H447   Unit F454: Computing Project

F454: Project
# Contents

A2 Project .................................................................................................................. 1

Section 1 - Definition, Investigation and Analysis ..................................................... 4

Introduction ................................................................................................................. 4

Initial Interview .......................................................................................................... 4

What I Found out ......................................................................................................... 5

Background to the Organisation ................................................................................... 5

Initial Problem Definition ........................................................................................... 6

Fact Finding ................................................................................................................. 7

Second Interview ........................................................................................................ 7

Current Methods/Documentation Used ........................................................................ 9

Interview ................................................................................................................... 13

This was the interview with Mr Keen ......................................................................... 14

Data Flow of Current System ..................................................................................... 15

Match Data ................................................................................................................ 15

Formatted Player Data ............................................................................................... 15

Formatted Match Data ............................................................................................... 15

Statistical Data ........................................................................................................... 16

Results data processed .............................................................................................. 15

Origins and Forms of data .......................................................................................... 16

Fact Recording Justification ........................................................................................ 16

Restatement of Problem ............................................................................................ 16

Requirement Specification .......................................................................................... 17

Overview ................................................................................................................... 17

Output requirements ................................................................................................... 17

Input requirements ...................................................................................................... 17

Processing requirements ........................................................................................... 17

Estimation of file sizes ............................................................................................... 18

Alternative Approaches ............................................................................................. 19

Selected Approach .................................................................................................... 20

Hardware and Software Requirements ......................................................................... 20

Section 2 - Design ...................................................................................................... 21

Design Objectives ....................................................................................................... 21

Design Specification ................................................................................................... 21

Input ............................................................................................................................ 21

Input, Processing & Output ......................................................................................... 23

User Interface ............................................................................................................. 24

Files and Data Structures ......................................................................................... 31

Data Structure ........................................................................................................... 35

Error Messages ........................................................................................................... 36

Message Reason ......................................................................................................... 36

Error ............................................................................................................................ 36

File Sizing ................................................................................................................... 37

Intended Benefits ....................................................................................................... 37

Limitations of the System .......................................................................................... 38

Section 3 - Software Development, Testing and Implementation ............................. 39

Test Plan ..................................................................................................................... 40

Test Detail ................................................................................................................... 41

User Testing ............................................................................................................... 64

User Questionnaire ................................................................................................... 65

New Test Plan ............................................................................................................ 70

Test Detail ................................................................................................................... 70

Ongoing User Testing ............................................................................................... 74
Section 1 – Definition, Investigation and Analysis

Introduction

My name is ___________________________ and I am doing Computing and this is my A2 computing project based on developing a Computerised Scoring System. My users are 2 members of the new staff cricket team, I ___________________________.

Initial Interview

To begin I will interview both users to firstly get a brief description about the staff cricket team, including details such as when they play and what part the users play in the team. I will then ask about how the current system works with an aim to identify the problems with the current system and possibly take note of any requested suggestions for the new system.

These are the questions I plan to ask:

About the team
  - How often does the staff cricket team play?
  - Who captains the team?
  - What roles do you play in the team?

Problem definition
  - How is data collected, used and stored?
  - What data is collected?
  - What problems do you come across with the current system?

Further questions may be asked to find out about the problems with the current system.
How often does the staff team play?
The staff team normally plays once a week on a Wednesday evening, for about 12 weeks during May through to July. Generally there are no other sessions (such as training).

Who captains the team?

Who captains the staff team.

What roles do you play in the team?

I play most games and then take the score card at the end of the game and put the results on the school website. I play if I'm needed, and usually do the score book whilst our team is batting.

How is the data collected, used and stored?
The data is collected by the scorekeeper from watching the game, it is then written into the scorebook, after the game has finished. I will take the data and type it up into a spreadsheet or sometimes just straight onto Firefly. The data is then stored on Firefly, in the score book as a hard copy and potentially on a spreadsheet.

What data is collected?
For Batting, the runs made, balls faced and number of times the batsman has been out over the course of the season is kept. From this the batsman's average (Total runs/Number of times out) and strike rate (Average number of runs that would be scored from facing 100 balls). For Bowling, Overs bowled, Middens (An over where no run is scored), Runs and Wickets taken. From this the bowler's economy (average number of runs per wicket) can be calculated.

What problems do you come across with the current system?
A common error is simply knowing which batsman is facing or which bowler is bowling, as from the boundary it is often difficult to identify which batsman is which or which bowler is bowling, especially when trying to identify a player from the opposing team.

Also as a scorer it is quite easy to forget to fill part of the scorebook after a ball, or make another mistake simply from not correctly seeing what happened in that ball.

Another problem is that if the relevant bowler or batsman cannot be identified then it is likely that the information that needs to be filled in for that player will be put in the wrong place, it then becomes a laborious to correct the mistake as the incorrect marks need to be rubbed out, then the correct marks put in place, this will usually happen whilst the game is still being played and can cause the scorer to lose track of what is currently happening.

When I type up the information from the score book it is a very time consuming, laborious and error-prone process. The excel spreadsheet used has not been setup to store the information. It is an unformatted spreadsheet with data on it. Sometimes it is simply easier just to upload the results straight onto Firefly rather than putting them into excel first, which makes the spreadsheet inaccurate for calculating statistics for the season.

Another problem is that different scorers may use different methods of marking down the score in the score book. For example a bowler may be written as having bowled 2.5 overs, I don't know if the scorer means 2 overs and 6 balls, or 2 and a half overs in which case it would be 2 overs and 3 balls.

What I found out

Background to the Organisation

The --- 3 staff cricket team is a group of members of staff that play a cricket match once a week on a Wednesday evening, they are captained by --- Member --- often ends up scoring the match using the score book and alter the match in --- match will put the score onto Firefly.
Initial Problem Definition

As a scorer in the game, there are problems identifying who is bowling and which batsman is facing as it is difficult to see from the boundary of the pitch who a particular person is. This error often means that the scorer will write down the score in the column of a batsman/bowler knowing that they may not be writing in the right column.

When they eventually identify who the batsman/bowler is it takes them a while to copy the score over to the correct column and then erase the score from where it was incorrectly placed. This can often mean the scorer loses track of the game and potentially makes another error.

As the scorebook is filled in manually there is always the potential for human error to occur whereby the correct markings are not written down due to misunderstanding or not knowing how to fill in a scorebook correctly.

When transferring the data from the scorebook to the computer, the process can be very laborious and also the data may not be consistent, some data e.g. number of balls faced by a batsman may not be recorded by all the scorers as different scorers may use the scorebook to score in different ways.

With the current system the method of transferring data onto the website isn’t consistent; sometimes the data is put into a spreadsheet, certain statistics are calculated then it is transferred to the site whilst at other times the data is put straight onto the site, which means the spreadsheet is inaccurate for statistics over the whole season.
Fact Finding

Second Interview
In the second interview I will be aiming to find out how the current scoring system works in as much detail as possible.

Preparation

I will ask to see a copy of the scorebook from a recent fixture.

- What are the different outcomes from when a ball is bowled?
- For each outcome... What is written down for this outcome?
- What are the limitations of this method of scoring?
- What would you want the new system to do?

I will ask to see the current master file and any other relevant documents.

- How do you normally transfer the data from the scorebook to the website?
- How reliable and accurate is the information that you upload?
- How often do you upload the data?
- How is the information presented?
- Does the information have any specific use other than for general viewing?
- What master files do you use?
- What would you want the new system to do?
Interview

- What are the different outcomes from when a ball is bowled?
- For each outcome... What is written down for this outcome?

For each outcome:

Dot Ball:
- A dot in the batsman’s column
- A dot in the bowler’s column

Runs Scored
- The number of runs scored is put in the bowler’s column
- The number of runs scored is put in the batsman’s column
- The amount of runs scored is added on to the score tally

Wide with no extra run
- A + is put in the bowler’s column
- The amount of runs gained from a wide is put in the wides column in the extras.
- The amount of runs gained from a wide is put added to the score tally.

Wide with an extra run
- A + is put in the bowlers column with the number of additional runs scored written in the top right corner of the +
- The amount of runs gained from a wide is put in the wides column in the extras, the additional runs are added to this.
- The total amount of runs gained from the ball is added to the score tally.
- An extra ball is normally given.

No Ball
- A O is put in the bowlers column
- The amount of runs gained from a no ball are written in the no ball column
- If extra runs are scored off the bat they are written in the batsman’s column.
- If extra runs are scored not off the bat, they are added to the no ball column.
- The total amount of runs from the ball is added to the score tally.

Byes/Leg Byes/Penalties
- A dot is put in the bowlers column
- The number of runs scored from the bye is added to the byes/penalties column.
- The number of runs scored is added to the score tally.

Wicket
- A W is put in the bowler’s column if it is the bowler’s wicket, a dot if it isn’t.
- The type of wicket is put in the How Out column.
- The bowlers name is put in the bowler’s column if it is the bowlers wicket, if it isn’t the column is left blank.
- If a fielder is involved in the wicket (i.e. caught) the name of the fielder is written down.
- If it is a run out then runs may be scored before the wicket occurs, these runs are added to the batsman’s column and the score tally.
- The Score at the fall of the wicket and the number of the batsman who is out is recorded in the fall of wicket information.

End of the Over
- The score at the end of the other, total runs scored and the number of the bowler who bowled the over is recorded in the over’s information.
What are the limitations of this method of scoring?

The main limitation is the ease at which human error is made, which means the system is not very reliable. The other limitations are the ability to score more detailed features such as the amount of time a batsman has been in for or the length of time of the game. Also all the information recorded in the scorebook has to be used manually to calculate any statistics and to do any comparisons.

What would you want the new system to do?

The new system would be able to simply run on a laptop at the pitch side, it should overall make scoring an easier and more accurate process.

At the start of the game the user will have a menu to input information about the game (e.g. limited overs, bowling restrictions, which teams are playing, which players are playing). The system will save information about both teams when it is entered. It will then allow the user to select the relevant team next time they are being played against.

At the start of every innings the user will be asked which team is batting, which 2 batsmen are opening (indicating which one is facing first) and which bowler is bowling. From this a database should be created for the RGS staff team, so that any subsequent games players can be chosen from the database. This database should also contain statistics about the players from the matches they have played which can be viewed at any time.

After every ball, a keystroke is pressed to indicate what has occurred that ball.

Any outcome which requires extra information (such as a no ball) will cause a menu to pop up with the relevant options for what could happen.

The system will provide the ability to change features such as players and scores, and the user will also be able to override the automatic features of the system (such as when an over finishes).

At the end of an innings or the end of the match a set of summary statistics will be produced.

Current Methods/Documentation Used

On the next page is an annotated copy of the scorebook from 1 innings of a match.
<table>
<thead>
<tr>
<th>Player</th>
<th>Overs</th>
<th>Runs</th>
<th>Wickets</th>
<th>Fall of Wickets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>3</td>
<td>10</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Mary</td>
<td>3</td>
<td>15</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Mike</td>
<td>2</td>
<td>12</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Peter</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Barry</td>
<td>1</td>
<td>10</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

**Bowling Analysis**

<table>
<thead>
<tr>
<th>Bowler</th>
<th>Wickets</th>
<th>Overs</th>
<th>Runs</th>
<th>Extras</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Black</td>
<td>2</td>
<td>15</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>Green</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Blue</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

**Summary**

- Total Runs: 55
- Total Wickets: 10
<table>
<thead>
<tr>
<th>Batsmen</th>
<th>Runs</th>
<th>Bowler</th>
<th>Ov</th>
<th>Wkts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>12</td>
<td>Baker</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Fox</td>
<td>27</td>
<td>Wiley</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Miller</td>
<td>17</td>
<td>Packer</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>N. Walsh</td>
<td>9</td>
<td>Duffett</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Davis</td>
<td>6</td>
<td>A. Cole</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>K. Walsh</td>
<td>5</td>
<td>Cock</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Hansen</td>
<td>12</td>
<td>Hull</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bowling Analysis</th>
<th>Club</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baker</td>
<td>4.0</td>
</tr>
<tr>
<td>Wiley</td>
<td>4.0</td>
</tr>
<tr>
<td>Packer</td>
<td>4.0</td>
</tr>
<tr>
<td>Duffett</td>
<td>4.0</td>
</tr>
<tr>
<td>A. Cole</td>
<td>4.0</td>
</tr>
<tr>
<td>Cock</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Scorer: [Signature]
This is a flowchart for the scorebook at a cricket match

**Ball is bowled**

- **No wicket or run occurs.**
  - A dot is written in the bowler’s column.

- **Runs are scored.**
  - The number of runs scored is added to the score tally.
  - Were all the runs scored by the batsman?
    - **No**
      - Write the number of runs scored in the batsman’s column and the bowler’s column.
    - **Yes**
      - Was it the bowler’s wicket?
        - **No**
          - Write a dot in the bowler’s column, add the amount of runs scored to the relevant bye/leg byes/penalties extra column.
        - **Yes**
          - Write a W in the bowler’s column and fill in the name of the bowler in the wicket information.

