

Professor: Dr. SABAH MOHAMMED

Contents

Native Messages Directory App – README	1
Summary of the Native Messages Directory App.....	1
Home Screen.....	1
Directories Screen	2
Messages Screen	2
Assets Images	2
SQLite Database (Optional Future Integration)	3
Key Technologies Used	3
Explanation of Each Folder	3
Javascript File	3
ChatApp Database Overview	4
CRUD Messages Screen	4
How to Run	4
GitHub Repository Name.....	5
ScreenShot.....	6

Native Messages Directory App – README

Summary of the Native Messages Directory App

This Native Messages Directory App is a simple WhatsApp-style messaging app built with React Native. It allows users to view and manage categorized messages in a user-friendly interface. Here's a breakdown of its core features:

Home Screen

This is the landing screen of the app that welcomes users and allows them to navigate to the main directory. It features:

- A welcoming message with emoji styling.

Exercise 2 (React Native Message Directory (CS5450 MOBILE PROGRAMMING))

- A button to navigate to the **Directories** screen.
 - A footer displaying student information.
-

Directories Screen

This screen displays categorized message directories using a **SectionList**. Each section includes:

- Orders (Pending Orders, Returns)
- Support (Tickets)
- Promotions (Discount Offers)

Features:

- Icons for each directory and item.
 - Tap an item to open related messages.
 - Clean UI with color-coded sections.
 - Empty state message if no data is found.
-

Messages Screen

This is the chat-style interface for viewing and managing messages:

- Displays messages from the selected directory.
 - Allows sending, editing, and deleting messages.
 - Messages are styled based on the sender (You / Support).
 - Data is persisted using **AsyncStorage** for offline use.
 - Supports long-press context menu for edit/delete.
-

Assets Images

- Assets are static image files (e.g., .png, .jpg) stored in the project.
- They are placed inside the assets/images folder.
- Used to display icons, logos, backgrounds, etc.

Exercise 2 (React Native Message Directory (CS5450 MOBILE PROGRAMMING))

- Loaded in code using `require()` or Image component like:
-

SQLite Database (Optional Future Integration)

Although **AsyncStorage** is used currently, a basic setup with **react-native-sqlite-storage** is included to:

- Initialize a database (chatApp.db)
 - Create a messages table with id, chatTitle, sender, and text
 - Can be extended to fully replace AsyncStorage for improved data control and searchability.
-

Key Technologies Used

- React Native components: `SectionList`, `FlatList`, `useState`, `useEffect`
 - Navigation: `@react-navigation/native`, `@react-navigation/stack`
 - Data Storage: `@react-native-async-storage/async-storage`
 - Database (optional): `react-native-sqlite-storage`
-

Explanation of Each Folder

assets/ → Stores non-code resources like fonts, images, and JSON data.

database/ → Houses local database implementations (e.g., SQLite, Realm).

main/ → The heart of the project:

java/com/Ex2NativeMsgDir/ → Contains source code (Java/Kotlin files).

res/ → UI design elements like XML layouts, drawable icons, strings, themes.

AndroidManifest.xml → Defines permissions, activities, and configurations.

screens/ → Contains different UI screens or page-related components

Javascript File

✓ **database.js** → Handles SQLite database operations.

✓ **api.js** → Provides mock API responses for support queries.

Exercise 2 (React Native Message Directory (CS5450 MOBILE PROGRAMMING))

✓ `utils.js` → Contains helper functions (formatting, navigation, logging).

✓ `config.js` → Stores configuration values (API settings, debug options)

ChatApp Database Overview

The `chatApp.db` database stores chat messages in the `messages` table. Each message has an `id` (unique identifier), `chatTitle` (chat category like "You" or "Support"), `sender` (who sent the message), and `text` (message content). SQLite ensures message persistence and supports CRUD operations for dynamic interaction.

CRUD Messages Screen

- **Create Message:**

1. Type your message in the input box at the bottom.
2. Tap **Send Message** to add the message to the chat.

- **Edit Message:**

1. Long-press on the message you want to edit.
2. Select **Edit** from the options.
3. The message text will appear in the input box.
4. Modify the text and tap **Update Message** to save changes.

- **Delete Message:**

1. Long-press on the message you want to delete.
2. Select **Delete** from the options.
3. Confirm deletion by tapping **Delete** in the alert.

