### Example Project - Analysis

### Contents Table

[Analysis 2](#_Toc459748180)

[Background to and Identification of the Problem 2](#_Toc459748181)

[The Current System 2](#_Toc459748182)

[Specific Requirements of the User and Acceptable Limitations 4](#_Toc459748183)

[Analysis Data Dictionary 6](#_Toc459748184)

[E-R Model For Existing System 6](#_Toc459748185)

[Data Volumes 6](#_Toc459748186)

[Data Flow Diagrams For The Existing System 7](#_Toc459748187)

[Numbered Objectives Of The Project (This is the most important section) 8](#_Toc459748188)

[Proposed Method of Solution 9](#_Toc459748189)

# 

# Analysis

## Background to and Identification of the Problem

I play in a ten pin bowling league on Tuesday nights. There are currently 18 teams in the league but there can be as many as 24 teams and as few as 6. Each team can have up to 7 players, only three of whom play on any night.

Matches consist of three games. Each player plays against their 'opposite number'.

Each player is awarded a handicap based on his or her average score (called pin-fall) since the start of the season. A player wins a game if their pin-fall plus handicap is more that the opposition player pin-fall plus handicap.

For each game, the team gets one point for every game won by a player, and four points if the team's combined score (including handicap) beats the opposition's combined score.

The teams also get one point for each player whose total score over the three games beats the opposition player's total and four points if the team's overall total beats the opposition's overall total.

Each match, therefore, is for a total of 28 points. Half points are awarded for drawn games.

The league secretary, Fred Bloggs, collects scores at the end of the night and uses them to produce an updated league table and a list of individual players' handicaps for the following Tuesday. He pins the league table and list of handicaps to the notice board at the start of the evening. He currently uses a spreadsheet to record the details and produce the league table.

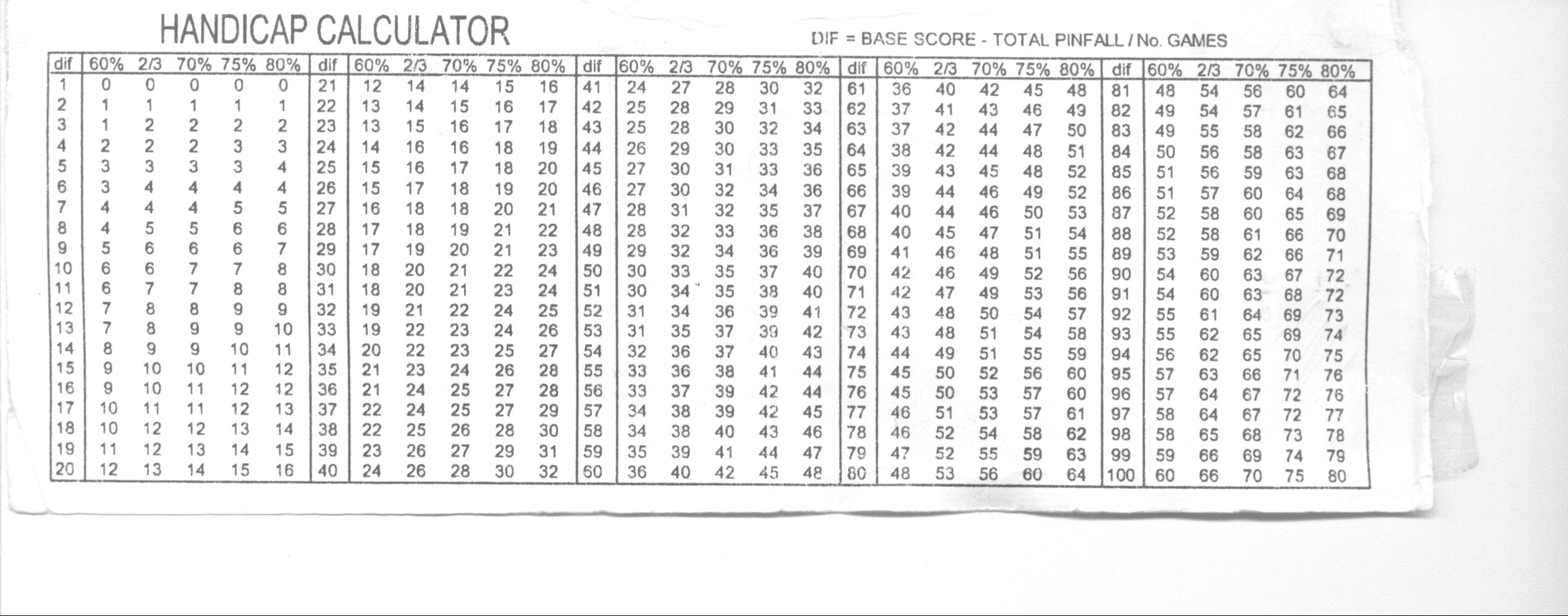
I talked to Fred to find more details about the current system and to find out what he would like the new system to do. I have summarised my findings below.