- **A wicket is taken.**
  - A double lined arrowhead is written in the batsman’s column, the type of wicket and total runs scored by the batsman are written in the wicket information. The runs at the fall of wicket and the batsman out are written in the fall of wicket column.

**Was the ball a no ball/wide?**

- **Yes**
  - Write a cross for a wide or circle for a no ball in the bowlers column, put a number in the top right of the cross or centre of the circle indicating any extra runs scored off the bat. Add any runs off the bat to the batsman’s column. Write the runs scored in the relevant no ball/wide extras column.

- **No**
  - Was it the bowler’s wicket?
    - **No**
      - Write a dot in the bowler’s column.
    - **Yes**
      - Write a W in the bowler’s column and fill in the name of the bowler in the wicket information.
This was the interview with Mr

- How do you normally transfer the data from the scorebook to the website?
The information from the scorebook is manually typed up onto the intranet, and then copied to a spreadsheet with formula's on to work out statistics.

- How reliable and accurate is the information that you upload?
Errors are often made typing the data up firstly due to misinterpretation of the scorebook or just general human error.

- How often do you upload the data?
Once a week on the Thursday morning after the staff game, it takes over half an hour to complete the whole process.

- How is the information presented?
Every match innings is displayed by 2 tables, one for batting figures and one for bowling figures, there is also a separate table for overall season figures.

- Does the information have any specific use other than for general viewing?
Not really, occasionally an email is sent round notifying users that the site has been updated.

- What master files do you use?
The master file is basically the file is on the intranet as there is no actual file with all the data stored. So the scorebook remains the only thing which holds all the data to be referred to.

- What would you want the new system to do?
The new system would as far as possible automatically update the statistics and the recent scorecards onto firefly either by automatically uploading them or just simply creating a file which can be copied onto the intranet.
Overall the new system should greatly speed up the process of uploading the scorecard onto the intranet.
Origins and Forms of data

Data Dictionary for the information taken from the scorebook in 1 innings:

<table>
<thead>
<tr>
<th>Name of Data</th>
<th>Type of data</th>
<th>Size of data (Characters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player Name (Batsman or Bowler)</td>
<td>Text</td>
<td>5-15 per player</td>
</tr>
<tr>
<td>Date of the match</td>
<td>Date/Time</td>
<td>6</td>
</tr>
<tr>
<td>Opposing team</td>
<td>Text</td>
<td>5-15</td>
</tr>
<tr>
<td>Total Extras</td>
<td>Number</td>
<td>3</td>
</tr>
<tr>
<td>Total runs scored with number of wickets taken</td>
<td>Number</td>
<td>5</td>
</tr>
<tr>
<td>Bowlers figures</td>
<td>Number</td>
<td>4-7 per player</td>
</tr>
<tr>
<td>Batsman's Total</td>
<td>Number</td>
<td>1-3 per player</td>
</tr>
<tr>
<td>Wicket Information</td>
<td>Text</td>
<td>5-20 per batsman that batted</td>
</tr>
<tr>
<td>Overs Bowled</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Ball Information</td>
<td>Text</td>
<td>10</td>
</tr>
</tbody>
</table>

Fact Recording Justification

My main method of fact finding here has been the interviews, due to the fact that the cricket scorebook is the most common way to score a school standard cricket match, I felt it was important to use interviews mainly to find out directly from the user what the problems are with such a widely used book, so therefore I have based most of my analysis on these interviews.

I have however also used some documentation from an actual match which showed me how these problems appear and how they can be avoided, whilst also giving me various ideas about how the new system could work. It has also provided me with the information for the data dictionary and the flowchart.

Restatement of Problem

Having investigated and gathered information about the current system I can come to the following conclusions, which restate the problem:

- The current manual scorebook system is difficult to use and any user will need some training to be taught how to use it.

- From the interview I found that there is no full universal set of symbols used in the scorebook, different users may record the same result by a different symbol. This can then cause further problems when using the scorebook for further analysis of the game at a later date.

- In the interview said that often during the cricket game he will find he cannot identify the current batsman or bowler, however as the game doesn't stop he must simply record what has happened in the column that he thinks to be correct. This however can lead to errors in the scoring. When these errors are later identified, the score from the incorrect column has to be laboriously erased and re-recorded onto the correct column.

- has said that the task of typing up the results onto the intranet from the handwritten scorebook for several reasons: The symbols and numerals written onto the scorebook are often difficult to read, whilst also some scorers will record information such as when a batsman faces a dot ball, whilst others won't so the information becomes inaccurate. Whilst also the method by which the results are typed up is inconsistent (sometimes the results will be written onto a spreadsheet first, sometimes they won't)

- Because there is no master spreadsheet created containing all the results, the intranet is the only place that holds all the data collected, and should this fail then all the data would be lost.
Requirement Specification

Overview

A system that allows the user to enter the actions of a cricket match easily and reliably, the user can then easily see the current score and all other information relevant to the game as it progresses. After a game has finished the system will summarise all the data from the game and generate updated statistics for the players that played in the game. So that it can be easily copied onto the intranet site.

The system will also contain a database with all the teams that have played a match scored by this system, including details of all the players that played. The system will be able to create a set of statistics for these players upon request.

Output requirements

The system should be able to show all the relevant information about a cricket match as the match is being played, this will include the total number of runs, total number of wickets, overs bowled, each batsman’s current total, last innings score, current run rate, required run rate, extras, last batsman’s score, along with several other options to see overall match statistics and bowling figures.

Once the match has finished the system should be able to create a set of summary statistics on a spreadsheet so that it can be copied easily onto the intranet. This spreadsheet should also be in a printable form so that hard copies can also be created.

At any time the system should be able to generate a set of statistics for any player on screen.

Input requirements

The main input will be at the start of the match, name of each player, the date of the match, the name of the opposing team, the number of overs that

Processing requirements

When any input is given after a ball the system will update the scoreboard to reflect what happened that ball, as the objective of the system is to make scoring an easier task, the majority of outcomes from a ball will require one keystroke or mouse click to input them. To allow this to happen the following processes will occur:

- The system will ask for the names of all the players at the start of the match for at least the BGS staff team, the names of the opposition team can be entered separately when that relevant player comes into play during the game or at the end of the match.
- The system will always know which batsman is facing so that if runs are scored off the bat then the system will add the runs to the correct batsman.
- The system will add runs to the total whenever they are inputted.
- The system will add runs to the correct section of the scoreboard (either batsman or the extras total).
- When a wicket is taken the scoreboard will add that wicket on to the total number of wickets.
- When a wicket is taken the system will ask for the name of the new batsman which can be selected from a list of batsmen or entered by the user.
- The system will have a separate window with a full scorecard of the whole match.
- The scorecard will display 2 tables per innings: one table containing the batting figures and the other the bowling figures.
- The batting table will show the runs scored by a batsman, and if that batsman is out; the method by which they were out.
The bowling table will show each bowler which has bowled, with the number of runs they have had scored against them, the number of overs they have bowled, the number of wickets they have taken, and the number of maidens they have bowled.

At the end of the match the system will produce 4 tables, a batting and bowling table from each innings, along with a summary statement.

Both users have agreed to these requirements.

Estimation of file sizes

This is the data dictionary from earlier in the analysis; I will now use it to work out how much storage the system will use.

<table>
<thead>
<tr>
<th>Name of Data</th>
<th>Type of data</th>
<th>Size of data (Characters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player Name (Batsman or Bowler)</td>
<td>Text</td>
<td>5-15 per player</td>
</tr>
<tr>
<td>Date of the match</td>
<td>Date/Time</td>
<td>0</td>
</tr>
<tr>
<td>Opposing team</td>
<td>Text</td>
<td>5-15</td>
</tr>
<tr>
<td>Total Extras</td>
<td>Number</td>
<td>3</td>
</tr>
<tr>
<td>Total runs scored with number of wickets taken</td>
<td>Number</td>
<td>5</td>
</tr>
<tr>
<td>Bowlers figures</td>
<td>Number</td>
<td>4-7 per player</td>
</tr>
<tr>
<td>Batsman's Total</td>
<td>Number</td>
<td>1-3 per player</td>
</tr>
<tr>
<td>Wicket Information</td>
<td>Text</td>
<td>6-20 per batsman that batted</td>
</tr>
<tr>
<td>Overs Bowled</td>
<td>Number</td>
<td>2</td>
</tr>
<tr>
<td>Ball Information</td>
<td>Text</td>
<td>10</td>
</tr>
</tbody>
</table>

From the data dictionary I can calculate the maximum number of bytes that potentially could be needed per table:

Batting table:

- 15x11 batsman: 155 bytes
- Total Extras: 3 bytes
- Total runs/wickets: 5 bytes
- 3x11 batsman totals: 30 bytes
- 20x11 Wicket information: 200 bytes
- Overs bowled: 2 bytes
- Total bytes: 406 bytes

Bowling Table:

- 15x10 bowlers: 150 bytes
- 2x10 Overs bowled: 20 bytes
- 2x10 Runs conceded: 20 bytes
- 1x10 Wickets taken: 10 bytes
- 1x10 Maidens bowled: 10 bytes
- Total bytes: 210 bytes

Scorecard:
6x50 Overs 300
300x10 Ball Information 3000

Extra info:

Date 6
Opposing team 15
Total bytes 21

For every match two batting tables, two bowling tables, two Scorecards are required. So the system will need 405 + 405 + 210 + 210 + 3000 + 3000 bytes or 843 KB.

Assuming 12 matches a season are played, the storage of match data amount considering most modern Laptops which this system will be used on, 000GB.

Alternative Approaches

There are several different approaches I can take towards this system. I have decided that using Visual Basic will be my best option as it is a high level language to perform all the necessary tasks in the requirement specification.

There are other options which I have considered; one of these is to create a spreadsheet based system. Although it follows all the requirements it will not be ideal for low standard cricket games where there are limitations to the interface due to the grid aspect of the spreadsheet and also contains features that are not needed for low standard cricket games.

Below is a screenshot of the L&W system, due to the many features it is very compact with a lot of information being displayed on one window. It would be ideal, for a simple club game it is not needed and much of the information is not required. So I intend to make an interface which is simpler and easier to use than this one, so that it requires less training to use and reduces the amount of information about the match.
There are other off the shelf packages that could be used that fill all the requirements specification such as PitchPad, this system is far more complex than what is given in the requirement specification. A piece of software like PitchPad will require the user to be trained to use the software, whilst also requiring a lot of attention whilst the match is being played. For example the screenshot below shows the lower wheel diagram the system creates. Although this would be a nice feature for a system to have, it would require the scorer to enter where the batsman's shot has travelled every time a run is scored off the bat. Whilst one of the main objectives of the system is to make scoring simple and easy, with the user not having to focus so much on the scoring, so they can enjoy watching to game.

Due to the complexity of the PitchPad system any user wanting to use the system will require training of some sort, also the amount of data the scorer has to enter every ball may cause the scorer to lose track of the game or make mistakes with the information they do enter. This is exactly what the new system would be trying to prevent.

**Selected Approach**

My selected approach will therefore be Visual Basic as it will allow me to fulfil all the requirements whilst also being flexible and efficient. The visual part of the program makes the process of creating an interface far less time consuming, whilst the basic style of high level language also makes writing the code a simpler task.

In summary the benefits of the system will be that:

- It will save a large amount of time when transferring the score onto the school intranet, and will also save time when a cricket match is being scored.
- Once made, the system can easily be further tailored to the requirements of the user.
- The system will require little or no training for a first time user, it will also allow an inexperienced cricket scorer to score a match where as they would have been unconfident to do so with a scorebook.

Both my users have recognised and agreed with these points.

---

**Hardware and Software Requirements**

Table of hardware and software required with justifications

<table>
<thead>
<tr>
<th>Software</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 98</td>
<td>Operating system required to run VB 6.0</td>
</tr>
<tr>
<td>Visual Basic 6.0</td>
<td>Software selected to create and run the system.</td>
</tr>
<tr>
<td>Laptop with keyboard and mouse pad</td>
<td>A laptop will be required to allow the system to be brought to the cricket pitch side.</td>
</tr>
<tr>
<td>Spare Battery</td>
<td>Required if the battery life of the original battery is not likely to last the full match.</td>
</tr>
<tr>
<td>Network connection</td>
<td>Required to upload the scorecard onto the intranet.</td>
</tr>
</tbody>
</table>
Section 2 – Design

Design Objectives

The system will need to achieve the following objectives:

- On starting a game the user will be asked to fill in the names of all the players playing in the match, either by entering a new name, or choosing from the database.
- The user will be able to change features about the game prior to the start such as number of overs and number of runs for a wide.
- Once the game starts the user can score either by using the mouse or by key strokes, every ball will require no more than 3 key strokes/mouse clicks to enter the outcome.
- The user will be able to undo any action they perform, whilst also being able to edit any piece of data they have entered.
- The user will also be able to manually override the automatic features such as the end of the over and which batsman is facing.
- The system will show a real time scoreboard of the match as it is in progress, so any data that is entered will immediately change the scoreboard and scorecard as necessary.
- The real time scoreboard will display all the features of a standard cricket scoreboard: The total number of runs scored, the number of wickets taken, the number of overs bowled, the total runs scored by each batsman, the score of the previous innings. Along with features of a more advanced scoreboard:
  - Last fall of wicket, how the previous batsman was out, and previous batsman’s total & number.
  - Alongside the main scoreboard will be a sub scoreboard showing the current bowler who is facing, the current batsman, what has happened this over, the run rate and the required run rate.
  - The system stores all the statistics about any player that has been scored by the system, this database can be accessed through the system to be searched.
  - The system will also create a spreadsheet that stores all the statistics for the players of the BGS staff team, with the summary statistics for each match.
  - During a match the main window will be the window with the scoreboard, there will be buttons that provide access to 2 other windows, one window will show the scorecard, the other will provide access to the database.
  - At the end of a match the system will produce a summary of a match in a spreadsheet that includes the batting and bowling figures for each innings, the complete scorecard, date and name of opposition.

