💣 Bomb Flip Betting Game - Flutter Mobile App Specification

📱 Project Overview

Platform: Flutter Mobile App (iOS & Android)
Game Type: Lottery-style card flipping game with betting mechanics
Backend: Existing Django REST API
Timeline: 4-5 weeks development

🎮 Game Concept

What Players Do

1. **Place Bet**: Choose stake amount (~~₦~~200-~~₦~~1000)
2. **Flip Cards**: Tap numbered cards (like lottery tickets) to reveal safe cards or bombs
3. **Build Multiplier**: Each safe card increases payout multiplier
4. **Cash Out**: Secure winnings before hitting a bomb, or lose everything

Core Mechanics

- **25 numbered cards** arranged in 5x5 grid (like lottery tickets)
- **Higher stakes = more bombs** (3% to 40% bomb rate)
- **Higher stakes = Higher Multipliers** (0.05x to 0.2x Win multiplier)
- **Progressive rewards**: Each safe card adds 0.05x to multiplier (Based on the multiplier)
- **Minimum 2 flips** required before cashing out
- **All-or-nothing**: Hit bomb = lose entire stake

🎯 Game Logic Overview

Bomb Rate System

- **Low Stakes (~~₦~~200-~~₦~~500)**: 3-8% bombs (very safe)
- **Medium Stakes (~~₦~~500-~~₦~~2000)**: 8-25% bombs (moderate risk)
- **High Stakes (~~₦~~2000-~~₦~~5000)**: 25-40% bombs (high risk)
- **Above ~~₦~~5000**: Capped at 40% maximum
- **Randomization**: ±5% variation each game for unpredictability

Multiplier System

- **Starting**: 1.00x multiplier
- **Growth**: +0.05x for each safe card flipped
- **Examples**:
  - 5 safe cards = 1.25x multiplier
  - 10 safe cards = 1.50x multiplier
  - 20 safe cards = 2.00x multiplier
- **Payout**: Stake × Final Multiplier

Board Generation

- **Grid**: 5x5 = 25 numbered cards (1, 2, 3... 25)
- **Bomb Placement**: Each card independently evaluated for bomb placement
- **No Guarantees**: Actual bomb count varies around expected rate
- **Example**: 20% bomb rate might produce 3-7 actual bombs

📱 App Screens & Features

Main Screens

1. **Setup Screen**: Enter name, choose stake, see bomb rate, start game
2. **Game Screen**: 5x5 grid of numbered cards, game controls, stats display
3. **Results Screen**: Win/loss message, return to setup

Visual Design

- **Theme**: Premium lottery ticket aesthetic with gold/yellow colors
- **Cards**: Numbered 1-25 with fancy Orbitron font
- **States**:
  - Unflipped: Gold numbered card
  - Safe: Green with ✅ checkmark
  - Bomb: Red with 💣 explosion
- **Animations**: Smooth card flip transitions

Key UI Elements

- **Wallet Display**: Shows current balance (₦10,000 starting)
- **Stake Input**: Slider or input for bet amount
- **Bomb Rate Display**: Live calculation based on stake
- **Game Stats**: Current multiplier, potential winnings, safe cards count
- **Controls**: Cash Out button (disabled until 2+ flips), New Game button
- **Sound Toggle**: Enable/disable audio effects

🌐 Backend Integration

API Overview

- **Base URL**: `https://to-be-shared/api`
- **Purpose**: Track game sessions and player statistics
- **Offline Mode**: App works without internet, syncs when available

Key Endpoints

1. **Start Game**: Send player info, stake, bomb rate → Get session ID
2. **Log Events**: Track each card flip, cashout, or bomb hit
3. **Get Stats**: Retrieve player history and analytics (optional)

What Gets Tracked

- **Game Start**: Player name, stake amount, bomb probability
- **Card Flips**: Which card flipped, current multiplier, balance
- **Game End**: Cashout amount or bomb hit, final balance
- **Session Data**: Duration, number of flips, outcome

Offline Capability

- Game works completely offline if server unavailable
- Events queued locally and synced when connection restored
- No gameplay interruption from network issues

🎵 Audio & Effects

Sound Effects Required

- **Safe Card Flip**: Pleasant "ding" sound when revealing safe card
- **Bomb Hit**: Explosion sound when hitting bomb
- **Perfect Game**: Celebration sound when flipping all safe cards
- **Sound Toggle**: Players can mute/unmute all sounds

Audio Files Needed

- `ding.mp3` - Safe card sound
- `explosion.mp3` - Bomb hit sound
- `hurray.mp3` - Perfect game celebration

Implementation Notes

- Use Flutter `audioplayers` package
- Sounds should be short (under 2 seconds)
- Provide mute option for users
- Handle audio permissions properly

💾 Core Data Structure

Game State Management

The app needs to track:

- **Player Info**: Name, wallet balance (starts at ₦10,000)
- **Current Game**: Stake amount, multiplier, safe cards flipped
- **Board State**: 25 cards with positions, bomb status, flip status
- **Game Status**: Active, game over, can cash out

Key Data Points

- **Wallet Balance**: Current money available
- **Current Stake**: Amount bet this round
- **Multiplier**: Current payout multiplier (starts 1.0x, +0.05x per safe card)
- **Safe Cards Count**: Number of safe cards flipped
- **Bomb Rate**: Calculated percentage for current stake
- **Game Active**: Whether game is in progress
- **Cards**: 25 numbered cards (1-25) with bomb/safe status

State Management

- Use Flutter Provider or similar for state management
- Update UI automatically when game state changes
- Persist wallet balance locally between sessions

🎮 Game Flow Logic

Game Sequence

1. **Setup**: Player enters name, selects stake → Calculate bomb rate → Generate board
2. **Gameplay**: Player taps cards → Check if bomb or safe → Update multiplier → Check win conditions
3. **End Game**: Cash out (win) or bomb hit (lose) → Update wallet → Return to setup

Key Functions Needed

- **Calculate Bomb Rate**: Based on stake amount (3-40%)
- **Generate Board**: Place bombs randomly based on calculated rate
- **Handle Card Flip**: Check bomb/safe, update multiplier, play sound
- **Check Cash Out**: Ensure minimum 2 flips, calculate winnings
- **Reset Game**: Clear board, reset multiplier, return to setup

Win/Loss Conditions

- **Win**: Player cashes out after 2+ safe cards → Receive stake × multiplier
- **Loss**: Player hits bomb → Lose entire stake amount
- **Perfect Game**: Flip all safe cards → Automatic cashout with celebration