Implementation Notes:

1. Component Architecture:
   - Main ChessGame component acts as container
   - Modular services for API and language
   - Separate concerns for state management

2. State Management:
   - Uses React useState for game state
   - Periodic board updates with useEffect
   - Centralized state in ChessGame

3. API Integration:
   - RESTful service calls
   - Guest game functionality
   - Move validation on server
   - Board state synchronization

4. Internationalization:
   - Language provider integration
   - Translation key system
   - Dynamic text rendering

5. Key Features:
   - Guest play mode
   - Real-time board updates
   - Piece capture tracking
   - Move validation
   - Multi-language support

The architecture follows a modified MVC (Model-View-Controller) pattern adapted for a React-based web application.

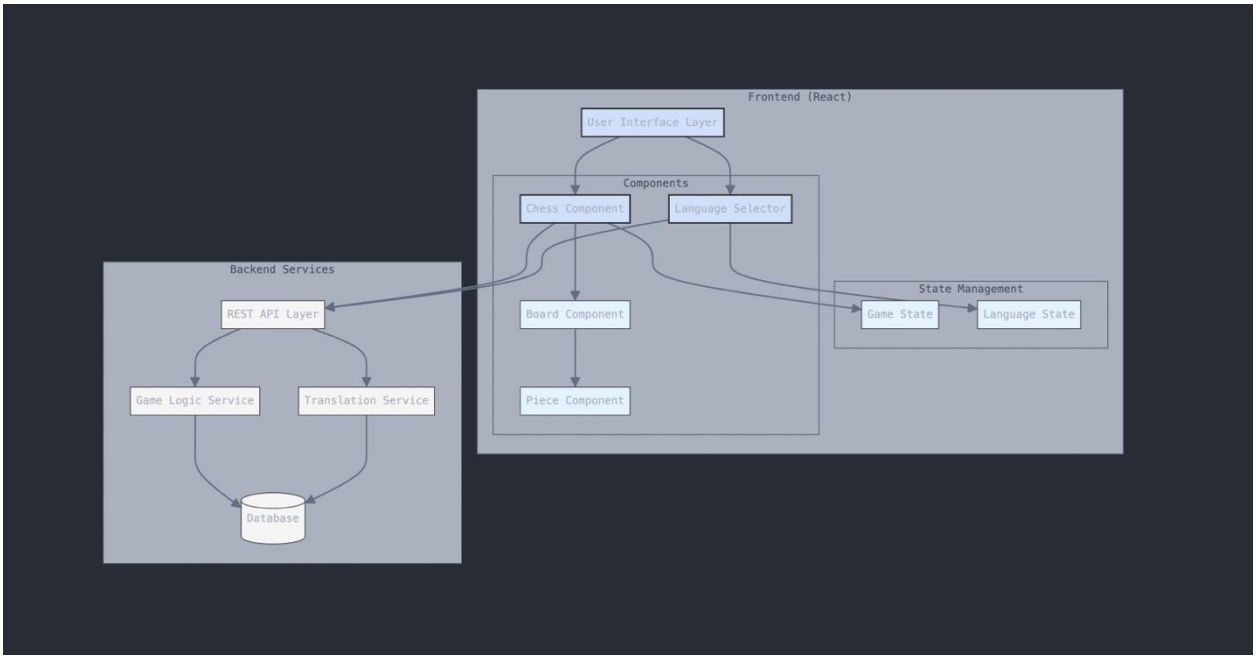
1. Frontend Layer (View):
   - User Interface Layer: Main container for all React components
   - Components:
     - Chess Component: Main game controller
     - Language Selector: Handles language switching
     - Board Component: Manages chess board state
     - Piece Component: Individual chess piece logic

2. State Management (Model):
    - Game State: Manages chess game state using React useState
    - Language State: Handles language preferences and translations

3. Backend Services (Controller):
    - REST API Layer: Handles communication between frontend and services
    - Game Logic Service: Implements chess rules and validations
    - Translation Service: Manages multilingual support
    - Database: Stores game states and language data

Key Architectural Features Covered:

1. Component-Based Architecture:
    - Modular components for reusability
    - Clear separation of concerns
    - Hierarchical component structure
2. State Management:
    - Centralized state management
    - Immutable state updates
    - Clear data flow patterns
3. Service Layer:
    - RESTful API design
    - Microservices architecture
    - Scalable backend services
4. Data Flow:
    - Unidirectional data flow
    - Event-driven architecture
    - Clear service boundaries


Frontend (React)
User Interface Layer
Components
Chess Component
Language Selector
Board Component
Piece Component
State Management
Game State
Language State
Backend Services
REST API Layer
Game Logic Service
Translation Service
Database