Design Specification

When the scorer wishes to start a match they will go through the following process:

Input

```
| Enter Match Date and the Opposition |
```

```
Enter the names of the all players from both teams (with the option to select players from the database or enter a new player) |
```
Ability to change the settings: Runs per no ball/wide and set the number of overs the match is limited to.

Start Game

Enter which team is batting, and choose from that team's list which batsmen are opening, and which is facing first.

Enter the opening bowler from the fielding side.

Once this process has been done, the game will start and the main scoreboard will be shown. The flow chart for this window is on the next page.
Input, Processing & Output

**Main Interface (Scoreboard)**

- **Dot Ball**: Add no runs ($x=0$)
- **1 Run Scored**: Add 1 run ($x=1$)
- **2 Runs Scored**: Add 2 runs ($x=2$)
- **3 Runs Scored**: Add 3 runs ($x=3$)
- **4 Runs Scored**: Add 4 runs ($x=4$)
- **5 Runs Scored**: Add 5 runs ($x=5$)
- **6 Runs Scored**: Add 6 runs ($x=6$)

**Wicket**: A pop up requesting the method by which the batsman was out.

**Uno**: Deletes the last action made.

**Redo**: Replaces the action undone (only usable after Undo)

**Add x onto the Extra**: Add $x$ to the balls bowled this over except for a wide/no ball, in this case add nothing.

**Penalties**: Add the runs scored from the penalty ($x = \text{total runs scored from the penalty}$)

**Leg Bye**: Add the number of runs scored from the leg bye ($x = \text{total runs scored this ball}$)

**Bye**: Add the number of runs scored from the bye ($x = \text{total runs scored this ball}$)

**Wide**: Add the number of runs scored for a wide ($x = \text{runs scored this ball}$)

**No Ball**: Add the number of runs scored from a no ball, with any extra runs scored off the bat ($x = \text{total runs scored this ball}$)

**Add x onto the Facing batsman's total**: $x = \text{odd then switch the batsman who is facing. Update the run rate and required run rate accordingly. Add 1 to the balls bowled this over.}$

**When the balls bowled this over reaches 6, add 1 to the overs bowled and request the bowler of the bowler to bowl the next over.** When requesting the bowler suggest the bowler of the over before last.

**The number, total runs scored, wicket method and fall of wicket are shown on the scoreboard. 1 is added to the total number of wickets.**

In addition, when the overs bowled reaches the number of limited overs or the number of wickets reaches 10 than the innings is over. The scorer gets a message saying the innings is complete; they can then close down the program to have a break and then re-open the program to carry on with the match when the next innings starts.
If the program is closed down in the middle of a match at any point, (due to the nature of my data structure which will be explained later) the user will be asked on re-opening the program whether they wish to carry on with the last match which was saved, or to start a new match.

This means the user will be able to close down the program and completely turn off their laptop in between innings to save on battery. It also means should the user’s laptop run out of battery, or the user inadvertently close down the program the data up to the last ball entered will be saved.

When the user indicates that the second innings of a match is about to start they will be prompted with the following flowchart:

Once the second innings has started the main scoreboard will display the following additional information:
- The total runs scored in the first innings
- The run rate required by the batting team in the second innings

Once either team has won the match, the following flowchart will take place:

**User Interface**
The next pages contain my hand drawn interface designs for the system.
End of Innings Pop Up

Team \(xx\) scored \(yy\) runs.

Pause Start next innings

End of match Pop up

Team \(xx\) won by \(xx\) runs/wickets.

Undo Finish Summary Statistics

Finish the game taking the scoresheet.

Player: Statistics

Search box and button to search for a player.

Search

Seat Record Card

Player:

Bowling

Over Maidens Runs Wickets Econ. Rate Strike Rate

Batting

Innings Not Out Highest Score Runs Ave 00 50

Table showing the batting statistics for the player.

Button return to the scoreboard.

Button showing the statistics for the player.
Player Search Pop Up

Player List

Selected Players

A list of the players that have been selected.

Button to take a player out of the highlighted player, when players selected in the selected players list.

Button to accept the selected players list.

End of over Pop up

Drop Down list to select the next bowler.

Button to OK the selected bowler.

Settings Window

Limited Overs

Non-Overs

Extra Ball

Radio buttons: Tick box if to show how many overs remain.

The Button to return to the previous window.

The Button to select the next bowler.

Match Start Pop up

Facing Batsman

Other Batsman

Opening Bowler

Dropdown list to select the opening bowler.
Wicket Pop up

- Bowled
- Caught
- LBW
- Stumped
- Run Out
- Retired
- Batsmen Crooked

Done Cancel

Run Out Popup

Which Batsmen?
- Batsman A
- Batsman B

OK Cancel

Next Batsman

DropDown list to indicate the next batsman after a wicket

Bye/Leg Bye/Wide/No Ball/Penalties

How many runs scored?
- None
- 0
- 1-4
- 5-9

Manuel Override

Batsmen have crossed
End Over

Proceed Previous Over
Cancel

If any extra runs then the user can select how many extra runs were scored off the last of any
Interview

After another interview with """" reviewing the interface design and the overall structure of the system, I made the following changes which are made in pen on the interface designs:

- The Bating Table on the Scorecard will now contain a balls faced column.
- The player statistics will show the statistics for the current game if the player searched for is playing in the game currently in progression, otherwise it will just show the statistics for the selected player for the season.

Files and Data Structures

The file structure has been changed from the analysis section. Two files will be created per match, one with the summary statistics which includes the battery and bowling tables, and one with the scorecard showing what happened every ball.

File Name: [date]summary
Purpose: To store the statistics from a cricket match as it is played

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description of Contents</th>
<th>Data type</th>
<th>Length</th>
<th>Sample values / range / set</th>
<th>Type of validation expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>The Date of the Match</td>
<td>Date</td>
<td>6</td>
<td>21/06/09</td>
<td>NA</td>
</tr>
<tr>
<td>Opposition</td>
<td>The Opposing team</td>
<td>Text</td>
<td>20</td>
<td>QEH Staff</td>
<td>NA</td>
</tr>
<tr>
<td>Player10</td>
<td>Primary Key</td>
<td>AutoNum</td>
<td>3</td>
<td>21</td>
<td>Auto-Generated</td>
</tr>
<tr>
<td>PlayerName</td>
<td>The name of the player</td>
<td>Text</td>
<td>30</td>
<td>John Smith</td>
<td>NA</td>
</tr>
<tr>
<td>PlayerTeam</td>
<td>The players team</td>
<td>Text</td>
<td>20</td>
<td>BGS Staff</td>
<td>NA</td>
</tr>
<tr>
<td>Battingruns</td>
<td>The number of runs scored</td>
<td>Integer</td>
<td>3</td>
<td>58</td>
<td>NA</td>
</tr>
<tr>
<td>BattingBallsFaced</td>
<td>Number of balls faced</td>
<td>Integer</td>
<td>3</td>
<td>105</td>
<td>NA</td>
</tr>
<tr>
<td>BattingHowOut</td>
<td>How the player was out while batting</td>
<td>Text</td>
<td>10</td>
<td>Bowled</td>
<td>NA</td>
</tr>
<tr>
<td>BattingWktBowler</td>
<td>The name of the bowler who took the wicket</td>
<td>Text</td>
<td>30</td>
<td>John Smith</td>
<td>NA</td>
</tr>
<tr>
<td>Bowlingovers</td>
<td>The number of overs bowled by a player</td>
<td>Integer</td>
<td>2</td>
<td>5</td>
<td>NA</td>
</tr>
<tr>
<td>Bowlingruns</td>
<td>The number of runs scored against a player</td>
<td>Integer</td>
<td>2</td>
<td>16</td>
<td>NA</td>
</tr>
<tr>
<td>Bowlingwks</td>
<td>The number of wickets taken by a player</td>
<td>Integer</td>
<td>1</td>
<td>3</td>
<td>NA</td>
</tr>
<tr>
<td>Bowlingmaidens</td>
<td>The number of maidens bowled by a player</td>
<td>Integer</td>
<td>1</td>
<td>2</td>
<td>NA</td>
</tr>
</tbody>
</table>
Size of Individual Record: 111 Bytes
Maximum Records stored: 22
Total size required for data: 2442 Bytes
Overheads (10%): 2686 Bytes
Total size required: 0.0026 MB
<table>
<thead>
<tr>
<th>Fieldname</th>
<th>Description of contents</th>
<th>Data Type</th>
<th>Length</th>
<th>Sample values/range/set</th>
<th>Type of Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ballno</td>
<td>The number of the ball</td>
<td>Integer</td>
<td>3</td>
<td>14</td>
<td>NA</td>
</tr>
<tr>
<td>Runs</td>
<td>Number of runs scored off the ball</td>
<td>Integer</td>
<td>1</td>
<td>2</td>
<td>NA</td>
</tr>
<tr>
<td>Batsman</td>
<td>The number of the batsman who scored the runs</td>
<td>Integer</td>
<td>2</td>
<td>8 (0 = neither batsman scored the runs)</td>
<td>NA</td>
</tr>
<tr>
<td>Bowler</td>
<td>The number of the bowler who bowled the ball</td>
<td>Integer</td>
<td>2</td>
<td>4</td>
<td>NA</td>
</tr>
<tr>
<td>Extra</td>
<td>The type of extra scored</td>
<td>Integer</td>
<td>1</td>
<td>0 = no extra, 1 = wide, 2 = no ball, 3 = bye, 4 = leg bye, 5 = penalties</td>
<td>NA</td>
</tr>
<tr>
<td>Wicket</td>
<td>The type of wicket off the ball</td>
<td>Integer</td>
<td>2</td>
<td>0 = no wicket, 1 = bowled, 2 = caught, 3 = lbw, 4 = stumped, 5 = run out, 6 = obstruction, 7 = handling ball, 8 = hit ball twice, 9 = timed out, 10 = hit wicket, 11 = retired</td>
<td>NA</td>
</tr>
<tr>
<td>Batsmanout</td>
<td>The number of the batsman who was out from the wicket</td>
<td>Integer</td>
<td>2</td>
<td>5 (0 = no batsman out)</td>
<td>NA</td>
</tr>
</tbody>
</table>

Size of Individual Record: 13
Maximum Records Stored: 2,000
Total size required for data: 26,000 bytes per match
Overheads (10%): 26,260
Total size required: 0.0256 MB

I will assume that every over will not exceed 20 balls, so assuming 5 ball overs the first over will use ball numbers 1 to 6, second over ball numbers 21 to 26 and so on. This will allow the system to easily identify which over the ball was bowled in by looking at the ball number.
File Name: CricketMasterFile
Location:
Purpose: To store the overall statistics for players who have played with the system.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description of contents</th>
<th>Data type</th>
<th>Length</th>
<th>Sample Values/range/set</th>
<th>Type of validation expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlayerID</td>
<td>Primary Key</td>
<td>AutoNumber</td>
<td>3</td>
<td>54</td>
<td>Auto-generated</td>
</tr>
<tr>
<td>PlayerName</td>
<td>The name of the player</td>
<td>Text</td>
<td>30</td>
<td>John Smith</td>
<td>NA</td>
</tr>
<tr>
<td>PlayerTeam</td>
<td>The team that the player plays for</td>
<td>Text</td>
<td>20</td>
<td>BGS Staff</td>
<td>NA</td>
</tr>
<tr>
<td>Innings</td>
<td>Number of innings played</td>
<td>Integer</td>
<td>2</td>
<td>6</td>
<td>NA</td>
</tr>
<tr>
<td>AverageRuns</td>
<td>The average runs scored</td>
<td>Integer</td>
<td>3</td>
<td>34</td>
<td>NA</td>
</tr>
<tr>
<td>HighScore</td>
<td>The highest score achieved</td>
<td>Integer</td>
<td>3</td>
<td>69</td>
<td>NA</td>
</tr>
<tr>
<td>NotOuts</td>
<td>The number of times not out</td>
<td>Integer</td>
<td>2</td>
<td>4</td>
<td>NA</td>
</tr>
<tr>
<td>Overs</td>
<td>How many overs bowled</td>
<td>Integer</td>
<td>2</td>
<td>13</td>
<td>NA</td>
</tr>
<tr>
<td>Runs</td>
<td>How many runs scored against</td>
<td>Integer</td>
<td>3</td>
<td>102</td>
<td>NA</td>
</tr>
<tr>
<td>Wickets</td>
<td>How many wickets taken</td>
<td>Integer</td>
<td>2</td>
<td>8</td>
<td>NA</td>
</tr>
<tr>
<td>Maidens</td>
<td>How many maidens bowled</td>
<td>Integer</td>
<td>2</td>
<td>5</td>
<td>NA</td>
</tr>
<tr>
<td>Economy Rate</td>
<td>The economy rate</td>
<td>Single</td>
<td>4</td>
<td>6.42</td>
<td>NA</td>
</tr>
<tr>
<td>Strike Rate</td>
<td>The strike rate of a player</td>
<td>Single</td>
<td>4</td>
<td>9.23</td>
<td>NA</td>
</tr>
</tbody>
</table>