## The Current System

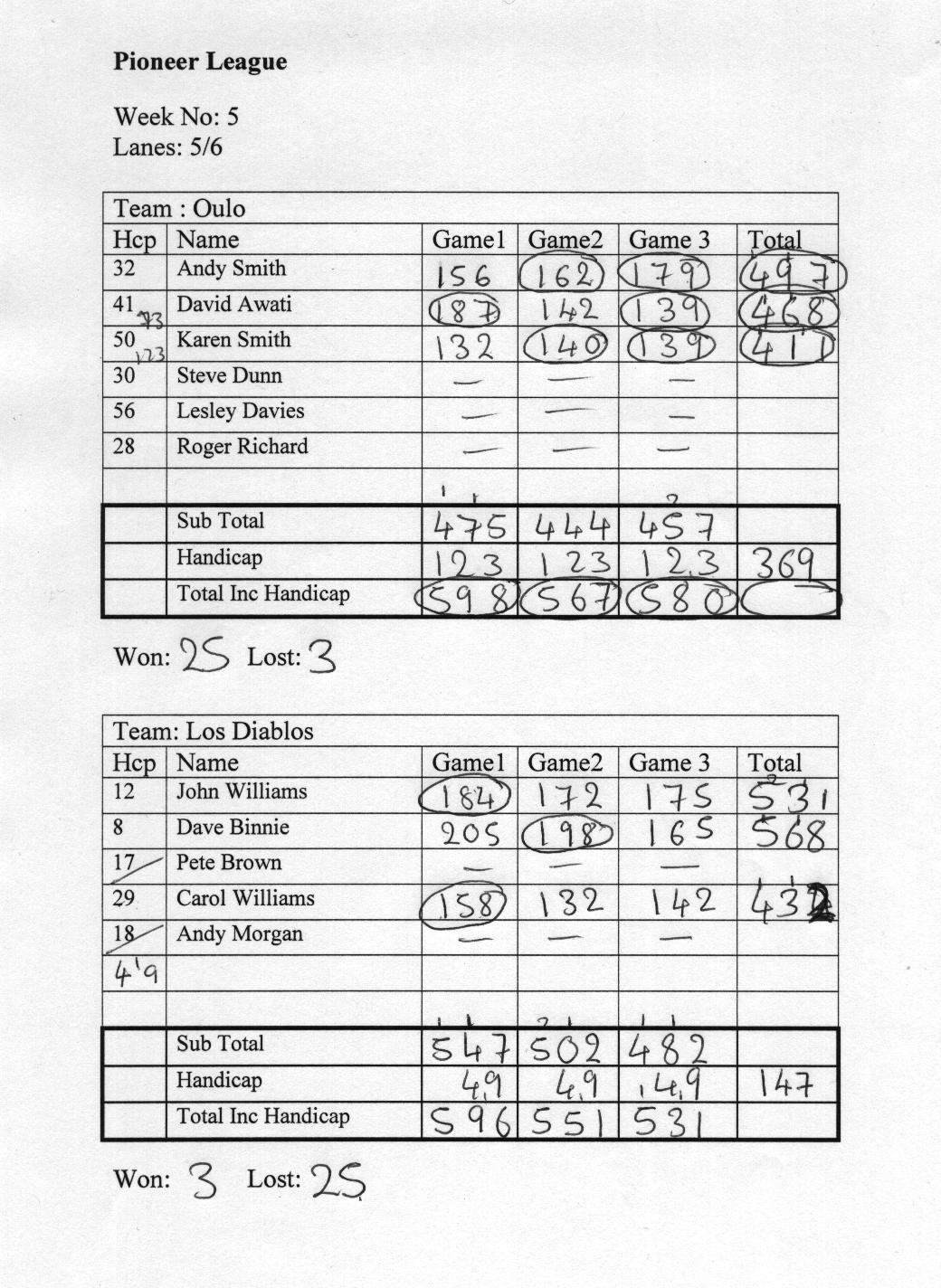
Before the start of each season a reformation meeting is held. During this meeting team captains fill in a form to register the team for the new season. They need to fill in the team name, the names of the players in the team and a contact number and/or e-mail address for themselves. Additional players can be added during the season up to a maximum of 9 in any team. However players cannot be transferred between teams during the season.

