

* "Flip Camera": Switch between front and back cameras
* "Take Picture": Capture current view
* After taking a picture, choose to:
  + "Save Picture": Store in app's gallery
  + "Retake": Discard current photo and return to camera view

## **Permissions**

* Camera access required
* First-time users will be prompted to grant camera permissions

## **Camera Settings**

* Default camera facing: Back camera
* Can toggle between front/back cameras

# **Developer Documentation**

## **Project Structure**

* Framework: React Native with Expo
* Navigation: React Navigation (Bottom Tab Navigator)
* Database: SQLite (expo-sqlite)
* Camera: Expo Camera

## **Key Dependencies**

* expo-camera: Camera functionality
* expo-sqlite: Local database management
* expo-file-system: File handling
* react-native-gesture-handler: Navigation gestures
* @react-navigation/bottom-tab-navigator: Screen navigation

## **Database Management**

### **Images Table Schema**

sql

Copy

CREATE TABLE images (

id INTEGER PRIMARY KEY AUTOINCREMENT,

uri TEXT NOT NULL,

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP

)

## **Configuration**

### **app.json Configuration**

* Portrait orientation
* Light UI style
* Camera and storage permissions
* Platform-specific settings (iOS/Android)

## **Key Components**

### **CameraScreen**

* Manages camera state
* Handles image capture
* Saves images to local storage
* Manages camera permissions

### **Database Initialization**

javascript

Copy

useEffect(() => {

const initDatabase = async () => {

const database = await SQLite.openDatabaseAsync('imageDatabase');

await database.execAsync(`

CREATE TABLE IF NOT EXISTS images (

id INTEGER PRIMARY KEY AUTOINCREMENT,

uri TEXT NOT NULL,

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP

);

`);

};

initDatabase();

}, []);

## **Potential Improvements**

* Add image preview functionality
* Implement image deletion
* Create more robust error handling
* Add image metadata capture
* Implement cloud backup options

## **Debugging Tips**

* Check camera permissions
* Verify SQLite database connection
* Monitor file system storage
* Use console logging for tracking issues

## **Performance Considerations**

* Optimize image storage
* Implement lazy loading for gallery
* Manage database connection efficiently

# **Troubleshooting**

## **Common Issues**

1. Camera not working
   * Check device permissions
   * Restart application
   * Verify camera hardware
2. Image not saving
   * Confirm database connection
   * Check file system permissions
   * Verify storage availability

## **Technical Support**

Contact developer with:

* Device model
* OS version
* Exact error messages

CopyRetry

Claude does not have the ability to run the code it generates yet.