Size of Individual Record: 80
Maximum Records Stored: 1,000
Total size required for data: 80,000
Overheads (10%): 8,000
Total size required: 0.0859 MB

Virtually all the fields require no validation, this is because the data generated for these fields is processed and inputted by the system, meaning the values entered are controlled.
Data Structure

When a match is being played, I will be using the following variables:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Length</th>
<th>Sample Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalRuns</td>
<td>The total runs scored this innings</td>
<td>Integer</td>
<td>3</td>
<td>123</td>
</tr>
<tr>
<td>TotalWks</td>
<td>The total wickets this innings</td>
<td>Integer</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>BatATotal</td>
<td>Runs scored by Batsman A</td>
<td>Integer</td>
<td>3</td>
<td>54</td>
</tr>
<tr>
<td>BatBTotal</td>
<td>Runs scored by Batsman B</td>
<td>Integer</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>Overs</td>
<td>Total Overs bowled</td>
<td>Integer</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>LastInns</td>
<td>Runs scored last innings</td>
<td>Integer</td>
<td>3</td>
<td>180</td>
</tr>
<tr>
<td>Extras</td>
<td>Total extras scored</td>
<td>Integer</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>LMruns</td>
<td>Runs scored by the last man in</td>
<td>Integer</td>
<td>3</td>
<td>103</td>
</tr>
<tr>
<td>LMinum</td>
<td>Number of the last man in</td>
<td>Integer</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>LMout</td>
<td>How the last man in was out</td>
<td>Text</td>
<td>17</td>
<td>Bowled</td>
</tr>
<tr>
<td>LNFow</td>
<td>Number of runs when last wicket fell</td>
<td>Integer</td>
<td>3</td>
<td>23</td>
</tr>
<tr>
<td>CurBat</td>
<td>Name of the current batsman facing</td>
<td>Text</td>
<td>30</td>
<td>Smith</td>
</tr>
<tr>
<td>CurBowl</td>
<td>Name of the current bowler</td>
<td>Text</td>
<td>30</td>
<td>Jones</td>
</tr>
<tr>
<td>RunRate</td>
<td>Current run rate</td>
<td>Single</td>
<td>4</td>
<td>4.29</td>
</tr>
<tr>
<td>RfRunRate</td>
<td>Required run rate to win</td>
<td>Single</td>
<td>4</td>
<td>8.13</td>
</tr>
<tr>
<td>Bowled</td>
<td>Number of balls bowled in the over</td>
<td>Integer</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

During the match, after every ball the system will update the summary file and the statistics file. This means the system will always have an up to date file of the match so far. This will prevent any data loss from crashes etc.
This is the structure I will use to store the files:

- Main Folder
  - Main Program
  - CricketMasterFile
- Data Store
  - Scorecard files
  - Summary files (txt format)
  - Spreadsheets
    - Summary files (Spreadsheet format)

Error Messages

The design for an error message is on the interface section. Due to the nature of the system there are very few errors that can be made, these are the errors that can be made:

<table>
<thead>
<tr>
<th>Message Reson</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selecting more than 1 team when searching</td>
<td>More than 1 team has been selected</td>
</tr>
<tr>
<td>Selecting more than 11 players in the search</td>
<td>More than 11 players have been chosen</td>
</tr>
<tr>
<td>Not selecting the opening batting team in the settings</td>
<td>Select a team to open the batting</td>
</tr>
<tr>
<td>Not selecting one or more of the players to open the batting and bowling</td>
<td>You have not selected one or more of the option(s)</td>
</tr>
<tr>
<td>Attempting to view the scorecard before a complete over.</td>
<td>The scorecard cannot be shown until a full over is bowled.</td>
</tr>
</tbody>
</table>
Limitations of the System

Hardware:
A video camera could be used to record every ball visually, however this would significantly increase the memory usage and cost of the system.
As the system will be used on a cricket pitch side, there is never going to be a guarantee that mains power is available. This means that the user will always have to limit the usage of the system by the amount of battery life they have access to. If they do not have a spare battery and their standard battery life is only 1 to 2 hours it is likely the battery will run out before the end of the match. The user will also not be able to access the internet from the cricket pitch side, if they could have it would mean the cricket score could be updated live on the internet as the match is played.

Software:
As the output will be given in an excel spreadsheet, anyone trying to use the system without this software will not be able to view the statistics.

User:
If the user is new to computers then it may take them a long time to input the correct option after each ball, especially after balls which require additional options to be selected such as a wide ball. This means some parts of the screen will need more explanation than is required for a more advanced user of computers.

Imposed:
The user has imposed that the statistics given after the game must be produced on a spreadsheet, whilst other viable methods are available.

Storage Limitations

This is a sample layout for the sizes, using the assumption of 12 matches per year this would be the expected file size required:

<table>
<thead>
<tr>
<th>Filename</th>
<th>No of Records</th>
<th>File Size</th>
<th>No of Records</th>
<th>File Size</th>
<th>No of Records</th>
<th>File Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>CricketMasterFile</td>
<td>1,000</td>
<td>88,000</td>
<td>1,000</td>
<td>88,000</td>
<td>1,000</td>
<td>88,000</td>
</tr>
<tr>
<td>Summary</td>
<td>264</td>
<td>32,232</td>
<td>528</td>
<td>64,484</td>
<td>792</td>
<td>96,696</td>
</tr>
<tr>
<td>Statistics</td>
<td>24,000</td>
<td>315,120</td>
<td>48,000</td>
<td>630,240</td>
<td>72,000</td>
<td>945,360</td>
</tr>
</tbody>
</table>

Total File Size Required Year 1: 0.43MB
Total File Size Required Year 2: 0.76MB
Total File Size Required Year 3: 1.1MB
Section 3 – Software Development, Testing and Implementation

This is the table that I will use to create the system, doing each task in order:

<table>
<thead>
<tr>
<th>Task Number</th>
<th>Task</th>
<th>Date Started</th>
<th>Date Finished</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create the scoreboard interface</td>
<td>13/10/08</td>
<td>16/10/08</td>
</tr>
<tr>
<td>2</td>
<td>Create the data structure</td>
<td>16/10/08</td>
<td>16/10/08</td>
</tr>
<tr>
<td>3</td>
<td>Create the pop ups</td>
<td>16/10/08</td>
<td>17/10/08</td>
</tr>
<tr>
<td>4</td>
<td>Create the database</td>
<td>17/10/08</td>
<td>24/10/08</td>
</tr>
<tr>
<td>5</td>
<td>Program the buttons to perform their respective task</td>
<td>24/10/08</td>
<td>7/11/08</td>
</tr>
<tr>
<td>6</td>
<td>Create the Scorecard interface</td>
<td>7/11/08</td>
<td>21/11/08</td>
</tr>
<tr>
<td>7</td>
<td>Code the spreadsheets showing end game statistics</td>
<td>21/11/08</td>
<td>25/11/08</td>
</tr>
</tbody>
</table>
Test Plan

Due to the nature of the system, I will take a different approach to testing it. I will play through a whole game and test the outputs of as many different situations as possible. To do this I will use the following test plan.

I will be simulating a 5 over match where the BGS team bats first, bats out all their overs. The test opposition team then is all out and the BGS team win.

Indicates a button press]

1. [Press start game]
2. Date: 31 July 2008 (Extreme data for day chosen)
3. Teams: BGS Staff and Test Opp (picked from the database)
4. BGS staff – Ian Rolling, Colin Waday, Andy Keen, Andy Barker, Graham Clark, Justin Harford (chosen from the database) Ben Scott, Kevin Blackburn, Andrew Flintoff, Kevin Peterson, Steve Hamason (added as new players)
5. Test Opp – Opp one, Opp two, Opp three, Opp four, Opp five, Opp six (Added from database)
   Thirty Character Name For Test (extreme data for a name), New Opp eight, New Opp nine, New Opp ten, New Opp eleven (added as new players)
6. [Press Done]
7. BGS Staff to bat first (radio button selected)
8. One run per wide/no ball (radio button selected)
9. There will be an extra run for a wide/no ball (Check box ticked)
10. Limited to 5 overs [Text box changed to 5]
11. [Press Done]
12. Ian Rolling is the opening batsman. Andy Barker is the other batsman.
13. Opp one is the opening bowler.
14. [Press Done]
15. [Press Start Game]
16. Over 1: dot, dot, dot, dot, dot, dot, dot
17. Opp three bowls the next over
18. Over 2: 1, 2, 3, 4, 5, 6
19. New Opp eight bowls the next over
20. Over 3: Bye(1), Bye(5), Leg Bye(1), Leg Bye(6), Penalty(1), Penalty(6)
21. Opp two bowls the next over
22. Over 4: Wide(0), No Ball(0), 2 [Press Undo] [Manual Override, Batsmen Swap Ends], dot, Wide(6), dot, dot, 6, 4, dot
23. Opp one bowls the next over
24. Over 5: Wicket(Bowled, next batsman Colin Waday), Wicket(Caught, next batsman Andy Keen), Wicket(LBW next batsman Graham Clark), 3 (reduced to 2 by short run), 5,
25. [Press Scorecard]
26. [Press Back]
27. Over 6 continued: 5
28. [Press OK]
29. Opp five will open the batting. Thirty Character Name For Test will be the other batsman.
30. Ian Rolling will open the bowling.
31. [Press Done]
32. [Press Start Next Innings]
33. Over 1: Wicket( Run Out, Opp five is out after running 2 runs, next batsman: Opp one), Wicket(Caught, next batsman: Opp two), 2, 3, dot, dot
34. Andrew Flintoff will bowl the next over
35. Over 2: Wicket(Bowled, next batsmen: Opp three), Wicket(Obstruction, next batsman: Opp four), Wicket(Timed Out, next batsman: Opp six), Wicket(Hit Ball Twice, next batsman: New Opp eight), Wicket(Handing the ball, next batsman: New Opp nine), Wicket(Retired, next batsman: New Opp ten) [Prolong Over], dot
36. **Kevin Blackburn** will bowl the next over
37. Over 3: dot, 1, 2, Wicket(LBW), next batsman: New Opp eleven, 6, Wicket(Bowled).
38. [Press Ok]
39. Go to the Spreadsheets Folder and Open the spreadsheet for the match.
40. Open the Master File Spreadsheet.

