**Offline Messaging & Calling App - Project Report**

## ****1. Introduction****

### **1.1 Project Overview**

The Offline Messaging & Calling App is designed to enable peer-to-peer communication without relying on mobile networks or the internet. It leverages **Wi-Fi Direct** and **Bluetooth** technologies to facilitate text messaging and voice calls between nearby users. When an internet connection becomes available, the app can optionally sync messages with a cloud backup.

### **1.2 Objectives**

* Enable offline messaging using **Wi-Fi Direct** and **Bluetooth**.
* Support **voice calling** without a network.
* Ensure seamless **message synchronization** when an internet connection is available.
* Provide an intuitive **user interface** for easy navigation.
* Maintain **data security** and **privacy** for users.

### **1.3 Scope**

* **Primary Users**: People in remote areas, emergency responders, outdoor adventurers, and users facing network disruptions.
* **Platforms**: Android (initially), with potential expansion to iOS.
* **Technologies Used**: React Native, Node.js (backend), Firebase (cloud sync), WebRTC (optional for online calls).

## ****2. System Architecture****

### **2.1 Architecture Overview**

The app consists of three major layers:

1. **User Interface (Frontend)**: React Native-based mobile UI.
2. **Networking & Communication Layer**: Handles **Wi-Fi Direct**, **Bluetooth**, and optional WebRTC connections.
3. **Data Management Layer**: Local storage (AsyncStorage) for offline messages, Firebase for cloud sync.

📂 project-root/

├── 📂 src/

│ ├── 📂 components/ # UI components (buttons, inputs, etc.)

│ ├── 📂 screens/ # Home, Chat, Call, Settings screens

│ ├── 📂 services/ # Wi-Fi Direct & Bluetooth handlers

│ ├── 📂 utils/ # Helper functions for networking & storage

│ ├── App.js # Main entry point

├── 📂 backend/ # Optional backend for cloud sync

├── README.md # Documentation

### **2.2 Key Features**

* **Wi-Fi Direct Communication**: Establishes peer-to-peer connections.
* **Bluetooth Support**: Alternative communication method.
* **Message Status Indicators**: Sent, Delivered, Pending.
* **Voice Calling**: Direct VoIP calls over Wi-Fi Direct.
* **Cloud Synchronization**: Optional message backup when online.
* **User Profiles**: Editable name, avatar, and status.

## ****3. How This App Works (With Examples)****

### **Scenario 1: Festival or Protest (No Network Coverage)**

#### **Problem:**

A group of friends attends a crowded festival where mobile networks are overloaded. They can't send messages or make calls.

#### **How Our App Works:**

* **Mesh Network Activation**: Each person’s phone automatically connects to nearby phones using Wi-Fi Direct & Bluetooth.
* **Peer-to-Peer Messaging**: Alice types a message to Bob.
* **Message Relay**: If Bob is not within direct range, the message hops through other nearby users until it reaches him.
* **Voice Calls**: If Alice wants to call Bob, the app establishes a direct peer-to-peer call over Wi-Fi.

#### **Result:**

Alice and Bob stay connected without needing mobile data or SIM networks.

### **Scenario 2: Traveling Without Internet**

#### **Problem:**

John is traveling in a remote area without mobile coverage. He wants to send a message to his family.

#### **How Our App Works:**

* **Store-and-Forward**: John writes a message to his mom. The app holds the message.
* **Opportunistic Delivery**: When John’s phone comes near another user who is eventually connected to the internet (maybe at a Wi-Fi hotspot), the message gets forwarded.
* **Delayed but Guaranteed Delivery**: His mom receives the message once her phone connects to someone with internet access.

#### **Result:**

Even in remote areas, messages can still reach their destination when possible.

### **Scenario 3: Office or Apartment Chat Without Internet**

#### **Problem:**

A group of coworkers wants to chat without using their company’s internet or mobile data.

#### **How Our App Works:**

* **Local Wi-Fi Hotspot**: One person starts a Wi-Fi hotspot through the app. Others connect to it.
* **Instant Communication**: All connected users can send messages and make calls.
* **Secure & Private**: Since it's a direct connection, no external servers store data.

#### **Result:**

The team communicates freely without relying on any external service.

## ****4. UI Flow & Wireframe****

### **4.1 Screen Breakdown**

* **Welcome Screen**: Introduction and onboarding.
* **Home Screen**: Lists nearby users and active chats.
* **Chat Screen**: Displays messages, with options to send text or initiate calls.
* **Call Screen**: Real-time call interface with controls.
* **Settings Screen**: Connectivity toggles and profile customization.

### **4.2 Navigation Flow**

[Welcome Screen]

|

v

[Home Screen] <-------------------------------

| |

v |

[Chat Screen] <--> [Active Chats] |

| |

v |

[Call Screen] ---------------------> [Call Logs]

|

v

[Settings Screen] <--------------------------

## ****5. Implementation Plan****

### **5.1 Development Phases**

#### **Phase 1: Wireframing & UI Design**

* Create wireframes and mockups using Uizard/Figma.
* Finalize the **user experience (UX)** and **user interface (UI) components**.

#### **Phase 2: Core Functionality Development**

* Set up **Wi-Fi Direct** and **Bluetooth communication**.
* Implement **chat messaging** with status updates.
* Develop **voice calling feature**.

#### **Phase 3: Data Storage & Sync**

* Integrate **local storage (AsyncStorage)** for messages.
* Add **Firebase for cloud syncing**.
* Implement **encryption for secure data storage**.

