**Cricket Score Sheet**

**Software Development Project Report**



**Supervisor**

**Dr. Md. Sazzad Hossain**

**Submitted by**

**Sagor Sarker**

**Department of Computer Science and Engineering,**

Mawlana Bhashani Science and Technology University, Santosh, Tangail-1902

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **INDEX** | **CONTENT** | **PAGE** |
| 1 | Analysis and Domain Modeling |  |
| 2 | Interaction Diagrams |  |
| 3 | Class Diagram and Interface Specification |  |
| 4 | Algorithms and Data Structures |  |
| 5 | User Interface Design and Implementation |  |
| 6 | Design of Tests |  |
| 7 | Project Management and Plan of Work |  |
| 8 | References |  |

1. **Analysis and Domain Modeling**
2. **Conceptual Model**
3. **Concept definitions**
4. **Association definitions**
5. **Attribute definitions**
6. **Traceability matrix**

Catch

Run Out

LBW

Stamping

BOWLED

WICKET

Wide

No run

User

Input

EXTRA

Legby

1-6

RUN

extra

By

NO BALL

Wicket

Run

By Run

1. **System Operation Contracts**

In this application when it is clicked it will be turned on and ask if the user wants to start a new game. If the user wants to start then a welcome window will be opened. After that, the user will see some instructions to run the application. Then some option of game details like venue, playing team, opponent, total over and toss winning team will be asked. Then the user will input every data of a cricket match.

1. **Data Model and Persistent Data Storage**

Yes, the system need to save data by using text file to store player details like name and skills.

1. **Mathematical Model**

Not applicable.

1. **Project Management**
2. **Interaction Diagrams**
3. **Class Diagram and Interface Specification**
4. **Class Diagram**
5. **Data Types and Operation Signatures**

In this class diagram menu is main class. There is some sub-class of this menu class. Like instructions, play, high score, level, settings and quit. Level has some object types those are easy, medium and hard. There will be other object types like sound on or off setting as well.

1. **Traceability Matrix**

Not applicable.

1. **Algorithms and Data Structures**
2. **Algorithms**

Algorithm for this application:

Step-1: start

Step-2: initialize sdl2

Step-3: variable declaration

Step-4: reading high score from text

Step-5: getting images

Step-6: getting name from user

Step-7: display welcome image

Step-8: main menu

Step-9: if quit then go to step 15

Step-10: sub menu

Step-11: creating obstacles

Step-12: play the game

Step-13: displaying score

Step-14: go to step 8

Step-15: save current high score

Step-16: end.

1. **Data Structures**

Yes, in this system we use array. We used array to store structural data and

collection of primitive data types such as integer.

1. **Concurrency**

Not applicable.

1. **User Interface Design and Implementation**

We designed the interface manually in c language. We also made some motion effects manually, like for every boundary 4 or 6 will move left to right of the window. It will also work for fifty or hundred scored by a batsman.

1. **Design of Tests**
2. **Test Cases**
3. Scoreboard
4. Game Details
5. Player Details
6. Instruction
7. Data of Batsman
8. Data of Bowler
9. Update Batsman
10. Update Bowler
11. Update Run
12. Update Over
13. Update Wicket
14. Update Scorecard
15. Update batting team
16. Update bowling team
17. Calculate final result
18. Declare winning team
19. **Coverage**
20. **Integration Testing strategy**

Not applicable.

1. **Project Management and Plan of Work**
2. **Merging the Contributions from Individual Team Members**

|  |  |  |
| --- | --- | --- |
| List | CONTRIBUTIONS BY  Mostofa Meem | CONTRIBUTIONS BY  Ruhan Sabab |
| Requirements Specification | 50% | 50% |
| Software design | 37% | 63% |
| Coding | 77% | 23% |
| Debugging and Testing | 50% | 50% |
| Report Preparation | 50% | 50% |

1. **Project Coordination and Progress Report**
2. **Plan of Work**
3. **Breakdown of Responsibilities**
4. **References**

-----------------------------------

Signature of supervisor Date