### Test Detail

<table>
<thead>
<tr>
<th>Test #</th>
<th>Test name</th>
<th>Input Data</th>
<th>Expected Result</th>
<th>Output (SS = Screenshot)</th>
<th>Actual Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Test Date entry</td>
<td>31 July 2008</td>
<td>Displayed in Match Spreadsheet</td>
<td>SS 49</td>
<td>As Expected</td>
</tr>
<tr>
<td>3</td>
<td>Test team chosen from database</td>
<td>Test Opp</td>
<td>Displayed in opposition text box greyed out</td>
<td>SS 4</td>
<td>As Expected</td>
</tr>
<tr>
<td>4</td>
<td>Some data chosen from database, other player names entered</td>
<td>Names listed in plan #4</td>
<td>New players added to the database, existing players greyed out</td>
<td>SS 5 &amp; SS 6</td>
<td>As Expected</td>
</tr>
<tr>
<td>5</td>
<td>Some players chosen from the Test Opp, other new players added</td>
<td>Names listed in plan #5</td>
<td>New players added to the database, existing players greyed out</td>
<td>SS 7 &amp; SS 12</td>
<td>As Expected</td>
</tr>
<tr>
<td>7</td>
<td>BGS selected to bat first</td>
<td>BGS radio button selected</td>
<td>BGS players listed to bat first, Test Opp listed to bowl first</td>
<td>SS 8 &amp; SS 9</td>
<td>As Expected</td>
</tr>
<tr>
<td>8</td>
<td>One run per wide/no ball selected</td>
<td>1 radio button selected</td>
<td>One extra run is added whenever a wide/no ball occurs</td>
<td>SS 8, SS 31 &amp; SS 32</td>
<td>As Expected</td>
</tr>
<tr>
<td>9</td>
<td>Extra ball for a wide/no ball selected</td>
<td>Check box ticked</td>
<td>An extra ball will be given when a wide/no ball occurs</td>
<td>SS 8</td>
<td>As Expected</td>
</tr>
<tr>
<td>10</td>
<td>Limited Overs set to 5</td>
<td>5 entered in the text box</td>
<td>Innings finishes after 5 overs are bowled</td>
<td>SS 44</td>
<td>As Expected</td>
</tr>
<tr>
<td>12</td>
<td>Opening batsmen set</td>
<td>Ian Rolling and Andy Barker selected</td>
<td>Ian Rolling will be the first facing batsman, with Andy Barker as the other batsman.</td>
<td>SS 11 &amp; SS 14</td>
<td>As Expected</td>
</tr>
<tr>
<td>13</td>
<td>Opening bowler set</td>
<td>Opp one selected</td>
<td>Opp one appears as the first bowler</td>
<td>SS 11 &amp; SS 14</td>
<td>As Expected</td>
</tr>
<tr>
<td>16</td>
<td>First over data inputted</td>
<td>Dot, dot, dot, dot, dot, dot, dot</td>
<td>No value on the whole scoreboard will increase</td>
<td>SS 14</td>
<td>As Expected</td>
</tr>
<tr>
<td>17</td>
<td>Next bowler selected</td>
<td>Opp three selected</td>
<td>The overs value increases to one, The current bowler becomes Opp three</td>
<td>SS 15</td>
<td>As Expected</td>
</tr>
<tr>
<td>18</td>
<td>Second Over data inputted</td>
<td>1, 2, 3, 4, 5, 6</td>
<td>Each ball the Total increases by the runs scored, the total for the facing batsman increases by the runs scored. If the number of runs is odd</td>
<td>SS 17 – 22</td>
<td>As Expected</td>
</tr>
</tbody>
</table>
then the Facing Batsman swaps with the other batsman. The run rate increases to the number of runs divided by the number of overs every ball.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Third Over data Inputted</td>
<td>Bye(1), Bye(5), Leg Bye(1), Leg Bye(6), Penalty(1), Penalty(6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bye, Leg Bye and Penalty all bring up the extras window with the None radio button greyed out, once the number of runs is selected and OK is pressed the relevant number of runs is added onto the total runs and to the total extras. When the number of runs scored is 1 the batsman should swap ends</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SS 23 – 28 The Penalty window did not have the None button greyed out.</td>
</tr>
<tr>
<td>22</td>
<td>Fourth Over data Inputted</td>
<td>Wide(0), No ball(0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each of these balls should add 1 to the total and 1 to the extras</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SS 29 – 31 As expected</td>
</tr>
<tr>
<td>22</td>
<td>Fourth Over data Continued</td>
<td>2 Runs [Undo]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 2 runs added should be taken away from the total and the batsman scores</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SS 32 – 34 As Expected</td>
</tr>
<tr>
<td>22</td>
<td>Fourth Over data continued</td>
<td>[Batsman Swap Ends]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Swap the facing batsman without changing anything else</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SS 35 &amp; SS 36 As Expected</td>
</tr>
<tr>
<td>24</td>
<td>Fifth Over: Batsman bowled out</td>
<td>Wicket(Bowled), next batsman Colin Wadley selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Select the bowled option from the wicket menu to then bring up the next batsman menu. Once the batsman is chosen they should appear in place of the batsman who is out. The Last Man information should be updated accordingly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SS 39 - 41 As Expected</td>
</tr>
<tr>
<td>24</td>
<td>Fifth Over: 2 more batsman out</td>
<td>Wicket(Caught) and Wicket(LB W)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The same process should occur for the wickets, with the Last man information being updated correctly with the type of wicket</td>
</tr>
<tr>
<td></td>
<td></td>
<td>As Expected</td>
</tr>
<tr>
<td>24</td>
<td>Fifth Over: Check Scorecard</td>
<td>Click the Scorecard button</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See Below</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SS 45A &amp; 45 At first the scorecard bugged as shown in SS 45A, but was then fixed in SS 45</td>
</tr>
</tbody>
</table>

The scorecard should bring up a new window. This window contains statistics for the current inning match. The scorecard should when clicked in the test should contain the following data:
<table>
<thead>
<tr>
<th>Batsman</th>
<th>How Out</th>
<th>Bowler</th>
<th>Total</th>
<th>Balls Faced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ian Rolling</td>
<td>Bowled</td>
<td>Opp Five</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>Andy Barker</td>
<td>Not</td>
<td>Out</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>Colin Wady</td>
<td>Caught</td>
<td>Opp Five</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Andy Keen</td>
<td>LBW</td>
<td>Opp Five</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Graham Clark</td>
<td>Not</td>
<td>Out</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bowler</th>
<th>Overs</th>
<th>Runs</th>
<th>Wickets</th>
<th>Maidens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opp one</td>
<td>1</td>
<td>8</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Opp two</td>
<td>1</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Opp three</td>
<td>1</td>
<td>21</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>New Opp eight</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bowler</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opp one</td>
<td>000000</td>
<td>WW</td>
<td>WW</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Opp two</td>
<td>+0+60</td>
<td>0640</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opp three</td>
<td>123456</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Opp eight</td>
<td>000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test #</th>
<th>Test name</th>
<th>Input Data</th>
<th>Expected Result</th>
<th>Output</th>
<th>Actual Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>Innings finishes</td>
<td>The last ball of the innings</td>
<td></td>
<td>SS 44</td>
<td>As expected</td>
</tr>
<tr>
<td>32</td>
<td>Start second innings</td>
<td>Confirm the starting players</td>
<td></td>
<td>SS 46</td>
<td>As Expected</td>
</tr>
<tr>
<td>33</td>
<td>Over 1 data: Run Out</td>
<td>Wicket: Run Out</td>
<td>A run out window will pop up, allowing the user to select the batsman who is out and the runs scored. Upon completion the scoreboard should be updated correspondingly.</td>
<td></td>
<td>SS 47</td>
</tr>
<tr>
<td>35</td>
<td>Over 2 data</td>
<td>Various different wicket data is entered, the over is also prolonged</td>
<td></td>
<td>As Expected</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Over 3 data, end of match</td>
<td>The final wicket of the match is inputted</td>
<td></td>
<td>SS 49 (summary file), SS 50 (master file)</td>
<td>As Expected</td>
</tr>
</tbody>
</table>

These are screenshots showing the system as the test run took place. Some Screenshots do not directly relate to the testing, they just show parts of the system in action example Screenshot 48 shows how the list for new batsman coming in is reduced to only the remaining possible batsman rather than the whole team.
Press Start Game to start a new game

Start Game

Match Start - Enter teams

BGS Staff

Opposition

Player 1
Player 2
Player 3
Player 4
Player 5
Player 6
Player 7
Player 8
Player 9
Player 10
Player 11

Search BGS Players

Player 12
Player 13
Player 14
Player 15
Player 16
Player 17
Player 18
Player 19
Player 20
Player 21
Player 22

Search Opposition Players

Done
**Scorecard**

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Andy Barker</th>
<th></th>
<th>Total</th>
<th>Andy Barker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ian Rolling</td>
<td>0</td>
<td>00</td>
<td>For</td>
<td>0</td>
<td>00</td>
</tr>
<tr>
<td>Overs</td>
<td>0</td>
<td>00</td>
<td></td>
<td>0</td>
<td>00</td>
</tr>
<tr>
<td>Wickets</td>
<td>0</td>
<td>00</td>
<td>Last Innings</td>
<td>000</td>
<td></td>
</tr>
<tr>
<td>Extras</td>
<td>0</td>
<td>00</td>
<td></td>
<td>0</td>
<td>00</td>
</tr>
<tr>
<td>Last Man Name</td>
<td>_______</td>
<td>_______</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>000</td>
<td>000</td>
<td>How Out</td>
<td>_______</td>
<td></td>
</tr>
<tr>
<td>Fall of Wicket</td>
<td>000</td>
<td>000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Current Batsman**

- Name: Opp one
- Bowler: Opp three
- Run Rate: 00.0
- Required Run Rate: 00.0

**Current Bowler**

- Name: Andy Barker
- Bowler: Opp three
- Run Rate: 00.0
- Required Run Rate: 00.0
<table>
<thead>
<tr>
<th>Player</th>
<th>Total</th>
<th>Andy Barker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ian Rolls</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Overs</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Last Man</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>000</td>
<td>00</td>
</tr>
<tr>
<td>How Out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall of Wicket</td>
<td>000</td>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Current: Ian Rolls, Batsman, Facing, Current Bowler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opp three bowler</td>
</tr>
<tr>
<td>Run Rate: 1.5</td>
</tr>
<tr>
<td>Required Run Rate: 0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Player</th>
<th>Total</th>
<th>Andy Barker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ian Rolls</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Overs</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Last Man</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>000</td>
<td>00</td>
</tr>
<tr>
<td>How Out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall of Wicket</td>
<td>000</td>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Current: Andy Barker, Batsman Barker, Facing, Current Bowler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opp three bowler</td>
</tr>
<tr>
<td>Run Rate: 3</td>
</tr>
<tr>
<td>Required Run Rate: 0.0</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Ian Rolling</th>
<th>Total</th>
<th>Andy Barker</th>
<th>Current</th>
<th>Ian Rolling</th>
<th>Batsman Barker</th>
<th>Facing</th>
<th>Current</th>
<th>Opp three</th>
<th>Bowler</th>
<th>Run Rate</th>
<th>Required</th>
<th>Run Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>15</td>
<td>10</td>
<td></td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.5</td>
<td>00.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>000</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ian Rolling</td>
<td>Total</td>
<td>Andy Barker</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>-------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overs</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last Man Name</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>000</td>
<td>0</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How Out</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall of Wicket</td>
<td></td>
<td>000</td>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Run Rate: 10.5
- Required Run Rate: 0.00

Extra Scored:

- How many runs were scored from the bye?
  - Options: 0, 2, 4, 6

Cancel | Ok
<table>
<thead>
<tr>
<th>Overs</th>
<th>Last Man Name</th>
<th>Total</th>
<th>Andy Barker</th>
<th>Wickets</th>
<th>Extras</th>
<th>Last Innings</th>
<th>Run Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Ian Rolling</td>
<td>43</td>
<td>10</td>
<td>0</td>
<td>22</td>
<td>000</td>
<td></td>
</tr>
</tbody>
</table>

Current Bat: Ian Rolling
Facing: Opp two
Current Bowler: 10.5
Required Run Rate: 0.0
<table>
<thead>
<tr>
<th>Name</th>
<th>Total</th>
<th>For</th>
<th>Last Man</th>
<th>Last Innings</th>
<th>Wickets</th>
<th>Extras</th>
<th>Name</th>
<th>Overs</th>
<th>Extras</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ian Rolling</td>
<td>11</td>
<td>44</td>
<td>0</td>
<td>0</td>
<td>000</td>
<td>23</td>
<td>Andy Barker</td>
<td>10</td>
<td>23</td>
</tr>
</tbody>
</table>

Current Ian Rolling
Batsman Barker
Facing
Opp two Bowler

Run Rate 11.5
<table>
<thead>
<tr>
<th></th>
<th>Colin Wadey</th>
<th>Andy Barker</th>
<th></th>
<th>Colin Barker</th>
<th>Andy Barker</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total</strong></td>
<td>61</td>
<td>20</td>
<td><strong>For</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Overs</strong></td>
<td>4</td>
<td></td>
<td><strong>Last Innings</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Wickets</strong></td>
<td>1</td>
<td></td>
<td><strong>Extras</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Last Man</strong></td>
<td>Ian Rolling</td>
<td></td>
<td><strong>Run Rate</strong></td>
<td>15.2</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>11</td>
<td>30</td>
<td><strong>Required Run Rate</strong></td>
<td>00.0</td>
<td></td>
</tr>
<tr>
<td><strong>How Out</strong></td>
<td>Bowled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Fall of Wicket</strong></td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Graham Clark</td>
<td>64</td>
<td>Andy Barker</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>How Out</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Fall of Wicket</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Graham Clark</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>How Out</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Fall of Wicket</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The innings has finished, click OK to start the next innings.
### Batting Scorecard

<table>
<thead>
<tr>
<th>Player</th>
<th>Innings</th>
<th>Runs</th>
<th>Balls Faced</th>
<th>dismissal</th>
<th>Partner</th>
<th>Team</th>
<th>Opponent</th>
<th>Venue</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>50</td>
<td>100</td>
<td>Out</td>
<td>B</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>2023</td>
</tr>
</tbody>
</table>

