****

**Bowling Game application**

**Software Requirement Specification (SRS) Document**

**Sprint 1 Implementation**

**Project Timeline: 24.08.2022 to 30.08.2022**

**INDEX**

1. Introduction

1.1 Purpose -------------------------------------------------- 4

1.2 Intended audience -------------------------------------------------- 4

1.3 Intended use -------------------------------------------------- 4

1.4 Scope -------------------------------------------------- 4

2. Overall description -------------------------------------------------- 5

2.1 Assumptions and dependency -------------------------------------------------- 5

3. System feature and requirements -------------------------------------------------- 5

3.1 Functionality -------------------------------------------------- 5

3.1.1 Frame Number function —----------------------------------------------- 5

3.1.2 ScoreSoFar function -------------------------------------------------- 5

3.1.3 GameIsOver function -------------------------------------------------- 5

3.1.4 Roll function -------------------------------------------------- 5

3.1.5 Bowler database -------------------------------------------------- 6

3.1.6 Bowling database -------------------------------------------------- 6

3.1.7 Bowler\_data\_sheet function -------------------------------------------------- 6

3.1.8 Bowler\_day\_report function -------------------------------------------------- 6

3.1.9 linkedlist\_to\_file function -------------------------------------------------- 6

3.1.10 file\_to\_linkedlist function -------------------------------------------------- 6

3.2 System requirement -------------------------------------------------- 6

3.2.1 Tools to be used -------------------------------------------------- 6

3.3 System feature ------------------------------------------------ 6

4. DataFlow Diagram

4.1 DFD level 0 ------------------------------------------------ 8

4.2 DFD level 1 ------------------------------------------------ 9

### **1.** **Introduction: -**

The introduction of the software requirement specification provides an overview of the entire software. The entire SRS with overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyze and give an in-depth insight into the complete Automatic Bowling Scorer application by defining the problem statement in detail. The detailed requirements of the Automatic Bowling Scorer application is provided in this document.

**1.1** **Purpose**: **-**The purpose of this document is to show the requirements for the Automatic Bowling Scorer application, in which bowler scores will be kept and after each ball, their scores will be updated in the records.

**1.2** **Intended Audience: -**This document is intended to be read by, Client.

**1.3** **Intended Use: -**

* Development Team
* Maintenance Team
* Clients

Since this a general-Purpose Software any one can access it.

**1.4** **Scope: -**This project aims to create the development of an Automatic Bowling score application. Which takes the bowler information such as ID and Name, adds it to the database and returns the Bowling data of the user as to how many tournaments won along with also displaying the bowling game score of the user as to which frame the user is currently bowling in and what is the total score after each frame.

**2. Overall Description: -**

It is an automatic bowling scorer application used to keep the track of scores scored by a bowler. This consists of the database which will store the information of the bowler. It will give the information about which frame and bowl is about to be next. Continuously shows the score of the bowler so far after each bowl. It will create the reports of the player on demand like Bowler Data Sheet which will on entering the bowler id his details should be displayed which includes the data available in the bowler database and his bowling data from the current tournament. Bowling Day Report which will contain all details of the entire bowling tournament. The main purpose of this project is to automate the process of keeping track of the scores and data of the player.

**2.1 Assumptions and Dependency: -**

* System should have Ubuntu Linux installed.
* · System should have either 4GB or more RAM.
* The service is used preferably on a desktop or laptop.

**3.System Features and Requirements: -**

**3.1 Functionality: -**

**3.1.1 BG\_01-> Maintain\_database:** This is the first main menu level function that provides flexibility to enable the user to do modifications inside the database. It allows you to Add, Edit, Delete and View contents or records of the database.

**3.1.2 BG\_02-> Add\_bowler**: This function is the start point of the application as the user creates a bowler profile of the player by adding the ID and other details about the player in the record.

**3.1.3 BG\_03-> Edit\_Information**: This function is used to edit/update any kind of changes into the existing records in the database itself. After entering the ID the user can edit other related information about the player.

**3.1.4 BG\_04->** **Delete\_bowler:**  As the name suggests, the job of this function is to delete a certain record from the database.

**3.1.5 BG\_05-> View\_bowler\_information:** This function basically returns the values existing in the bowler database of a particular player after its ID is entered.

**3.1.6 BG\_06-> Play\_the\_game:** This is a main menu level function that further calls for the play function which in turn does the scoring of the game.

**3.1.6 BG\_06-> Random\_number:** This function’s task is to generate a random number every time it is called for in the range of 1 to 10. Used for generating scores for the player for each ball of the frame.

**3.1.6 BG\_06-> Strike condition:** In compliance with the scoring system rules, it is important to identify each strike condition, i.e if if all the pins are bowled in the first ball itself. This function checks for all such conditions and accordingly processes scores.

**3.1.6 BG\_06-> Spare\_condition:** In compliance with the scoring system rules, it becomes important to find all the spare conditions, i.e if if all the pins are bowled in the first and second ball of the frame itself. This function checks for all such conditions and accordingly processes scores.