- **Cancel Action:**

- When the options menu (Edit/Delete) appears, tap **Cancel** to close it without any changes.
 - When editing a message, clear the input or press **Cancel** on your keyboard to stop editing.
-

How to Run

1. Clone the repo.

Exercise 2 (React Native Message Directory (CS5450 MOBILE PROGRAMMING))

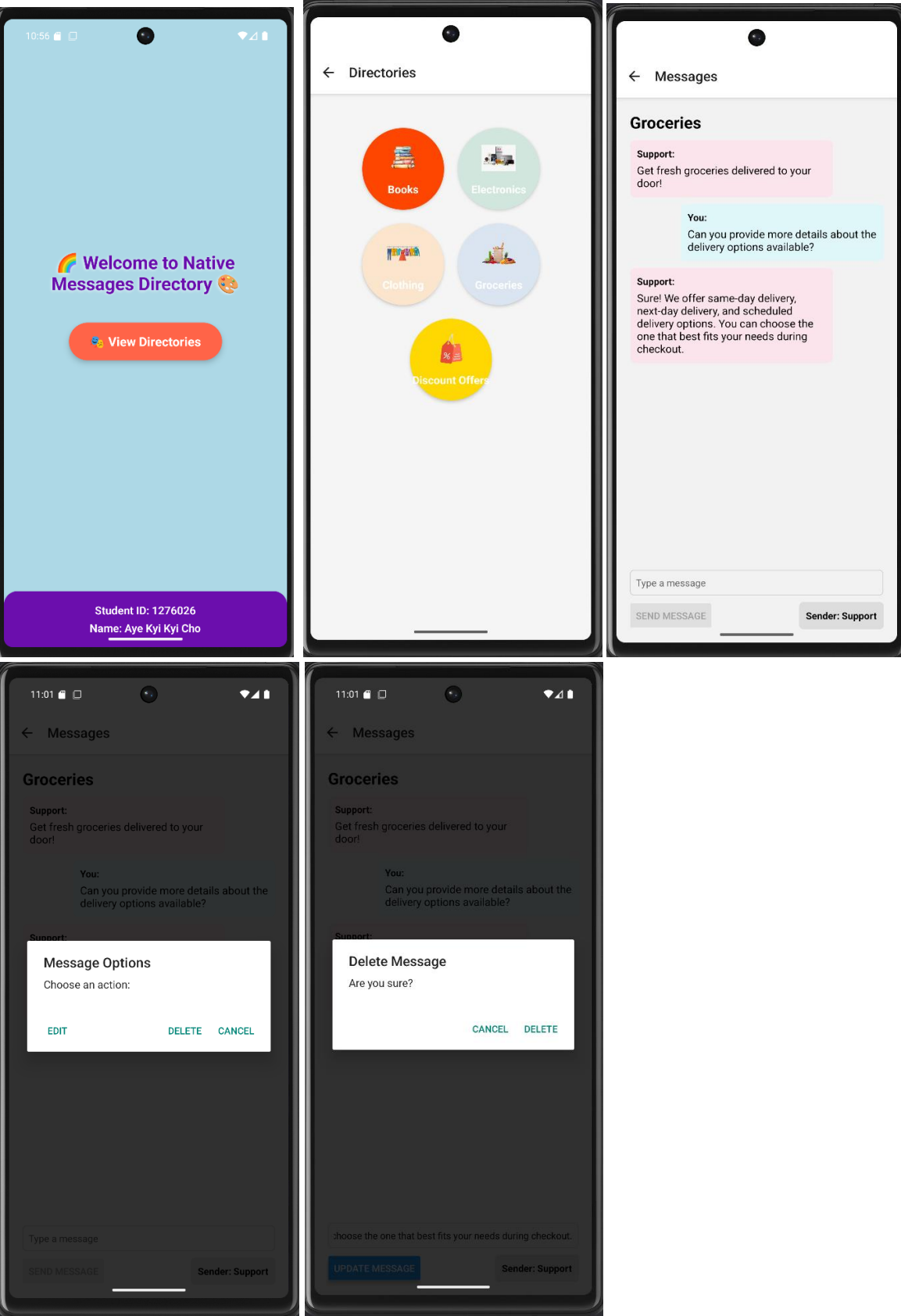
2. Run npm install.
3. Run the app using npx react-native run-android
4. Icons should be placed in assets/icons/ directory.

GitHub Repository Name

https://github.com/AyeKyiCho/msgdir_app_exercise2

Exercise 2 (React Native Message Directory (CS5450 MOBILE PROGRAMMING))

ScreenShot



Exercise 2 (React Native Message Directory (CS5450 MOBILE PROGRAMMING))