### Bowling Scorecard

<table>
<thead>
<tr>
<th>Bowler</th>
<th>Ovrs</th>
<th>Wkt</th>
<th>Ma</th>
<th>Cc</th>
<th>St</th>
<th>Dismissal</th>
<th>Partner</th>
<th>Team</th>
<th>Opponent</th>
<th>Venue</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>Out</td>
<td>B</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>2023</td>
</tr>
</tbody>
</table>
User Testing

After testing the system myself, I asked my users to test the system. Firstly, I created a test the system. On the next page, I asked my users to fill in upon completing the testing.
<table>
<thead>
<tr>
<th>#</th>
<th>Question</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Do all the buttons/selection lists/tick boxes/option buttons work?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Are all buttons in the right place?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Are the on screen prompts/statements useful, correct and sufficient?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Does the system provide enough on screen information?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Is the colour scheme appropriate?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Is it clear how to enter information for any ball?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Is there evidence of validation (Error messages for incorrect data entry)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Are the error messages clear and useful?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Is it possible to enter invalid data anywhere?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>For the data that is stored, is it represented correctly and clearly?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Can stored data be retrieved?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Is the on screen output correct and easy to view?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Are the Spreadsheets easy to access, do they contain the correct data?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Does the application meet all the requirements of the initial specification?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I also gave them an extra sheet to make additional comments/bugs found in the system. The first test the system gave the following questionnaire, with additional comments.
### User Questionnaire

<table>
<thead>
<tr>
<th></th>
<th>Question</th>
<th>Yes</th>
<th>No</th>
<th>N/A</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Do all the buttons/selection lists/tick boxes/option buttons work?</td>
<td>✔</td>
<td></td>
<td></td>
<td>See Comments</td>
</tr>
<tr>
<td>2</td>
<td>Are all buttons in the right place?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Are the on screen prompts/statements useful, correct and sufficient?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Does the system provide enough on screen information?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Is the colour scheme appropriate?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Is it clear how to enter information for any ball?</td>
<td>✔</td>
<td></td>
<td></td>
<td>See Comments</td>
</tr>
<tr>
<td>7</td>
<td>Is there evidence of validation (Error messages for incorrect data entry)</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Are the error messages clear and useful?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Is it possible to enter invalid data anywhere?</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>For the data that is stored, is it represented correctly and clearly?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Can stored data be retrieved?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Is the on screen output correct and easy to view?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Are the Spreadsheets easy to access, do they contain the correct data?</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Does the application meet all the requirements of the initial specification?</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Signature:** [Signature Image]
These were the comments made:

1. The Penalty button should say penalty
2. The run rate should be updated every ball
3. It isn't possible to score more than 6 (whilst in reality there is the possibility of 3 runs then 4
   overthrows, leading to 7 runs scored off a single ball)
4. Pressing the Prolong over button lead to all potential bowlers being listed twice
5. Short run pressed twice caused the program to crash
6. When batsman crossed is pressed in a wicket, the incorrect batsman is displayed
7. On certain occasions when a bye is called the box for "Click here if the runs are byes" appears
8. A bye, followed by a short run leads to runs being deducted off the batsman
9. Runs scored off extras don't switch the batsmen's ends
10. Retirement gives the bowler and batsman a wicket!
11. If an over is ended early with the bowler bowling no balls, the over isn't counted in the bowlers
    figures
12. When the innings switches the back button should be removed, as it allows the changing of settings

I took each comment individually and took necessary action:

1. Fixed, changed the text on the Penalty button

2. Fixed, the run rate will now update every ball, this was achieved simply by changing a line of code:

   \[
   \text{lblRunRate} = \frac{\text{lblTotal}}{(\text{lblOvers} + 1)}
   \]

   \[
   \text{lblRunRate} = \frac{\text{lblTotal}}{(\text{lblOvers} + (\text{Bowled} / 6))}
   \]

   This means the run rate is an accurate representation of how many runs are being scored per over at the
   current rate.

3. Although this is a problem, I have discussed with my user and have agreed that 7 runs is such a
   rare occurrence that this feature is not needed in the program. Equally it is possible to get around it
   (using 2 balls to represent 1 ball, then prolonging the over)

4. Fixed, this simply needed the following line of code added in the module when the Prolong button is
   pressed:

   \[
   \text{cmdNextPlayer.Clear}
   \]

5. Fixed, this crash was caused through the file not being closed after a short run was pressed, this
   was fixed by adding the following line to the Short run module:

   \[
   \text{Close #1}
   \]

6. Fixed, this was caused by a coding error, the following code was changed:

   If Wicket.chkBatcrossed.Value = True Then
   Was changed to:
   If Wicket.chkBatcrossed.Value = 1 Then
7. Fixed, this was caused by the visible property of the check box not being set to false for any extra other than a no ball. The following line of code was added to all the other extra modules:

```
Extras.chkextrabyes.Visible = False
```

8. Fixed, this was not initially included and was missed out in the design. The code was changed as follows in the short run module:

```
If lbiBatsmanAText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
    lbiBatsmanA.Caption = lbiBatsmanA - 1
Else
    lbiBatsmanB.Caption = lbiBatsmanB - 1
End If
```

Was changed to:

```
If lbiBatsmanAText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
    lbiBatsmanA.Caption = lbiBatsmanA - 1
ElseIf lbiBatsmanBText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
    lbiBatsmanB.Caption = lbiBatsmanB - 1
Else
    lbiTotalExtras.Caption = lbiTotalExtras - 1
End If
```

This just meant that if an extra was scored, a run is deducted from the extras total rather than either batsman.

9. Fixed, this was a problem caused by using a wrong variable i, this was fixed by simply placing in the correct variable.

```
If i = 1 Or i = 3 Or i = 5 Then
```

Was changed to:

```
If k = 1 Or k = 3 Or k = 5 Then
```

10. Fixed, this was done by adding an If statement to the code that adds 1 to the total wickets, and adding retired as a wicket that doesn't count for the bowler:

```
If WicketType <> "Retired" Then
    Scoreboard.lblWickets.Caption = Scoreboard.lblWickets + 1
End If
```

```
If PlayerMatchData.MatchPlayerName = CurBowl And optWicket(3).Value = False And optWicket(4).Value = False And optWicket(10).Value = False Then
    PlayerMatchData.Bowlingwks = PlayerMatchData.Bowlingwks + 1
End If
```

11. Although this is the case, having discussed with the user this issue isn’t necessarily a problem as technically the over hadn’t even started if no balls had been bowled.

12. This issue has been fixed by adding the following code to the EndInnings module:

```
StartPlayers.cmdselectback.Enabled = False
```
New Test Plan

Having dealt with all of the errors that have come up, I have devised a new test plan to test those corrections

[Start Game]
1. Use test scenario to start the game
2. Select BGS 1 and BGS 2 to bat first. Opp one to bowl first
3. [Start Game]
4. Check the Penalty button displays the word penalty
5. Over 1: dot, two, three, one, dot, dot
6. [Prolong Over]
7. Dot
8. Select opp two to bowl the next over, checking the bowling team isn’t listed twice
9. Three, dot dot
10. Check run rate
11. [Short run] [Short run]
12. Wicket (Caught batsman cross, BGS 3 comes in to bat next)
13. No ball(0), Wido(0), Bye(2)
14. [Short Run]
15. Bye(3)
16. Wicket(Retired, BGS 4 comes in to bat next)
17. Use 9 wickets test and then Wicket(Bowled) to end the innings.

Test Detail

<table>
<thead>
<tr>
<th>Test #</th>
<th>Test name</th>
<th>Input Data</th>
<th>Expected Result</th>
<th>Output (SS = Screenshot)</th>
<th>Actual Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check the penalty button</td>
<td>Starting the game</td>
<td>The penalty button has penalty on it</td>
<td>SS 1</td>
<td>As expected</td>
</tr>
<tr>
<td>2</td>
<td>Prolong over</td>
<td>Prolong over button pressed</td>
<td>When the select bower window comes up again, the team is only isled once</td>
<td>SS 2</td>
<td>As expected</td>
</tr>
<tr>
<td>3</td>
<td>Checking run rate</td>
<td>Second over: three</td>
<td>The run rate should be 9 / 1.5 = 6</td>
<td>SS 3</td>
<td>As expected</td>
</tr>
<tr>
<td>4</td>
<td>Double clicking short run</td>
<td>Press short run twice</td>
<td>Two runs should be deducted from the total and from BGS 1</td>
<td>SS 4</td>
<td>As expected</td>
</tr>
<tr>
<td>5</td>
<td>Wicket caught where batsman cross</td>
<td>Press wicket and tick the batsman cross box</td>
<td>BGS 1 should be facing for the next ball</td>
<td>SS 5</td>
<td>As expected</td>
</tr>
<tr>
<td>6</td>
<td>Checking if the extra tick box appears</td>
<td>Press No ball, then wide, then bye</td>
<td>On the wide and bye windows, the option for the runs to be byes should not appear</td>
<td>SS 6</td>
<td>As expected</td>
</tr>
<tr>
<td>7</td>
<td>Short run after a bye</td>
<td>Press bye(2) then short run</td>
<td>The sort run should be deducted from extras not from the batsman</td>
<td>SS 7</td>
<td>As expected</td>
</tr>
<tr>
<td>8</td>
<td>Batsman swap ends</td>
<td>Bye(3) pressed</td>
<td>The facing batsman should change to the other</td>
<td>SS 8</td>
<td>As expected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>9</td>
<td>Retirement shouldn't give a wicket</td>
<td>Wicket(Retired) selected</td>
<td>The total wickets shouldn't increase</td>
<td>SS:9</td>
<td>As expected</td>
</tr>
<tr>
<td>10</td>
<td>End innings should remove back button</td>
<td>End the innings through the test 9 wickets button</td>
<td>When the user reaches the startplayers window, the back button should not appear</td>
<td>SS:10</td>
<td>As expected</td>
</tr>
</tbody>
</table>

---

**Select Player**

Select the next Bowler

---

**Run Rate**

6

---

<table>
<thead>
<tr>
<th>BGS 0</th>
<th>Total</th>
<th>BGS 1</th>
<th>BGS 0</th>
<th>Total</th>
<th>BGS 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>
Second User Testing

After Mr. X tested the system, I then asked Mr. Y to do the same test and gave him the same questionnaire.

His test picked up one major bug, and that was that despite being created, the master file never had any statistics in it.

Looking into this I found the reason for this was because he was testing using my test scenario, and my test scenario didn’t write the used players to the master file, therefore their statistics never appeared on the master file. Having tested the system myself I am now confident that when used normally this bug will not occur.
With reference to the computerised system you have developed, I would like to take this opportunity to thank you for the time and effort you have put into the system and wish you well with your A level.

The requirements laid out for the system were as follows:

**Overview**

A system that allows the user to enter the actions of a cricket match easily and reliably, the user can then easily see the current score and all other information relevant to the game as it progresses. After a game has finished the system will summarise all the data from the game and generate updated statistics for the players that played in the game. So that it can be easily copied onto the intranet site.

The system will also contain a database with all the teams that have played a match scored by this system, including details of all the players that played. The system will be able to create a set of statistics for these players upon request.

**Output requirements**

The system should be able to show all the relevant information about a cricket match as the match is being played, this will include the total number of runs, total number of wickets, overs bowled, each batsman’s current total, last innings score, current run rate, required run rate, extras, last batsman’s score, along with several other options to see overall match statistics and bowling figures.

Once the match has finished the system should be able to create a set of summary statistics on a spreadsheet so that it can be copied easily onto the intranet. This spreadsheet should also be in a printable form so that hard copies can also be created.

At any time the system should be able to generate a set of statistics for any player on screen.

**Input requirements**

The main input will be at the start of the match; name of each player, the date of the match, the name of the opposing team, the number of overs that

**Processing requirements**

When any input is given after a ball the system will update the scoreboard to reflect what happened that ball, as the objective of the system is to make scoring an easier task, the majority of outcomes from a ball will require one keystroke or mouse click to input them. To allow this to happen the following processes will occur:

- The system will ask for the names of all the players at the start of the match for at least the RGS staff team, the names of the opposition team can be entered separately when that relevant player comes into play during the game or at the end of the match.
- The system will always know which batsman is facing so that if runs are scored off the bat then the system will add the runs to the correct batsman.
- The system will add runs to the total whenever they are inputted.
- The system will add runs to the correct section of the scoreboard (either batsman or the extras total).
- When a wicket is taken the scoreboard will add that wicket on to the total number of wickets.
- When a wicket is taken the system will ask for the name of the new batsman which can be selected from a list of batsman entered by the user.
- The system will have a separate window with a full scoreboard of the whole match.
- The scorecard will display 2 tables per innings:
  one table containing the batting figures and the other the bowling figures.
- The batting table will show the runs scored by a batsman, and if that batsman is out, the method by which they were out.
- The bowling table will show each bowler which has bowled, with the number of runs they have had scored against them, the number of overs they have bowled, the number of wickets they have taken, and the number of maidens they have bowled.
- At the end of the match the system will produce 4 tables, a batting and bowling table from each innings, along with a summary statement.