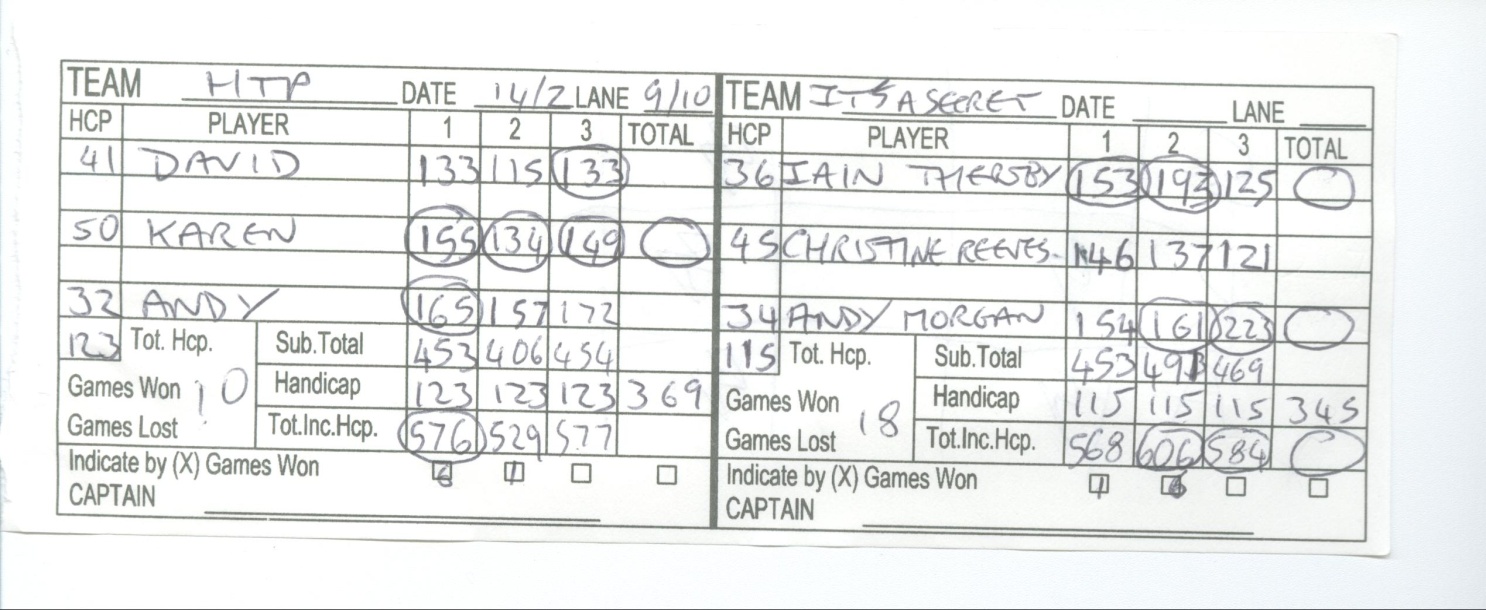
The league secretary keeps the contact details for each team in a folder and adds any new e-mail addresses to his address book in the e-mail program he uses. The names of the players in each team are typed into the spreadsheet program he uses for calculating scores and league positions. He then produces a fixture list (see page 5) for the season which is distributed on the first night of the season. If there are an odd number of teams, then one team will have a bye each week. There are a maximum of 44 matches during the year. This allows weeks off for the summer holidays, Christmas and Easter. If there are fewer than 23 teams it may be decided that each teams plays each other team twice.

At the start of each season, each team is issued with a score book.

A player's handicap is calculated by averaging their pin-fall since the start of the season, subtracting the average from 200, then calculating two thirds of this number. There is a grid in score book (see below) to help with this.

For the first time a player plays in a season, the handicap is worked out on the match just played. For future matches it is worked out using the previous weeks' scores.

Before the start of each match, Fred hands out a team sheet for each pair lanes which has on two grids for filling in the scores as shown below.

During each match, each team fills out the team sheets and may also fill in one of their score book to keep their own record of the match. See below.

After the match Fred collects in the team score sheets. He also gets a computer printout of the scores from the Bowling centre. He uses these sheets to enter every player's score into the appropriate week's worksheet on the spreadsheet; each week has a separate worksheet.

