Ministry of Science and Higher Education of the Russian Federation
“Ural Federal University named after the first President of Russia B.N. Yeltsin”
Institute of Radio electronics and Information Technologies – RTF

Food Journal Application

Project Report

Prepared by: Meriem Chibani

Group: RIM-140930

Github Repository Link : https://github.com/meriem-chibani/food-jornal

Contents

Introduction

This report outlines the technical review and improvements made to the Food Journal application. The project involved addressing inconsistencies in database integration, authentication, and UI components to enhance functionality and user experience.

1. Review of Home Screen Implementation

1.1 Database Schema Corrections

- **Issue:** Mismatched table and field names (e.g., "journals" vs. "journal") caused query failures.
- **Fix:** Updated SQL queries to align with the correct schema, ensuring consistency in field references (e.g., imageUri, desc).

1.2 Camera Functionality Improvements

- **Issue**: Improper use of useState for camera references led to instability.
- **Fix**: Implemented useRef for stable camera reference management.

1.3 Permission Handling

- **Issue:** Redundant permission requests in the media library function degraded performance.
- **Fix:** Streamlined permission checks using useEffect for a single request.

1.4 Data Integrity

- **Issue:** SQL operations lacked user validation, risking unauthorized data access.
- **Fix:** Added userId validation to DELETE and UPDATE queries.

```
DELETE FROM journal WHERE id = ? AND userId = ?;
```

2. Database Integration and Optimization

2.1 Synchronous Database Operations

- **Issue**: Incorrect use of async/await with expo-sqlite caused runtime errors.
- **Fix**: Replaced with synchronous API calls for reliable execution.

```
const db = openDatabase('myDB.db');
db.transaction(tx => {
    tx.executeSql('...');
});
```

2.2 Initialization and Error Handling

- **Issue**: Unreliable database initialization led to query failures.
- **Fix**: Ensured proper initialization and added robust error handling.

2.3 Performance Enhancement

- **Fix**: Implemented Write-Ahead Logging (WAL) mode for improved performance:
  `tx.executeSql('PRAGMA journal_mode = WAL');`

3. Authentication Module Refinements

3.1 Input Validation

- **Issue**: Broken conditional logic in validation caused unreliable checks.
- **Fix**: Restructured validation blocks for accuracy.

3.2 Query Result Handling

- **Issue**: Incorrect assumption about query result structure led to errors.
- **Fix**: Adjusted logic to handle results as a flat array.

```
if (users.length > 0) {
    const userId = users[0].id;
}
```

3.3 UI Enhancements

- **Issue**: Keyboard overlapped input fields on mobile devices.
- **Fix**: Integrated KeyboardAvoidingView for better UX.

Conclusion and Future Work

The project successfully addressed critical issues in database integration, authentication, and UI functionality. Key improvements include:

- Aligning database schema with application logic.
- Optimizing camera and permission handling.
- Strengthening data integrity with user validation.

Recommendations:

To ensure the long-term stability and scalability of the application, it is recommended to adopt centralized schema management, which will prevent inconsistencies by maintaining a single source of truth for database structure. Additionally, expanding testing protocols to cover edge cases in SQL operations might help identify and resolve potential issues before they impact functionality.