SereniTinnitus App Documentation with 3012 words

**\*\*1. Core Features Overview\*\***

The SereniTinnitus app is designed to assist users in managing tinnitus symptoms through therapeutic sound therapy, tinnitus severity assessment, AI-based chatbot assistance, and doctor referrals. Each module is developed to provide a seamless and user-friendly experience, ensuring users can access relevant resources efficiently.

**\*\*1.1 Therapeutic Sound Library\*\***

The app includes a therapeutic sound library categorized into natural sounds and Islamic content.

1. Natural Sounds (activity\_soothingsounds.xml):
2. Beach Waves: Mimics the soothing sound of ocean waves.
3. Rainfall: Provides a calming rain sound, ideal for relaxation.
4. Flowing River: A continuous stream sound to help mask tinnitus.
5. Night Bonfire: Crackling fire sound, commonly used for white noise therapy.
6. Islamic Content (activity\_soundformuslims.xml):
7. Quran Recitations: Users can listen to calming Quranic recitations.
8. Nasheeds: Islamic vocal music that can provide spiritual relaxation.
9. Duas: Special prayers for healing and peace.

**Key Components & Implementation:**

1. MediaPlayer API: Handles audio playback functionalities.
2. RecyclerView Implementation: Allows dynamic sound listing via adapters like quranAdapter.java.
3. Unified Playback UI: Reuses UI components such as activity\_beach.xml for consistency.
4. Custom Controls: Includes seek bars for volume adjustment, play/pause buttons, and next/previous navigation.

**\*\*1.2 Tinnitus Handicap Inventory (THI) Assessment\*\***

The THI assessment helps determine the severity of tinnitus by evaluating user responses to a 25-question survey.

1:Scoring Criteria (Questionnaire.java):

0-25 (Mild): Users with mild symptoms are recommended sound therapy.

26-50 (Moderate): Users with moderate symptoms are suggested a combination of therapy and self-help techniques.

51-75 (Severe): Users may need medical consultation along with therapy.

76-100 (Very Severe): Direct doctor consultation referral.

2:Result Handling:

For mild/moderate cases: Redirects to TherapyoptionsResult.java.

For severe cases: Redirects to DoctorCitiesActivity.java for specialist referral.

PDF Report Generation (Result.java): The iTextPDF library is used to generate downloadable assessment reports for doctors and personal reference.

**\*\*1.3 Doctor Referral System\*\***

The Doctor Referral System connects users with ENT specialists based on their location.

Process Flow:

1. City Selection (DoctorCitiesActivity.java)

2. ENT Specialist List Display (DoctorListActivity.java)

3. Doctor Profile Cards (item\_doctor.xml)

Displays name, specialization, clinic address, and contact details.

Additional Features:

Search functionality: Users can search doctors by name or location.

Integration with Google Maps: Allows users to view the clinic location.

Click-to-Call Option: Users can directly call the doctor's office.

**\*\*1.4 AI Chatbot\*\***

The AI-based chatbot provides instant assistance to users by integrating the Groq API.

Technical Implementation:

API Model: Uses llama3-8b-8192 model in ChatbotAPIHandler.java.

SharedPreferences: Saves chat history for persistent conversation.

UI Components:

User/Bot Message Bubbles (item\_user\_message.xml)

Quick-Select Options (item\_option.xml)

Example Chatbot Conversations:

User: "Play Surah Al-Fatiha"

Chatbot: "Launching Quran Recitation..." (Triggers quransounds.java)

User: "My THI score is 82"

Chatbot: "You should consult an ENT specialist." (Redirects to DoctorCitiesActivity)

\*\***2. Detailed Module Breakdown\*\***

**\*\*2.1 Audio Playback System\*\***

Backend Implementation:

1. MediaPlayer API: Manages streaming and playback.
2. AudioManager API: Controls volume levels.
3. Metadata Models (e.g., quranModel.java): Store track details.

Frontend Implementation:

1. Seek Bars: Adjust volume and progress.
2. Playback Controls: Play, pause, stop, next, previous.

\***\*2.2 THI Assessment Engine\*\***

Database Schema (SQLite):

CREATE TABLE UserResponses (

response\_id INTEGER PRIMARY KEY,

user\_id INTEGER FOREIGN KEY,

question\_id INTEGER,

score INTEGER,

timestamp DATETIME

);

Scoring Logic (Java):

if (totalScore >= 76) {

startActivity(new Intent(this, DoctorCitiesActivity.class));

} else {

startActivity(new Intent(this, TherapyoptionsResult.class));

}

**\*\*2.3 User Authentication System\*\***

Flow:

1. Splash Screen → Check Login State

2. If Logged In → Home Activity

3. If Not Logged In → Redirect to Login Activity

**Security Features:**

Password Requirements: Minimum 8 characters, including a number.

Session Timeout: Auto-logout after 30 minutes of inactivity.

\*\*3. User Workflows\*\*

\*\*3.1 New User Onboarding\*\*

1. Sign Up (Signup.java):

Email/password validation

SQLite account storage

2. Initial THI Assessment

User completes questionnaire.

Severity report is displayed.

\*\*3.2 Sound Therapy Session\*\*

1. Select Sound Category (e.g., Rain Sounds).

2. Choose Track from RecyclerView.

3. Playback Controls

Volume (seekbarvol).

Progress (seekbartime).

\*\*3.3 Doctor Appointment Booking\*\*

1. Select City → View ENT List.

2. Tap Doctor Profile → See Contact Info.

3. Chatbot Assistance: "How to book Dr. Ali?"

\*\*4. Technical Architecture\*\*

\*\*4.1 Class Diagram\*\*

MediaPlayer.java: start(), pause(), seekTo()

DatabaseHelper.java: insertUser(), getTHIScore()

ChatbotAPIHandler.java: sendMessage()

\*\*4.2 Key Dependencies\*\*

Library: Glide

Purpose: Image & GIF handling

Version: 4.12.0

Library: iTextPDF

Purpose: PDF report generation

Version: 7.1.15

Library: OkHttp

Purpose: API Calls

Version: 4.9.3

\*\*5. Chatbot Integration Guide\*\*

\*\*5.1 Training Scenarios\*\*

User Query: "Play Surah Al-Fatiha"

Bot Response: "Launching..."

Action: Opens quransounds.java

User Query: "My THI score is 82"

Bot Response: "Consult ENT"

Action: Opens DoctorCitiesActivity

User Query: "Reset Password"

Bot Response: "Follow steps..."

Action: Triggers ForgotPasswordDialog

\*\*5.2 Error Handling\*\*

try {

// API call logic

} catch (IOException e) {

showToast("Network error. Retry later.");

}

\*\*6. Security & Compliance\*\*

\*\*6.1 Risk Mitigation\*\*

Risk: Hardcoded API Key

Solution: Store in Android Keystore

Risk: Plaintext Passwords

Solution: Implement SHA-256 Hashing

\*\*6.2 Permissions\*\*

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.INTERNET"/>

\*\*7. Appendix: File Index\*\*

\*\*7.1 XML Files\*\*

activity\_chatbot.xml: Chatbot UI

item\_doctor.xml: Doctor Profile Card

\*\*7.2 Java Classes\*\*

DatabaseHelper.java: insertUser(), saveTHIResponse()

Result.java: generatePDF(), calculateSeverity()

---