

CCSL STAKEHOLDER REQUIREMENTS DOCUMENT

1. Functional Requirements

1.1 Game Setup

- Two-player game mode (Player vs Player or Player vs AI).
- Select match format: 1 Over, 2 Overs, 3 Overs, 5 Overs, 10 Overs, or Custom.
- Toss functionality to decide batting/bowling.

1.2 Card Play Mechanics

- Each player (batsman/bowler) selects a card (0–6) per turn.
- Power Cards (4 and 6) with cooldown logic and usage limits per over.
- Batsman has strategic tools: Strategic Pass, Hidden Card, Anticipation Bonus.
- Bowler has regulation tools: Defense Mode, Trap Mode, Power Blocker, Field Shift Bonus.

1.3 Special Rules & Conditions

- No Ball and Free Hit rules based on repeated values.
- Economy reward rule logic with forced 0 or 1 for next batsman move.
- Cooldown for power plays (e.g., no 6 allowed for 2 turns after playing one).
- Trap Mode and Power Blocker usages are limited per match size.

1.4 Turn Simulation

- Each over consists of 6 turns per player role.
- Real-time validation for rule conditions per ball.
- Live commentary/summary of each ball outcome.

1.5 Endgame Conditions

- Game ends after all overs are completed or wickets fall.
- Scoreboard display with runs, wickets, overs, extras.
- Result: Win, Lose, Draw, or Tie logic.



2. UI/UX Requirements

2.1 Dashboard / Home Screen

- Match Format Selector
- Start Game Button
- Access to Rules and About CCSL

2.2 In-Match UI

- Top Panel: Score, Over Count, Wicket Count
- Middle Panel: Card Picker UI (batsman/bowler)
- Power-Ups & Modes Panel (show cooldowns, usages left)
- Commentary Box (ball-by-ball events)

2.3 Card Selection Panel

- Batsman: Buttons for 0-6 + hidden card toggle

- Bowler: Buttons for 0–6 + trap card toggle
- Visual cue for activated power modes (e.g., defense, power blocker)

2.4 Result Screen

- Summary of score, overs, wickets
- Player of the match based on runs/wickets
- Button to replay or return to home

2.5 Visuals

- Use card-style animations and interactive feedback
- Each special mode should have distinct color/animation
- Dark mode and light mode switch

3. Technical Requirements

3.1 Platform

- Developed in Flutter (Dart), supports mobile & web
- Backend optional for multiplayer stats (can start locally)

3.2 Game Logic

- Use clear class-based model: Batsman, Bowler, MatchEngine
- Card actions validated before result
- Track stats: power cards used, traps, bonuses triggered

3.3 Storage

- Local storage (shared_preferences or Hive) to store match history and usage counters
- Optional: Firebase for real-time match logs or multiplayer expansion

3.4 Performance

- Lightweight animations
- All logic optimized to avoid frame drops

4. Optional Enhancements (Stretch Goals)

- Multiplayer (LAN or Firebase)
 - AI opponent (Easy, Medium, Hard)
 - Achievement Badges (e.g., Used All Power Cards, Clean Sweep)
 - Leaderboard or Tournament Mode
 - Sound Effects / Voice Commentary
-