On 5/12/2008 and 9/12/2008 you demonstrated the system to us and we are pleased to say this it filled all the criteria required.

On these tests the following faults were found:

1. The Penalty button should say penalty
2. The run rate should be updated every ball
3. It isn’t possible to score more than 6 (whilst in reality there is the possibility of 3 runs or 4 overthrows, leading to 7 runs scored off a single ball)
4. Pressing the Prolong over button lead to all potential bowlers being listed twice
5. Short run pressed twice caused the program to crash
6. When batsman crossed is pressed in a wicket, the incorrect batsman is displayed
7. On certain occasions when a bye is called, the box for “Click here if the runs are byes” appears
8. A bye, followed by a short run leads to runs being deducted off the batsman
9. Runs scored off extras don’t switch the batsmen’s ends
10. Retirement gives the bowler and batsman a wicket
11. If an over is ended early with the bowler bowling no balls, the over isn’t counted in the bowlers figures
12. When the innings switches the back button should be removed, as it allows the changing of settings

Once these faults are corrected we will be happy to accept the system to be implemented in the next cricket season.

Yours sincerely,

[Signatures]

Score

Statistician
Annotated Listings

This is the Extras form and contains:

1. Optextras(0 – 6) Array
2. lbIHowmanyextras
3. chkextrabytes
4. cmdextrasonsok
5. cmdextrascancel

Private Sub cmdextrascancel_Click()
    Me.Hide
End Sub

Private Sub cmdextrasonsok_Click()
    For i = 0 To 6
        If optextras(i).Value = True Then
            Scoreboard.lblRunstotal.Caption = Int(Scoreboard.lblRunstotal.Caption) + i
            Scoreboard.lblTotalExtras.Caption = Int(Scoreboard.lblTotalExtras.Caption) + i
        End If
    Next i
    If k = 1 Then
        Scoreboard.lblRunstotal.Caption = Int(Scoreboard.lblRunstotal.Caption) + Runsperwbh
        Scoreboard.lblTotalExtras.Caption = Int(Scoreboard.lblTotalExtras) + Runsperwbh
    End If
    If k = 1 And chkextrabytes.Value = 0 Then
        i = 1
    Else
        i = 0
    End If
    Call AfterBall
    Me.Hide
End Sub

Private Sub AfterBall()
    Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
    For j = 1 To LOP(1) / Len(PlayerMatchData)
        Get #1, j, PlayerMatchData
        If PlayerMatchData.MatchPlayerName = CurBat Then
            PlayerMatchData.Battingruns = PlayerMatchData.Battingruns + i
        End If
    Next j
End Sub
PlayerMatchData.BattingBallsFaced = PlayerMatchData.BattingBallsFaced + 1
Put #2, 1, PlayerMatchData
End If
Next j
Close #1
Close #2
Ballnum = Ballnum + 1
Extratype = ""
WicketType = ""
If Bowlled = 6 Then
Call PreEndOver
End If
If chkextrabytes.Value = True Then
chkextrabytes.Value = False
End If
End Sub

This is the ManualOverride form and contains:
1. cmdEndOver
2. cmdBatSwap
3. cmdMOcancel

Dim Tempbat As String
Private Sub cmdBatSwap_Click()
    Tempbat = CurBat
    CurBat = Otherbat
    Otherbat = Tempbat
    Scoreboard.lblFacingBatsman.Caption = CurBat
    Me.Hide
End Sub

Private Sub cmdEndOver_Click()
    Call PreEndOver
    Me.Hide
End Sub

Private Sub cmdMOcancel_Click()
    Me.Hide
End Sub

Add 1 to the ball num.
Reset Extra and Wicket variable.
If 6 balls have been bowled the over ends.
Reset the check box.

Saves the batsman when pressed
Calls end over
Hides the window.
### Match Start - Enter teams

#### BGS Staff
- Jim Rolling
- Colin Window
- Andy Keen
- Andy Dancer
- Graham Clark
- Justin Harford
- Ben Scott
- Kevin Blackburn
- Andrew Flintoff
- Kevin Peterson
- Steve Harmason

#### Opposition
- Opp One
- Opp Two
- Opp Three
- Opp Four
- Opp Five
- Opp Six
- Thirty Character Name
- New Opp Eight
- New Opp Nine
- New Opp Ten
- New Opp Eleven

---

```vbnet
Dim Month As String
Dim Numdays As Integer
Dim Filedate As String

Private Sub cmdOpposearch_Click()
    Me.Hide
    Unload Search
    Teamsearch = False
    searchteam = Trim(txtOpposition.Text)
    Search.Show
    End Sub

Private Sub cmdSearchBgs_Click()
    searchteam = "BGS Staff"
```

---

Me.Hide
Teamsearch = False
Unload Search
Search.Show
End Sub

Private Sub cmdStartdone_Click()
Open App.Path & "\Cricketmasterfile.txt" For Random As #1 Len = Len(PLAYERRecord)
For i = 0 To 10
If txtBGSPlayer(i).Enabled = True Then
    With PLAYERRecord
        lastrecord = LOF(1) / Len(PLAYERRecord) + 1
        .PLAYERID = lastrecord
        .PLAYERName = txtBGSPlayer(i)
        .PLAYERTeam = "BGS Staff"
        .Innings = 0
        .AverageRuns = 0
        .Highscore = 0
        .NotOuts = 0
        .Overs = 0
        .Runs = 0
        .Wickets = 0
        .Maidens = 0
        .econrate = 0
        .sinkerate = 0
        .bowlingruns = 0
    End With
    Put #1, lastrecord, PLAYERRecord
End If
Next i
For i = 0 To 10
    If txtOppPlayer(i).Enabled = True Then
        With PLAYERRecord
            lastrecord = LOF(1) / Len(PLAYERRecord) + 1
            .PLAYERID = lastrecord
            .PLAYERName = txtOppPlayer(i)
            .PLAYERTeam = txtOpposition
            .Innings = 0
            .AverageRuns = 0
            .Highscore = 0
            .NotOuts = 0
            .Overs = 0
            .Runs = 0
            .Wickets = 0
            .Maidens = 0
            .econrate = 0
            .sinkerate = 0
            .bowlingruns = 0
        End With
        Put #1, lastrecord, PLAYERRecord
    End If
Next i
Close #1
Open App.Path & "\ScoreFiles" & Trim(txtOpposition.Text) & cmbDD.Text & cmbMM.Text &
& cmbYY.Text & ".txt" For Random As #1 Len = Len(PLAYERMatchData)
For i = 0 To 10
    PlayerMatchData = PlayerMatchData & PLAYERMatchData(i)
Next i
Close #1
End Sub

and opens the search window.

Creates any new players within the master file. Setting all their variables to 0.

The same process is repeated for the opposing team.

Creates the summary file.
With PlayerMatchData
  .MatchPlayerID = i + 1
  .MatchPlayerName = txtBGSplayer(i).Text
  .MatchPlayerTeam = "BGS Staff"
  .Battingruns = 0
  .BattingBallsFaced = 0
  .BattingHowOut = ""
  .BattingWktBowler = ""
  .Bowlingovers = 0
  .bowlingruns = 0
  .Bowlingwickets = 0
  .Bowlingmades = 0
End With
Put #1, i + 1, PlayerMatchData
Next i
For i = 0 To 10
  With PlayerMatchData
    .MatchPlayerID = i + 12
    .MatchPlayerName = txtOppoplayer(i).Text
    .MatchPlayerTeam = txtOpposition.Text
    .Battingruns = 0
    .BattingBallsFaced = 0
    .BattingHowOut = ""
    .BattingWktBowler = ""
    .Bowlingovers = 0
    .bowlingruns = 0
    .Bowlingwickets = 0
    .Bowlingmades = 0
  End With
  Put #1, i + 12, PlayerMatchData
Next i
Close #1
OpposingTeam = txtOpposition.Text
Me. Hide
Settings.Show
End Sub
Private Sub cmdTeamsearch_Click()  
Me. Hide
Teamsearch = True
Unload Search
Search Show
End Sub
Private Sub cmbMM_Click()
  cmbDD.Enabled = True
  cmbDD.Clear
  cmbDD.Text = "Day"
  Month = cmbMM.Text
  Select Case Month
    Case "January", "March", "May", "July", "August", "October", "December"
      Numdays = 31
    Case "April", "June", "September", "November"
      Numdays = 30
    Case "February"
      Numdays = 28
  End Select
For i = 1 To NumDays
    cmbDD.AddItem i
Next i
thisyear = Year(Date)
For i = 2007 To thisyear
    cmbYY.AddItem i
Next i
End Sub

Private Sub Form_Load()
Unload Scoreboard
End Sub

This is the NextPlayer form and contains:
1. lblNextPlayerText
2. cmbNextPlayer
3. cmdNBok
4. cmdProlong

Private Sub cmdNBok_Click()
If j = 0 Then
    Me.Hide
    Call EndOver2
ElseIf j = 1 Then
    Me.Hide
    Call NewBatsman
ElseIf j = 2 Then
    Me.Hide
    Call NewBatsman
Unload RunOut
End If
End Sub

Private Sub EndOver2()
If cmbNextPlayer.Text = "" Then
    MsgBox "Please Select a Player"
Else
    TempBowl = CurBowl
    CurBowl = NextPlayer cmbNextPlayer.Text
    Scoreboard.lblCurrentBowler.Caption = CurBowl
    cmbNextPlayer.Clear
    Close #1
    Scoreboard.lblOvBowler.Caption = Scoreboard.lblOvBowler + 1
    Bowled = 0
End If
Call EndOver

Then, set the date for your from 2007 to the current year.
Unload the scoreboard

Select the next Bowler:
Score a-player
OK
Prolong over

Calls a subroutine depending whether it's a next bowler or batsman.

Validates to check a player is selected.
Sets the CurBowl to the new bowler.
Clears the nextplayer list.
Adds 1 to the overs bowled.
Then hides the window

Replaces the batsman out with the new batsman

If the batsman cross the batsman swap.

Updates the scoreboard

Sicks the next batsman into the batting order.

Casts the end of the over is 6 balls have been concluded.

Hide the window to allow another bowl to be bowl.

This is the Run Out form and contains:

1. optBatA
2. optBatB
3. cmdRunOutDone
4. cmdRunOutBack
5. cmdRunsScored

Select which batsman was out and how many runs were scored before the run out occurred.

<table>
<thead>
<tr>
<th>Opp five</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Character Name</td>
<td>Done</td>
</tr>
</tbody>
</table>
Private Sub cmdRunOut_Click()
    Me.Hide
    End Sub

Private Sub cmdRun OCIe_Click()
    Close #1
    Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
    Scoreboard.lblLastManHowOut = "Run Out"
    If optBatA.Value = True Then
        BatOut = BatA
    ElseIf optBatB.Value = True Then
        BatOut = BatB
    Else
        MsgBox "Select the batsman who is out" Update the total runs.
    End If
    j = Int(Scoreboard.lblRunstoTotal) Scoreboard.lblRunstoTotal = j + Int(cmbrunsscored.Text)
    Call EndBall(inf(cmbrunsscored.Text))
    Close #1
    Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
    If BatOut = BatA Then
        Scoreboard.lblLastManTotal = Scoreboard.lblBatsmanA.Caption
        Scoreboard.lblBatsmanA.Caption = "0"
    Else
        Scoreboard.lblLastManTotal = Scoreboard.lblBatsmanB.Caption
        Scoreboard.lblBatsmanB.Caption = "0"
    End If
    Scoreboard.lblWickets.Caption = Scoreboard.lblWickets + 1
    Scoreboard.lblLastManName = CurBat
    Scoreboard.lblLastManHow = Scoreboard.lblRunstoTotal
    For i = 1 To 22
        Get #1, i, PlayerMatchData
        If PlayerMatchData.MatchPlayerName = BatOut Then
            PlayerMatchData.BattingWktBowler = CurBowl
            Put #1, i, PlayerMatchData
        End If
    Next i
    For i = 1 To LOF(1) / Len(PlayerMatchData)
        Get #1, i, PlayerMatchData
        If PlayerMatchData.MatchPlayerName <> Trim(Otherbat) And
            Trim(PlayerMatchData.MatchPlayerName) <> Trim(CurBat) And Trim(PlayerMatchData.BattingHowOut) <> "" And Trim(PlayerMatchData.MatchPlayerTeam) = Trim(BattingTeam) Then
            NextPlayer.cmdNextPlayer.AddItem PlayerMatchData.MatchPlayerName
        End If
    Next i
    Me.Hide
    NextPlayer.lblNextPlayer.Caption = "Select the next batsman" j = 2
    Set the next player form and opens it.

Hints the window.
Sets the batsman who is out, who validates that a batsman has been selected.

Update the total runs.
 Opens the summary file then updates the last man information.