**3.1.6 BG\_06-> Extra\_balls:** Since we need to add next frame scores if there is a strike or spare, in case of a strike or spare in the tenth frame there won't be any other frames. So to tackle this, the extra\_balls func checks for this condition and accordingly gives the extra balls without adding a frame.

**3.1.6 BG\_06-> Make\_current\_tournament\_zero:** The application is designed to store many details including details of previous tournaments. So to make sure the points for the new tournament aren't being compared with the previous tournament scores we use this function. This function initializes the current tournament to 0 for each player, when when a player plays a tournament it is incremented to 1 differentiating it from the rest of the players.

**3.1.7 BG\_07-> Play\_the\_game\_functions:** Is the main function that is responsible for scores generated for the user while playing the game. It returns many values that are added to the bowling database such as the number of strikes, number of spares, total score of the game etc.

**3.1.8 BG\_08-> Show\_report:** This function is used to view the generated reports of the player after the match. It can be used to view the bowler details in general which describes the overall performance of the player or can be used to view the bowling details of all the players.

**3.1.9 BG\_09-> Bowler\_data\_sheet:** This function will take the input from the user, the bowler id, the function will display which includes the data available in the bowler database and his bowling data from the current tournament which is stored in the bowling database.

**3.1.10 BG\_10-> Bowler\_day\_report function:** This function will display all the records of all the bowlers stored in the bowling database.

**3.1.11 BG\_11-> List\_to\_Bowler:** This function will copy all the data from the linked list and store it permanently on a file. This function will be executed when the game gets over. This function will copy the contents of linked lists and store them in the bowler database. The function writes each node of the linked list to the file in serial order.

**3.1.12 BG\_12-> Bowler\_to \_list:** The data in the file will be converted or pushed into the linked list. This will be done at the starting of the program so the data can be used in the program. This function will connect and return the bowling and bowler database records to the main function

**3.1.11 BG\_13-> List\_to\_Bowling:** This function will copy all the data from the linked list and store it permanently on a file. This function will be executed when the game gets over. This function will copy the contents of linked lists and store them in the bowling database. The function writes each node of the linked list to the file in serial order.

**3.1.11 BG\_14-> Bowling\_to\_List:** The data in the file will be converted or pushed into the linked list. This will be done at the starting of the program so the data can be used in the program. This function will connect and return the bowling and bowler database records to the main function

**3.2 System Requirements: -**

### **3.2.1. Tools to be used:**

* Pthread Library
* C File Handling
* C Language
* System Programming

### **3.3 System Features: -**

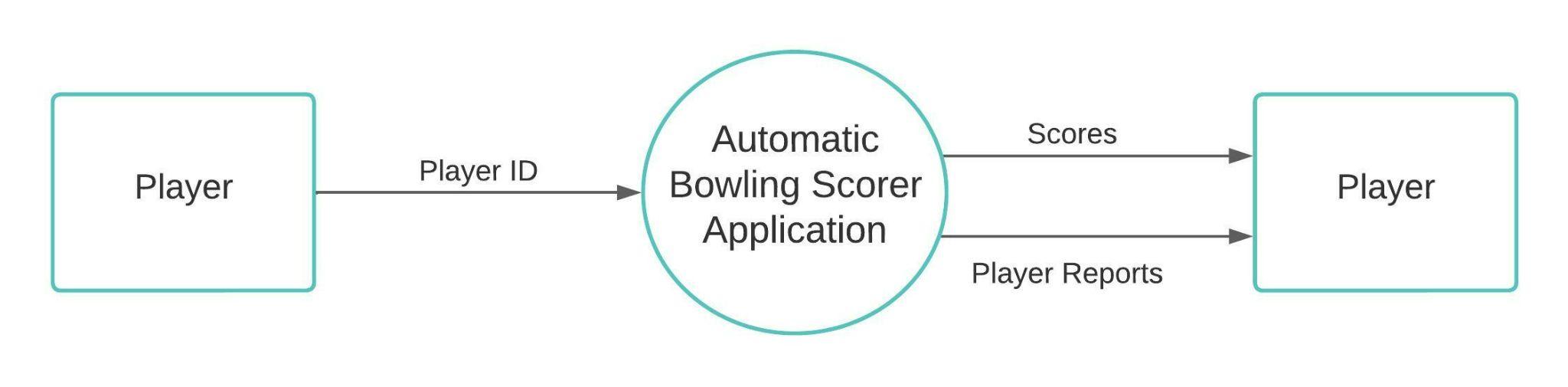
### Supportability:The system is easy to use.

* Design Constraints: The system is built using only C language.
* Usability:The automatic bowling scorer application can be used to replace the old means. Recording score by displaying various information to the player such as number of pins knocked down by each ball, the frame number which is currently going on, the cumulative score gained throughout the the end of each frame and the total number of points scored by the player after the completion of a total of ten frames.
* Reliability & Availability**:** The system is available 24/7 that is whenever the user would like to use the system, they can use it up to its functionalities.

### Performance: The system will work on the user’s terminal**.**

**4. DataFlow Diagram:**

**4.1 DFD Level 0 -**



**4.1 DFD Level 1 -**