If a team is missing a player, the team get a 'dummy player', called a blind, who is awarded a fixed score of 150 for each game.

If a player is playing for the first time, Fred has to calculate his or her handicap based on this week's scores, otherwise the spreadsheet calculates the handicap based on the average on the scores from the previous weeks.

The spreadsheet calculates the points awarded to each team and adds them to the totals for the season. The totals are stored in a table on a summary worksheet. The table is then sorted into points order to produce a league table.

Fred copies and pastes the league table into a word-processor document. He also checks through the spreadsheet to find the players with the highest game and match total, with and without handicap, for the season so far. He also looks for the teams with the highest game and match total with and without handicap. These are included in the word-processed document along with every player's new handicap.

The document is printed and pinned onto the notice board ready for the next match night.

Sometimes matches have to be postponed when one team cannot find enough players. When this happens, the next time that both teams have a match, their scores count for the match they are playing and the postponed match. This is known as a double-header. The postponed match is deemed to have been played first and the players' handicaps are recalculated before working out the scores for the second match.

Fred has said that the spreadsheet is good for working out the handicaps and scores but it is time-consuming to enter the details and to produce all the required reports. He would really like a tailor-made solution.

## Specific Requirements of the User and Acceptable Limitations

Fred has said that he would like a system that makes data entry easier. If possible he would like the data entry form for the scores to resemble the score sheets.

He would like a way of transferring the players' names from one season to the next so that he doesn't have to copy and paste them or type them in again. He would also like the system to store contact details for the captains so that he doesn't need to keep a separate folder.

The system needs to be able to do all the processes that the spreadsheet does including:

* calculating all the handicaps including when a player plays for the first time,
* calculating each players total score for the match,
* calculating the teams total score for each game and for the match,
* calculating the points awarded to each team and producing a league table.

It also needs to automatically find the players and teams with the highest scores.

He would like the system to print out the league table, details of players' handicaps and highest scores. Ideally he would like some way of incorporating these printouts into one document to which he can add news items but if it can't be done he is happy for the printouts to be separate pages.

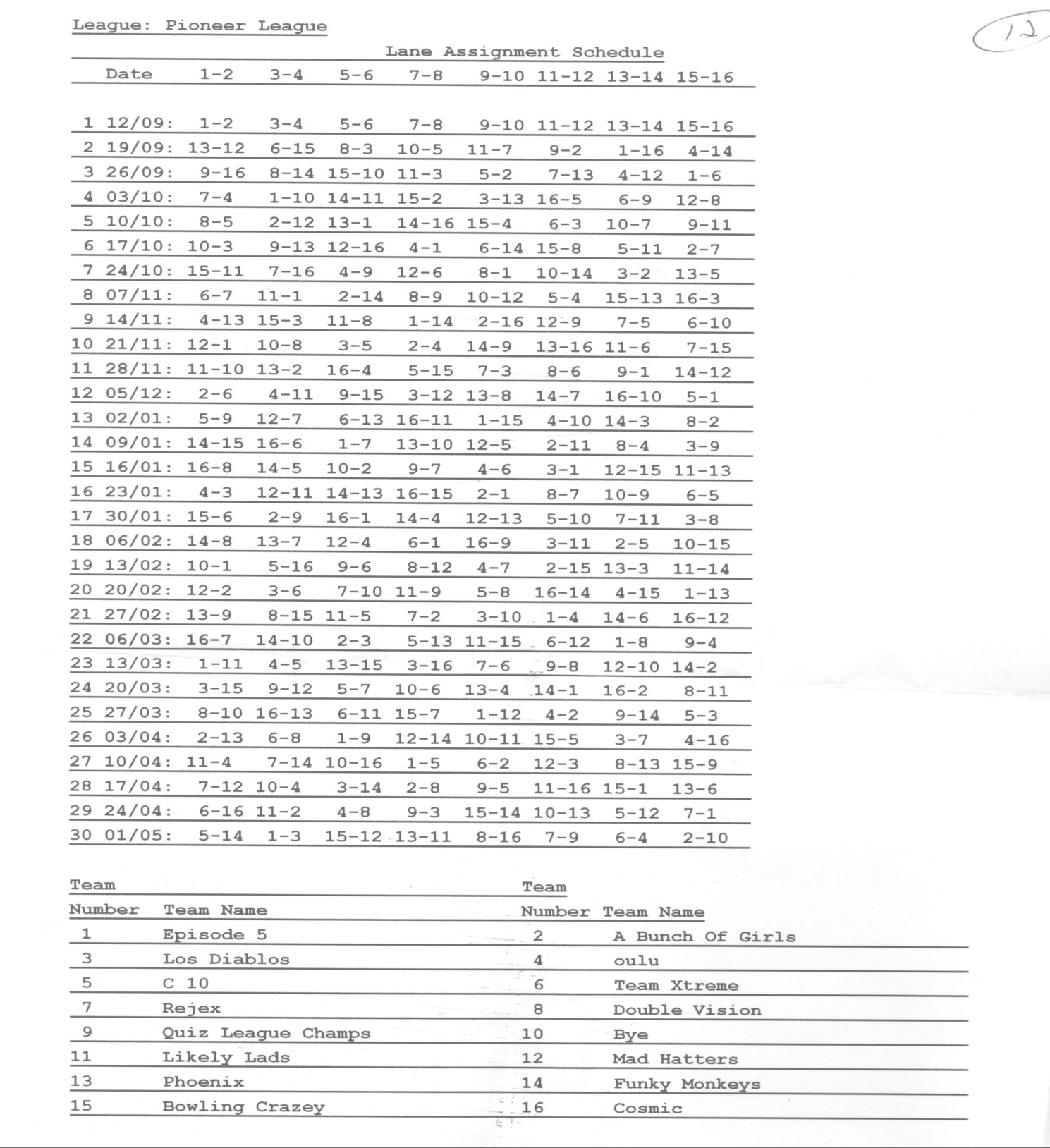
Fred has also said that if time allows he would really like the system to be able to work out the fixtures at the start of the season so that he does not have to. However, he knows that this is quite difficult to do so he will be content if the system just allows fixtures to be entered.

