



Bluetooth chat Application using Java and Android studio

DONE BY

V.Eswar sai,D.Roshini,K.sivaji,K.Yeswanth kumar,T.Manohar,Subham kumar

Students of Visakha Institute of Engineering and Technology

Computer Science Engineering

Visakhapatnam,Andhra pradesbh

Under the guidance of

SOUJANYA

Faculty of Computer Science Engineering

Visakhapatnam,Andhra pradesbh

Under the guidance of

A.S.C TEJASWINI KONE

HOD of Computer Science Engineering

Visakhapatnam,Andhra pradesbh

Abstract

In the era of mobile communication, Bluetooth technology has become ubiquitous, enabling seamless wireless data exchange between devices. This project proposes the development of a Bluetooth chat application using Java and Android Studio. The application aims to provide a platform for users to exchange messages in real-time over Bluetooth connections.

The proposed application will leverage the Android platform's Bluetooth API to facilitate communication between devices. Users will be able to discover nearby Bluetooth-enabled devices, establish connections, and exchange text messages within a user-friendly chat interface.

experience by configuring settings such as notification preferences, message fonts, and chat themes.

Introduction

Bluetooth is a wireless communication technology that enables the wireless transmission of data over short distances. Bluetooth can communicate with 2.4 GHz using radio frequencies. Bluetooth technology can reach up to 10 meters of shooting capacity in an open area. Data transfer speeds between 1 Mbps and 721 Kbps are achieved in transfers with Bluetooth.

“Bluetooth Chat application” is an android application that is used to send and receive messages. We can chat with our friend if we are in Bluetooth range and don't have access to the internet. Also, teachers will able to send notes to multiple students via Bluetooth Chat

application and able to teach them. It will be helpful to students if there is no wi-fi in your school, for travelers to chat between tents in mountains, and in many other cases.

Bluetooth is a technology designed by Ericsson in 1994 to wirelessly connect devices. It is used to exchange data among devices within short range, usually 10 metres. Bluetooth

operates in the frequency range 2400 MHz to 2483.5 MHz which is the ISM unlicensed band. Bluetooth uses the concept of frequency hopping to avoid collisions. It usually performs 1500 hops per second[3].

Bluetooth exchange data using the concept of packets. The architecture used by Bluetooth is master-slave architecture. One master can communicate with maximum 7 slaves in a piconet structure. In order to connect more than 7 slaves a scatternet needs to be formed. When more than one slave is connected then bluetooth operates in round-robin fashion. Typically, the master switches from one device to another for a fixed period of time[3].

There are several applications of bluetooth out of which few are mentioned below:

- a) Connecting two nearby computers for data exchange.
- b) Wireless communication with tablets and speakers such as iOS and Android devices.
- c) Sending small advertisements from Bluetooth-enabled advertising hoardings to other, discoverable, Bluetooth devices.
- d) Transfer of files, contacts and other information.
- e) Wireless control of and communication between a mobile phone and a handsfree headset.

Some of the limitations of Bluetooth technology are mentioned below:

- a) As compared to Wi-Fi, which transfer data at 54 Mbps, Bluetooth comparatively slower and its transfer rate is about 3 Mbps.
- b) Distance Limitations: For a bluetooth hardware device installed in our computers or mobile devices, the range is upto 30 feet which sometimes be too short.
- c) Interference: Since bluetooth operates in the unlicensed frequency spectrum, interference among Bluetooth signals is possible.

Objective:

1. **Enable Wireless Communication:** Facilitate seamless communication between Android devices without the need for an internet connection, relying solely on Bluetooth technology for data exchange.

2. **User-Friendly Interface:** Design an intuitive and visually appealing user interface (UI) that makes it easy for users to compose and send messages, view received messages, and manage their chat conversations.

3. **Real-Time Messaging:** Enable real-time messaging capabilities, allowing users to send and receive messages instantly, with minimal latency, creating an interactive chat experience.

4. **Bluetooth Connectivity:** Implement robust Bluetooth connectivity features to enable device discovery, pairing, and connection establishment between nearby devices. Ensure compatibility with a wide range of Android devices and Bluetooth protocols.

5. **Message Handling:** Develop mechanisms for encoding, transmitting, receiving, and displaying messages within the chat application. Handle various message formats, including text messages, emojis, and multimedia attachments.

6. **Security and Privacy:** Prioritize the security and privacy of user data by implementing encryption algorithms and authentication mechanisms to secure message transmission over Bluetooth connections. Prevent unauthorized access to message content and protect user privacy.

7. **Offline Messaging:** Support offline messaging functionality by caching messages when devices are out of range or disconnected, and delivering them once the connection is re-established. Ensure seamless message synchronization between devices.

8. **Error Handling and Resilience:** Implement error handling mechanisms to address common Bluetooth communication issues, such as connection drops, signal interference, and device unavailability. Provide informative error messages and graceful recovery strategies to enhance user experience.