#### **Phase 4: Testing & Debugging**

* Conduct **unit testing** for each module.
* Perform **peer-to-peer connection stability tests**.
* Optimize for **battery consumption and efficiency**.

#### **Phase 5: Deployment & User Feedback**

* Publish beta version for early users.
* Gather feedback and iterate on improvements.
* Deploy to **Google Play Store** (and later, **Apple App Store**).

## ****6. Future Enhancements****

* **Group Messaging** using **mesh networking**.
* **File Sharing** over Wi-Fi Direct.
* **Cross-Platform Support** (iOS version).
* **Custom Notification Sounds & Themes**.
* **Advanced Encryption & Privacy Controls**.

## ****7. Conclusion****

The Offline Messaging & Calling App provides an efficient, network-free communication platform. By leveraging **Wi-Fi Direct, Bluetooth, and opportunistic cloud syncing**, it ensures that users stay connected even in the most challenging environments. With further enhancements and user feedback, this project has the potential to become a robust alternative to conventional messaging services.

**Developed by: [Your Name/Team Name]**

🚀 **For contributions, visit [GitHub Repository Link]**

---------------------------------------------------------------------------------------------------------------------

Below is a visual wireframe in a text-based diagram format for offline messaging and calling app. I can use these diagrams as a blueprint to create high-fidelity mockups later.

**Offline Messaging & Calling App - Wireframe Overview**

**1. Welcome Screen**

+---------------------------------------------------+

| [App Logo] |

| |

| "Chat and call without mobile |

| networks!" (Tagline) |

| |

| [Get Started] |

| |

| [Small Settings Icon] (Top-right) |

+---------------------------------------------------+

**Notes:**

* The screen includes an app logo and tagline for branding.
* The “Get Started” button is central to guide the user forward.
* A settings icon in the top-right corner allows access to initial configurations.

**2. Home Screen**

+---------------------------------------------------+

| [Hamburger/Menu Icon] [App Logo] [Profile] |

+---------------------------------------------------+

| Nearby Users |

| -------------------- |

| [Avatar] User1 (Wi-Fi) → |

| [Avatar] User2 (Bluetooth) → |

| [Avatar] User3 (Wi-Fi) → |

| ... |

| |

| Active Chats |

| -------------------- |

| [Avatar] User1: "Last message preview..." (unread) |

| [Avatar] User4: "Hey, are you..." |

| ... |

+---------------------------------------------------+

| [Start New Chat] (Floating Button) |

+---------------------------------------------------+

| Navigation Bar: Home | Chats | Calls | Settings |

+---------------------------------------------------+

**Notes:**

* Displays a **menu, app logo, and profile icon** in the top bar.
* **Nearby Users** are discovered via **Wi-Fi Direct/Bluetooth** with connection indicators.
* **Active Chats** section lists recent conversations.
* **Navigation Bar** allows quick switching between screens.

**3. Chat Screen**

+---------------------------------------------------+

| [Back] [User's Profile Pic] User Name |

+---------------------------------------------------+

| Chat History |

| ------------------------------------------------ |

| [Sent] "Hi there!" [Time] |

| ------------------ |

| [Received] "Hello!" [Time] |

| ------------------ |

| (Messages aligned left/right for sender/receiver)|

| Status Icons: Pending (🕒), Sent (✔), Delivered (✔✔)|

| ------------------------------------------------ |

+---------------------------------------------------+

| [Text Input Field] [Send Button] [Voice Call] |

+---------------------------------------------------+

**Notes:**

* Displays **profile picture and name** of the chat partner.
* Shows **message history with timestamps and delivery statuses**.
* Users can **send messages or start voice calls**.

**4. Call Screen**

+---------------------------------------------------+

| [Back] Call with User Name |

+---------------------------------------------------+

| Call Status: Ringing... |

| |

| [Timer] |

| |

| [Mute Button] [Speaker Button] |

| |

| [End Call] |

+---------------------------------------------------+

**Notes:**

* Displays **caller information and call status**.
* Includes **Mute, Speaker, and End Call** buttons.
* The call **timer tracks duration**.

**5. Settings Screen**

+---------------------------------------------------+

| [Back] Settings |

+---------------------------------------------------+

| Connectivity Controls: |

| ----------------------- |

| [ ] Wi-Fi Direct (Toggle) |

| [ ] Bluetooth (Toggle) |

| [ ] Sync Messages (Toggle: On/Off) |

| |

| User Profile: |

| ----------------------- |

| [Avatar] Nickname: [Edit Field] |

| Status Message: [Edit Field] |

| |

| App Theme: [Light / Dark Mode Toggle] |

| |

| About: |

| ----------------------- |

| Version: 1.0 |

| Developer Info, Contact, etc. |

+---------------------------------------------------+

**Notes:**

* Allows users to toggle **Wi-Fi Direct, Bluetooth, and Message Syncing**.
* Profile settings include **nickname, avatar, and status message**.
* Users can switch **between Light and Dark mode**.

**Navigation Flow Diagram**

[Welcome Screen]

|

v

[Home Screen] <-------------------------------

| |

v |

[Chat Screen] <--> [Active Chats] |

| |

v |

[Call Screen] ---------------------> [Call Logs]

|

v

[Settings Screen] <--------------------------

**Flow Description:**

* **Welcome Screen:** User lands here and taps "Get Started".
* **Home Screen:** Displays Nearby Users and Active Chats.
* **Chat Screen:** Allows **text messaging and calls**.
* **Call Screen:** Handles **voice calls**.
* **Settings Screen:** Manages **connectivity and profile settings**.

**This wireframe serves as a text-based blueprint for designing high-fidelity UI mockups and developing the app efficiently.**