One problem with the new system will be the transferring of all the data about teams and players currently held on the spreadsheet. Someone will have to spend time typing the data in unless a way can be found to transfer it.

Fred doesn't want to buy a new computer so the system has to work on a PC with Windows 7 operating system. The spreadsheet he uses at the moment is MS Excel. He has the MS Office suite which also includes Access and Word. He is familiar with using Word but has never used Access.

The program will be installed on his home PC and so a log in system is not needed.

### Example of a Fixture List

****

Analysis Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Validation** | **Example Data** | **Comment** |
| TeamName | Text | Presence check | Brownian Motion | Each team has a different name |
| Captain | Text | Presence check | Fred Bloggs | Each team must have a captain who is the main contact person. |
| Contact | Text | - | 01612231781 | Contact telephone number |
| EMail | Text | - | Fred@hotmail.com | Contact e-mail address |
| PlayerName | Text | - | Fred Bloggs | Player's name |
| MatchDate | Date | - | 02/05/2007 | Date of match |
| Week number | Number | - | 5 | The week in the season. |
| GamesPlayed | Number | Calculated | 9 | The number of games played so far. |
| Score | Number | >0 | 150 | Player's scratch score in a particular game. |
| Handicap | Number | Calculated | 30 | Player's handicap |
| PointsFor | Number | Calculated | 14 | Total points awarded to the team so far. |
| PointsAgainst | Number | Calculated | 12 | Total points awarded against the team so far. |
| BestGame | Number | Calculated | 230 | Highest scratch game |
| BestSeries | Number | Calculated | 680 | Highest scratch series |
| BestHcpGame | Number | Calculated | 260 | Highest game including handicap |
| BestHcpSeries | Number | Calculated | 700 | Highest series including handicap |
| TeamGame | Number | Calculated | 680 | Highest team game including handicap |
| TeamSeries | Number | Calculated | 1800 | Highest team series including handicap |

## E-R Model For Existing System

have

has

Team

Players

Match

Results

**Data Volumes**

The system will not have to store vast amounts of data. There can be upto 24 teams in the league as that is the number of lanes in the bowling centre. The system will need to be able to cope with storing details of the 24 teams.

Each team can have upto 7 players registered making a maximum of 168 players in total. The amount of data stored about each player is quite small as individual results do not need to be stored.

Each season lasts a maximum of 44 weeks so the maximum number of fixtures that need to be recorded will be 528.

The team and player data will mainly need updating at the start of each season, although some updates may be needed during the season if, for example, a player joins a team.

The scores and match details will need updating every week after the matches have been played.

## Data Flow Diagrams For The Existing System

It may be appropriate to include a diagram or diagrams to show how the key processes in the current system link together. I have included a couple of DFDs to show this but you can use any type of diagram or even a list of steps.