Customization Options: Offer customization options to users, allowing them

to personalize their chat experience by configuring

1. settings such as notification preferences, message fonts, and chat themes.

2. Scalability and Reliability: Build a scalable and reliable chat application architecture that can handle increasing user load and maintain consistent performance under varying network conditions. Ensure backward compatibility with older Android versions and support for future updates.

HARDWARE&SOFTWARE REQUIREMENT

Hardware Requirements:

1. Android Device: You'll need at least one Android device to run and test your Bluetooth chat application. It's recommended to have multiple devices for testing communication between them.

2. Bluetooth Module (Optional): If you want to test Bluetooth communication with external hardware like Arduino or other Bluetooth-enabled devices, you might need a Bluetooth module compatible with your hardware.

Software Requirements:

1. Android Studio: Android Studio is the official Integrated Development Environment (IDE) for Android app development. Make sure you have it installed on your development machine.

2. Java Development Kit (JDK): Android development primarily uses Java. Ensure that you have the latest JDK installed on your system.

3. Android SDK: Android SDK is necessary for Android app development. Android Studio usually installs it automatically, but you should ensure that it's up to date.

4. Bluetooth API: Android provides Bluetooth API to develop Bluetooth-enabled applications. Familiarize yourself with Bluetooth API documentation provided by Android.

5. Android Virtual Device (AVD) or Physical Android Device: You can use either an

Android Virtual Device provided by Android Studio or a physical Android device connected to your development machine for testing your application.

6. Text Editor or IDE: While Android Studio is the primary tool for development, having a good text editor or IDE for writing and editing code can be helpful. Examples include Visual Studio Code, IntelliJ IDEA, etc.

Software Setup:

1. Android Studio: Install Android Studio on your development machine. Ensure that you have the latest version installed, along with the necessary SDK components and platform tools.

2. Java Development Kit (JDK): Make sure you have the Java Development Kit (JDK) installed on your development machine. Android development primarily uses Java for coding.

3. Android SDK: Android Studio usually installs the Android SDK automatically. However, ensure that the SDK is up to date and contains the necessary tools, platform, and system images.

Development Environment:

1. Project Setup: Create a new Android project in Android Studio for developing the Bluetooth chat application. Configure the project settings, including package name, minimum SDK version, and target SDK version.

2. Bluetooth Permissions: Declare the necessary Bluetooth permissions in the AndroidManifest.xml file to enable Bluetooth functionality in the application. Permissions may include `BLUETOOTH`, `BLUETOOTH_ADMIN`, and `ACCESS_FINE_LOCATION`.

3. User Interface (UI): Design the user interface for the chat application using XML layout files in Android Studio. Create UI elements such as text fields, buttons, message bubbles, and chat lists to facilitate messaging interactions.

4. Bluetooth Functionality: Implement Bluetooth connectivity features using Java code. Utilize Android's Bluetooth API to discover nearby devices, establish Bluetooth connections, and exchange messages between devices.

5. **Message Handling:** Develop logic for encoding, transmitting, receiving, and displaying messages within the chat application. Implement message queues, parsers, and UI adapters to handle message interactions effectively.

6. **Testing:** Deploy the application to the Android devices connected to your development machine or use Android Virtual Devices (AVDs) for testing. Conduct thorough testing to ensure that the application functions correctly under various scenarios, including device pairing, message exchange, and error conditions.

7. **Debugging and Optimization:** Use Android Studio's debugging tools to identify and fix any issues encountered during testing. Optimize the application for performance, battery efficiency, and compatibility with different Android device configurations.

External Hardware Setup (Optional):

1. **Bluetooth-enabled Devices:** If testing communication with external hardware, ensure that the devices are powered on and discoverable. Pair the devices with your Android devices through Bluetooth settings.

2. **Arduino or Other Devices:** Connect the Bluetooth modules to your external hardware (e.g., Arduino) and program them to communicate with the Android application using Bluetooth protocols. Test communication between the Android devices and the external hardware.

Result:

Setup Bluetooth Permissions: In your AndroidManifest.xml file, you need to add permissions for Bluetooth communication. These permissions include `<uses-permission android:name="android.permission.BLUETOOTH"/>` and `<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>`.

Design UI: Design the layout for your chat interface in Android Studio. You'll need text fields for displaying messages, an input field for typing messages, and buttons for sending messages and managing Bluetooth connections.

Implement Bluetooth Connectivity: Use the Android Bluetooth API to discover nearby Bluetooth devices, establish connections, and

manage data transfer. This involves using classes like `BluetoothAdapter`, `BluetoothDevice`, `BluetoothSocket`, etc. You'll need to handle Bluetooth permissions and runtime permissions if your app targets Android 6.0 (API level 23) or higher.

