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

Ural Federal University
Institute of radioelectronics and information technologies

Restaurant Choser Mobile Application

Student : Meriem Chibani

Group: RIM-140930

Executive Summary

This report evaluates the Restaurant Choser mobile application, focusing on error resolution and enhancements to improve functionality, usability, and code robustness. The project involved systematic error detection, correction, and proactive improvements to align with best practices in React Native development.

TABLE OF CONTENTS

1- Project Overview

The *Restaurant Choser* app is a React Native-based application designed for restaurant discovery and navigation. The review aimed to:

- Identify syntactic, runtime, and logical errors.
- Implement corrective measures.
- Enhance UI/UX and code maintainability.

Key tools/libraries:

- **React Navigation** (Tab navigation)
- **Expo SQLite** (Database operations)
- **PropTypes** (Component validation)

2. Error Identification Methodology

A multi-phase approach ensured thorough issue detection:

2.1 Environment Setup

- Cloned the repository and installed dependencies (npm install).
- Launched the development server (npm start) to verify baseline functionality.

2.2 Testing and Analysis

To ensure the application's reliability, a multi-faceted testing strategy was employed. Manual testing involved simulating user interactions, such as navigation and button clicks, while monitoring console logs for errors. A static code review was conducted to inspect component structures, validate props, and verify SQLite query patterns. Additionally, the implementation of key libraries like expo-sqlite and react-navigation was cross-checked against official documentation to confirm adherence to best practices. This comprehensive approach ensured robust error detection and resolution.

2.3 Documentation

All issues were logged with details (error type, reproduction steps, impact).

3. Error Analysis and Solutions

3.1 Component Styling Issue

- Component: CustomButton
- Issue: The width prop was defined but not applied in styling.
- Impact: Inconsistent button sizes.
- Fix: Integrated the width prop into the TouchableOpacity style

3.2 Navigation Configuration

- Issue: Unsupported tabBarPosition logic for iOS in createMaterialTopTabNavigator.
- Impact: Redundant code with no effect on iOS tabs (fixed to top).

- Solution: Switched to createBottomTabNavigator for bottom tabs on iOS.
- Removed platform-specific conditions for Material Top Tabs.

4. Enhancements and Optimizations

4.1 UI/UX Improvements

We centralized component styling for consistency and removed non-functional platform-specific navigation logic, simplifying maintenance and improving code clarity

4.2 Code Quality

To strengthen code reliability, we implemented PropTypes validation across all components, ensuring proper data types are passed throughout the application. We also optimized the codebase by reorganizing imports into a consistent structure and eliminating unused props, resulting in cleaner, more maintainable components with reduced potential for runtime errors.

4.3 Cross-Platform Consistency

- Standardized tab positioning (top for all platforms).
- Used Platform.OS selectively (e.g., padding adjustments).

5. Conclusion and Recommendations

The *Restaurant Choser* app now demonstrates improved stability, maintainability, and user experience. Key achievements:

- Resolved critical styling and navigation issues.
- Enhanced code scalability through modular practices.

Future Work:

- Implement automated testing (e.g., Jest).
- Expand SQLite error handling for robustness.
- Conduct user testing to validate UX improvements.