Start of Season

After Each Match

Team Captains

Enter Players into spreadsheet

Fred

Team Sheets

Fill in Team sheets

Team Captains

League spreadsheet

Produce fixture list

Fred

TeamName, contact details, players' details

Fixture List

TeamName, contact details, players' details

Team names, players' details

Team names, players' details

Team names

League spreadsheet

League spreadsheet

Handicaps and league table

Process scores

& calculate results

Computer

Team names, players' scores

Team names, players' scores

Team Captains

Enter Players into spreadsheet

Fred

Fill in Score sheets

Team Captains

Team Names, players' scores

Team Names, players' scores

Team names, players' scores

Bowling Alley Staff

Team Names, players' scores

Results and league positions

Print out handicaps and league table

Computer

Handicaps, results and league positions

Team Captains

Score Sheets

## Numbered Objectives Of The Project (This is the most important section)

Functionality Objectives

1. The system must create the database when the program is run the first time.
   1. The program must identify whether or not the database exists
   2. If the database doesn't exist the database and all the required tables need to be created.
2. At the start of the season the system must allow details about new teams to be entered and stored.
   1. It must allow the team name, the name of the captain and contact details to be entered and stored.
   2. It must allow the names of all the players (upto a maximumof 9) in the team to be entered and stored.
3. At the start of the season the system must allow the name of a team to be changed.
   1. It must allow the new team name to be entered and stored.
   2. It should identify players currently in the team and is should be possible to transfer them to the new team if required.
4. At the start of the season the system must allow old teams to be deleted.
5. The system must allow players to be transferred to new teams at the start of the season.
6. It must allow new players to be added during the season.
   1. It should be possible to enter the player's name and select which team they are to play for.
   2. It must not allow the number of players in a team to exceed 9.
7. It must allow the fixture list to be entered.
8. It must automatically 'know' which week it is and allow the scores of all the matches to be added.
   1. It should display the details (team names and all the players) for each match.
   2. It should allow the players who played in the particular match to be selected.
   3. It must enable each player's scores to be entered after each match.
   4. It must deal with 'dummy players' when a team fields fewer than three players.
   5. It must calculate the match results from the scores entered.
   6. It must store the results of each match.
   7. It must identify when a match is a bye and realise that scores are not needed for that match.
9. It must update the league table after all the match results have been entered.
10. It needs to calculate and store each player's handicap including when it’s the player's first match.
11. It needs to keep track of each players' highest handicap and scratch game and series scores.
12. It must allow team lists to be printed including a list of players with their handicaps.
13. It must print out an updated league table showing the team name, matches played, points awarded, total pin-fall for, total pin-fall against, best game and best series for each team.
14. It must identify and print out details of the team with the highest game and highest series so far.
15. It must identify and print out details of the players with the highest game with handicap, the highest series with handicap, the highest scratch game and the highest scratch series.
16. Data entry should be kept to a minimum by using combo boxes and grids/tables so that it is possible to simply click on the required data.
17. The system should allow players to be transferred to a new team for the new season without having to type all the names again.
18. The data entry screen for entering match scores should resemble the score sheets to make data entry easier.

**Extension Objectives**

1. Create the fixtures automatically at the start of the season.
   1. It should allow the number of teams playing in the league to be entered.
   2. It should calculate how many weeks are to be played.
   3. It should produce a fixtures list stating the week number and which team is playing which team and on which lanes.
   4. Each team must only have one fixture per week.
   5. Details of the fixtures must be stored.
   6. It should be possible to print the fixtures list.
2. Deal with postponements and double-headers for when a team can't play a match on the required night.
   1. The system must allow a match to be identified as being postponed so that no scores are entered for the match initially.
   2. The system should identify when the teams which postponed a match next play.
   3. It must allow the scores for the postponed match to be entered.
   4. It must recalculate the handicaps of all the players using the results of the match.
   5. It should automatically calculate and display the scores for the current week's match.

## Proposed Method of Solution

I have decided to solve the problem using Delphi with Access. I will store the data in Access tables and write the program to manipulate the data using Delphi. The tables will be created when the program is first run by using DDL within Delphi.

Writing the program in Delphi means that I have a lot of control over the way the user interface will look as there are lots of visual components available in Delphi.

Storing the data in Access tables rather than files means that I can use some of the built-in routines for manipulating data in tables. For example there is an in-built sort routine. Also if the user decides to add a new field to a table or to change the data structure in anyway, it will not require major alterations to the program as it would if I used files of records.