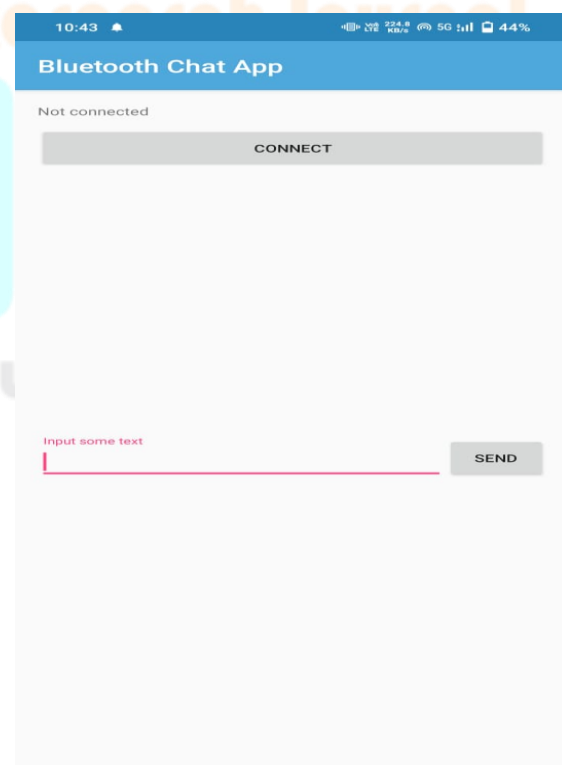
Create Chat Logic: Implement the logic for sending and receiving messages over Bluetooth. This includes encoding messages, handling message sending and receiving events, and updating the UI accordingly.

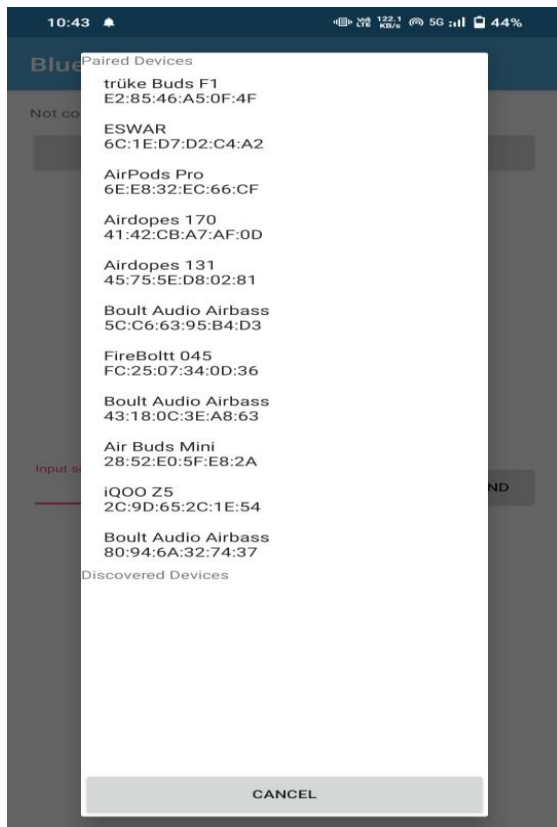
Handle Device Discovery: Implement device discovery functionality to find nearby Bluetooth devices that can be connected to for chatting.

Handle Connection Events: Manage the connection lifecycle, including connecting to devices, handling connection failures, and disconnecting from devices.

Test and Debug: Test your application on real devices and emulators to ensure it works as expected. Debug any issues that arise during testing.

Output:





Conclusion:

By setting up the experimental environment as outlined above, you can effectively develop, test, and validate the Bluetooth chat application using Java and Android Studio. Ensure thorough testing across different devices and scenarios to verify the functionality, reliability, and performance of the application before deployment.

References:

- Ayabe, B. S., Chander, S. S., & Mizikovsky, S. B. (2000). U.S. Patent No. 6,141,550. Washington, DC: U.S. Patent and Trademark Office.
- Rittman, D., & Schnapp, M. (2004). U.S. Patent Application No. 10/974,989.
- Wu, J., Huo, M., Cai, J., Wu, M., & Wang, Y. (2012, June). Research on Bluetooth expansion of communication based on android system. In World Automation Congress (WAC), 2012 (pp. 1-4). IEEE.
- O'sullivan, J. D., Daniels, G. R., Percival, T. M., Ostry, D. I., & Deane, J. F. (1996). U.S. Patent No. 5,487,069. Washington, DC: U.S. Patent and Trademark Office.
- Padgette, J., Scarfone, K., & Chen, L. (2012). Guide to Bluetooth Security: Recommendations of the National Institute of Standards and Technology (Special Publication 800-121 Revision 1).
- Church, K., & de Oliveira, R. (2013, August). What's up with whatsapp?: comparing mobile instant messaging behaviors with traditional SMS. In Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services (pp. 352-361). ACM.
- Yardeni, Z., & Londono, J. (2007). U.S. Patent Application No. 11/796,304