Software Requirement Specification (SRS) Document

# Project Title:

Hand Cricket Game (Console-Based in C) JACKFRUIT PROBLEM SOLVING

# Team No:

10

# Team Members:

YASH RAGHAVENDRA A (PES2UG24AM222)

YASHWANTHA R(PES2UG24CS614)

VISISHT KOUSHIK(PES2UG24EC129)

# 1. Brief Description of Project

The Hand Cricket Game is a console-based multiplayer simulation developed in C. It recreates the traditional hand cricket game where two-player teams compete in a turn-based match. The game includes a toss, bat/bowl decision, over-based innings play, ball-by-ball scoring, and a result system. It uses a custom header file rules.h to define key game parameters, enhancing flexibility and modularity.

# 2. Purpose / Goal

The purpose of this project is to implement a functional hand cricket game using structured C programming. It aims to provide an interactive console experience while showcasing proficiency in logic design, modular programming, and file handling.

# 3. Usefulness / Benefit

• For Players: Engaging digital version of hand cricket with player statistics and match outcome.  
• For Developers: Modular codebase with structure-based data handling and custom headers.  
• For Reviewers: Demonstrates core C programming skills with real-world application logic.

# 4. Hardware / Software Involved

Hardware Requirements:  
• Standard PC/Laptop  
• Minimum 2GB RAM  
• 500MB Disk Space  
  
Software Requirements:  
• Language: C  
• Compiler: GCC / Code::Blocks / Turbo C  
• Operating System: Windows/Linux/MacOS  
• User-defined Header: rules.h  
  
rules.h Contains:  
#define OVERS 5  
#define BALLS\_PER\_OVER 6  
#define TOTAL\_BALLS (OVERS \* BALLS\_PER\_OVER)  
#define PLAYERS\_PER\_TEAM 2

# 5. Detailed Feature List

Player Input Module:  
• Input player names for both teams  
• Interactive input during game: hidden batting input and visible bowling input  
• Validation of inputs (1–6)  
  
Match Engine:  
• Toss simulation (Heads or Tails)  
• Bat/Bowl selection  
• Two innings of play (5 overs each)  
• Alternate bowler each over  
• Early termination if target is reached  
  
Statistics Module:  
• Track individual runs, balls, wickets  
• Calculate strike rates  
• Record input frequency for each player  
  
Result and File Storage:  
• Display match outcome (win/loss/tie)  
• Write match summary to file hand\_cricket\_score.txt, including:  
 - Team scores  
 - Individual stats  
 - Bowler stats  
 - Input frequency  
 - Winner

# 6. Test / Demonstration Plan

Unit Testing:  
• Test functions: toss, getHiddenInput, getVisibleInput, score logic  
  
Integration Testing:  
• Toss → choice → innings → stats → file write  
  
System Testing:  
• Full match simulation for all paths (Team 1 wins, Team 2 wins, Tie)  
  
User Acceptance Testing:  
• Test by different users for usability, clarity, and performance

# 7. Expected Interaction Interface and Sample Use Cases

Interface:  
• Console-based  
• Step-by-step prompts  
• Ball-by-ball feedback with clear display of runs/wickets  
  
Sample Use Cases:  
1. Start and Play a Match  
 - Input team members → Toss → Choose bat/bowl → Play 5-over innings → View result  
2. Simulate a Ball  
 - Batter enters a hidden number (1–6) → Bowler inputs visible number → Match logic determines outcome  
3. Save Match Summary  
 - After result → Match statistics saved to hand\_cricket\_score.txt → File can be opened later

# 8. Individual Member Contribution

• Student 1 YASH RAGHAVENDRA A: Developed match engine and logic control  
• Student 2 YASHWANTHA R: File handling and result module

• Student 3 VISISHT KOUSHIK: Player Input Module