Jonda Mobile App – Technical Design Documentation

Mohamed Faizal
md.faizal.md.j@gmail.com

Table of Contents

Architecture Overview

System Architecture

The Jonda mobile application follows a client-server architecture with the following components:

- **Frontend**: React Native with Expo framework
- **Backend**: Node.js server with Express.js
- **Payment Processing**: Stripe API integration
- **File Storage**: IRIS
- **AI Services**: External API for chatbot responses (JondaX Engine)
- **Voice Processing**: Native speech recognition APIs

Technology Stack

Frontend Technologies

Technology	Version	Purpose
React Native	Latest	App framework
Expo	Latest	Development platform
TypeScript	Latest	Type safety and development
Expo Router	Latest	File-based navigation
Linear Gradient	Latest	UI gradient effects

Backend Technologies

Technology	Purpose
Node.js	Server runtime
Express.js	Web framework
Stripe SDK	Payment processing

Third-Party Services

Service	Integration	Purpose
Stripe	React Native SDK	Payment processing
Speech Recognition	Expo module	Speech-to-text
Document Picker	Expo module	File selection
Image Picker	Expo module	Camera & gallery access

Core Components

1. **Authentication System Template (index.tsx)**
   Intended Purpose: Secure user authentication with form validation
   Current implementation: Username/password input fields (no validation)

2. **Home Dashboard (home.tsx)**
   Purpose: Central hub for document upload & app navigation
   Key Features:
   - Multiple upload methods support (camera/gallery image, document)
   - Stripe payment integration for JondaX engine
     - Supports card, paynow, grabpay and more
   - Central Router (E.g. navigation to Chatbot)

   Upload Flow:
   a. User selects upload method
   b. Permission requests (camera/storage)
   c. File selection/capture
   d. Upload to server with loading indicator
   e. Navigation to record page (record.tsx)
      i. Displays harmonised data in a list unpacked from a JSON

3. **AI Chatbot (chatbot.tsx)**
   Purpose: Conversational AI interface with voice and text support
   Key Features:
   1. Real-time messaging with timestamps
   2. Speech-to-text integration
   3. Gradient message bubbles

   Future Implementation: Chat history and saving

Core Project Packages

expo-speech-recognition
The expo-speech-recognition package is the only free expo/react native package that works for development build using iOS and android native modules to recognise speech.

Note: Paid alternatives are a better option, since this package is an external community package and not an official one.

Official documentation: https://github.com/jamsch/expo-speech-recognition
This extensive documentation provides all necessary information to implement this package to fit your use case.

Expo document and image picker
A standard expo package used to retrieve documents and images or capture an image via user's camera. This module requires permission to access files, gallery and camera.

Note: For iOS platform, permissions need to be set up in Info.plist which can be found in jonda-app/ios/jondaapp/Info.plist. For more information visit here.

Stripe API integration

File location: backend-server/server.js
Endpoints:

1. POST /create-payment-intent (Creates Stripe payment intent)
2. GET /config (Returns Stripe publishable key)

An account setup is required to generate keys to use stripe. Create one here.
You may find additional information on customising Stripe payment gateway to your liking here.
Payment modes and other setup can be configurable on the Stripe account dashboard itself.

Client-side API Management

File location: rest/requester.ts
RESTful Functions:

1. uploadImage(): uploads image to JondaX
2. uploadDocument(): uploads document to JondaX
3. getDadJoke(): simulates responses from Jonda chat bot

Setup, Usage and References

Please read the readme file located in the project folder for more information on first-time setup, running and more information on building the project along with links to packages used in this project.