Loops through the summary file and updates the wicket information for that batsman.
Then loops through creating a list for the next batsman.

Sets the next player form and opens it.
NextPlayer.Show
Close #1
End Sub

Private Sub Form_Load()
optBatA.Caption = BatA
optBatB.Caption = BatB
End Sub

Sets the radio button text
This is the Scoreboard form, the contents of which are shown below.

The scoreboard contains the following labels which are changed/command buttons:
1. lblBatsmanAText
2. lblBatsmanA
3. lblunstotal
4. lblBatsmanB
5. lblBatsmanBText
6. lblOversBowled
7. lblWickets
8. lblLastInnings
9. lblTotalExtras
10. lblLastManName
11. lblLastManTotal
12. lblLastManHowOut
13. lblLastManFow
14. cmdManOverride
15. lblFacingBatsman
16. lblCurrentBowler
17. lblRunRate
18. lblRqsRunRate
19. cmdScoreboardShow
20. cmdrun(0)
21. cmdrun(1)
22. cmdrun(2)
23. cmdrun(3)
24. cmdrun(4)
25. cmdWicket
26. cmdBye
27. cmdLegBye
28. cmdrun(5)
29. cmdrun(6)
30. cmdWide
31. cmdNoBall
32. cmdUndo
33. cmdShortRun
34. cmdPenalties
35. cmdStartGame

Dim Totalruns As Integer
Dim TotalWkts As Integer
Dim OversBowled As Integer
Dim LastInns As Integer
Dim TotalExtras As Integer
Dim LmiRuns As Integer
Dim LmiNum As Integer

Private Sub cmdBye_Click()
    Extras.Show
Extratype = "Bye"
Extra.chckextrabytes.Visible = False
k = 0
Extra.lblHowmanyextras.Caption = "How many runs were scored from"
Extra.optextrax(0).Enabled = False
End Sub

Private Sub cmdLegBye_Click()
Extra.Show
Extratype = "Leg Bye"
Extra.chckextrabytes.Visible = False
k = 0
Extra.lblHowmanyextras.Caption = "How many runs were scored from"
Extra.optextrax(0).Enabled = False
End Sub

Private Sub cmdManOverRide_Click()
ManualOverride.Show
End Sub

Private Sub cmdNoBall_Click()
Extra.Show
Extratype = "No Ball"
Extra.chckextrabytes.Visible = True
k = 1
Extra.lblHowmanyextras.Caption = "How many runs were scored from"
Extra.optextrax(0).Enabled = True
End Sub

Private Sub cmdPenalties_Click()
Extra.Show
Extratype = "Penalty"
Extra.chckextrabytes.Visible = False
k = 0
Extra.lblHowmanyextras.Caption = "How many runs were scored from"
Extra.optextrax(0).Enabled = True
End Sub

Private Sub cmdrun_Click(Index As Integer)
Call EndBall(Index)
Call AfterBall(Index)
If Index <> 0 Then
  Maiden = False
End If
End Sub

Private Sub cmdrun_KeyPress(Index As Integer, KeyAscii As Integer)
If KeyAscii > 47 And KeyAscii < 56 Then
  Call cmdrun_Click(KeyAscii - 48)
ElseIf KeyAscii = 100 Then
  Call cmdrun_Click(0)
End If
End Sub

Private Sub cmdScorecardShow_Click()
If lblOversBowled.Caption = 0 Then
MsgBox "The scorecard cannot be shown until a full over has been completed"
Else
Scorecard.Show
End If
End Sub

Private Sub cmdShow_Click()
Close #1
Open StatisticPath For Random As #1 Len = Len(MatchData)
i = LDB(1) / Len(MatchData)
For #1, i, MatchData
MatchData.Runs = MatchData.Runs - 1
If lbBatsmanAText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
lbBatsmanA.Caption = lbBatsmanA - 1
ElseIf lbBatsmanBText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
lbBatsmanB.Caption = lbBatsmanB - 1
Else
lbTotalExtras.Caption = lbTotalExtras - 1
End If
lbRunstotal.Caption = lbRunstotal - 1
End If
Put #1, i, MatchData
End Sub

Private Sub cmdStartGame_Click()
cmdStartGame.Visible = False
lbFacingBatsman = OpeningBatsman
lbCurrentBowler = OpeningBowl
CurBowl = OpeningBowl
CurBat = OpeningBatsman
Otherbat = OpeningNonfaceBat
lbBatsmanAText.Caption = OpeningBatsman
lbBatsmanBText.Caption = OpeningNonfaceBat
lbBatsmanAText.Alignment = 2
lbBatsmanBText.Alignment = 2
BatA = OpeningBatsman
BatB = OpeningNonfaceBat
Battingorder(0) = OpeningBatsman
Battingorder(1) = OpeningNonfaceBat
cmdrun(0).SelFocus
Mainen = True
End Sub

Private Sub cmdUndo_Click()
Close #1
Open StatisticPath For Random As #1 Len = Len(MatchData)
i = LDB(1) / Len(MatchData)
For #1, i, MatchData
lbRunsTotal = lbRunsTotal - MatchData.Runs
If MatchData.Runs = 1 Or MatchData.Runs = 3 Or MatchData.Runs = 5 Then
TempBat = CurBat
CurBat = Otherbat
Otherbat = TempBat
End If
If Trim(MatchData.Extra) <> "Wide" And Trim(MatchData.Extra) <> "NoBall" Or ExtraBall = False Then
Bowed = Bowled - 1
End If
"Validates that the scorecard can be shown, and clears it."
"Takes away one run from the batsman and the total for a short run.
"This sets up the scoreboard at the start of a new game or innings, using variables set in the settings or start-players form.
"When undo is pressed the system looks at what happened to previous ball and resets the system to that ball."
If Bowled = -1 Then
    Get #1, i, MatchData
    Scoreboard.lblCurrentBowler.Caption = MatchData.Bowler
    Scoreboard.lblBowled.Caption = "Bowled"
    Scoreboard.lblTempBat.Caption = "TempBat"
    Scoreboard.lblOtherBat.Caption = "OtherBat"
End If

If Trim(MatchData.Wicket) <> "" Then
    Scoreboard.lblWickets.Caption = Scoreboard.lblWickets.Caption = "Wickets"
    Scoreboard.lblLastManName.Caption = LastManName
    Scoreboard.lblLastManTotal.Caption = LastManTotal
    Scoreboard.lblLastManBowled.Caption = LastManBowled
    Scoreboard.lblLastManHowOut.Caption = LastManHowOut
End If

If i = 1 Then
    Get #1, i, MatchData
End If

If lblBatsmanA.Caption = CurBat Then
    lblBatsmanA.Caption = lblBatsmanA.Caption = "Batsman"
Else
    lblBatsmanB.Caption = lblBatsmanB.Caption = "Batsman"
End If

With MatchData
    Batnumber = 0
    Batsman = ""
    BatsmanOut = 0
    Bowler = ""
    Extra = ""
    Runs = 0
    Wicket = ""
End With

Put #1, i, MatchData
Close #1
End Sub

Private Sub cmdWicket_Click()
    LastAction = "Wicket"
    Wicket.Show
End Sub

Private Sub cmdWide_Click()
    LastAction = "Wide"
    Extras.Show
    Extras.Extratype = "Wide"
    Extras.chkExtraB.Caption = "How many extras were score from this wide?"
    Extras.chkExtraB.Enabled = True
End Sub

This was a scenario text for updating the site and spreadsheet.

Private Sub Command1_Click()
    Call UpdateFile
End Sub

Private Sub Form_Load()
Private Sub AfterBall(i As Integer)
    lblRuntotal = lblRuntotal + i
    Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
    For j = 1 To LOF(1) / Len(PlayerMatchData)
        Get #1, j, PlayerMatchData
        If PlayerMatchData.MatchPlayerName = CurBat Then
            If PlayerMatchData.Battingruns = PlayerMatchData.Battingruns + i Then
                End If
                If PlayerMatchData.MatchPlayerName = CurBowl Then
                    PlayerMatchData.bowlingruns = PlayerMatchData.bowlingruns + i
                End If
            End If
            Put #1, j, PlayerMatchData
        Next j
    Close #1
    If CurBat = BatA Then
        lblBatsmanA.Caption = lblBatsmanA.Caption + i
    Else
        lblBatsmanB.Caption = lblBatsmanB.Caption + i
    End If
    If Bowled = Bowled + 1 Then
        If i = 1 Or i = 3 Or i = 5 Then
            Tempbat = CurBat
            CurBat = Otherbat
            Otherbat = Tempbat
        End If
    End If
    If lblFacingBatsman.Caption = CurBat Then
        If lblOversBowed <= 0 Then
            If lblLastInnings <> 0 Then
                lblRunRate = (lblLastInnings - lblRuntotal) / (limitedOvers - lblOversBowed)
                If lblRunRate < 0 Then
                    If BattingTeam = "BGS Staff" Then
                        k = 1
                    Else
                        k = 2
                    End If
                    Call EndMatch(k)
                End If
            Else
                If Bowled >= 6 Then
                    Call PreEndOver
                End If
            End If
        End If
    End Sub
The scorecard is created from the form load event, and therefore only has the back button

cmdScorecardBack

```
Dim SelectedBowler As String
Dim p As Integer

Private Sub cmdScorecardBack_Click()
    Dim CScorecard As String
    Dim CScorecard As String
    Dim CScorecard As String
    Dim CScorecard As String
    Dim CScorecard As String
```

The scorecard is created from the form load event, and therefore only has the back button.

cmdScorecardBack

```
Dim SelectedBowler As String
Dim p As Integer

Private Sub cmdScorecardBack_Click()
    Dim CScorecard As String
    Dim CScorecard As String
    Dim CScorecard As String
    Dim CScorecard As String
    Dim CScorecard As String
```

```
Private Sub Form_Load()
  I = 0

  Close #1
  Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
  For j = 1 To 22
    Get #1, j, PlayerMatchData
    If PlayerMatchData.Bowlingovers <> 0 And Trim(PlayerMatchData.MatchPlayerTeam) = Trim(BowlingTeam) Then
      I = I + 1
      PlayerMatchData.Listed = False
      Put #1, I, PlayerMatchData
  End If
  Next j
  For i = 0 To 10
    If i <> 0 Then
      Load lblBatsmanList(i)
      Load lblHowOut(i)
      Load lblWicketBowler(i)
      Load lblBattlingRuns(i)
      Load lblBallsFaced(i)
      End If
      lblBatsmanList(i).FontSize = 12
      lblBatsmanList(i).Top = 1320 + 480 * i
      lblBatsmanList(i).Left = 240
      lblBatsmanList(i).Caption = BattingOrder(i)
      lblBatsmanList(i).Visible = True
      lblBatsmanList(i).Width = 2000
      lblHowOut(i).Top = 1320 + 480 * i
      lblHowOut(i).Left = 1690
      lblHowOut(i).Visible = True
      lblHowOut(i).Caption = ""
      lblWicketBowler(i).Top = 1320 + 480 * i
      lblWicketBowler(i).Left = 3600
      lblWicketBowler(i).Visible = True
      lblWicketBowler(i).Caption = ""
      lblBattlingRuns(i).Top = 1320 + 480 * i
      lblBattlingRuns(i).Left = 4320
      lblBattlingRuns(i).Visible = True
      lblBattlingRuns(i).Caption = ""
      lblBallsFaced(i).Top = 1320 + 480 * i
      lblBallsFaced(i).Left = 6640
      lblBallsFaced(i).Visible = True
      lblBallsFaced(i).Caption = ""
      For j = 1 To 22
        Get #1, j, PlayerMatchData
        If PlayerMatchData.MatchPlayerName = BattingOrder(i) Then
          lblBatsmanList(i).Caption = PlayerMatchData.MatchPlayerName
          lblHowOut(i).Caption = PlayerMatchData.BattingHowOut
          lblWicketBowler(i).Caption = PlayerMatchData.BowlingWktBowler
          If Trim(lblWicketBowler(i).Caption) = "" Then
            lblHowOut(i).Caption = "Not"
            lblWicketBowler(i).Caption = "Out"
        End If
      Next j
  End If
End Sub
End If
    lblBatsman(i).Caption = PlayerMatchData.Batsman
End If
End i
Next i
For i = 0 To $ - 1
If i <= 0 Then
    Load lblBowlerlist()
    Load lblOversBowler(i)
    Load lblBowlerRuns(i)
    Load lblBowlerWicket(i)
    Load lblMaidens(i)
End If
lblBowlerlist(i).FontSize = 12
lblBowlerlist(i).Top = 1320 + 480 * (i)
lblBowlerlist(i).Left = 8150
lblBowlerlist(i).Visible = True
lblBowlerlist(i).Caption = """""""""""""
lblOversBowler(i).Width = 2000
_lblOversBowler(i).Top = 1320 + 480 * (i)
_lblOversBowler(i).Left = 9960
_lblOversBowler(i).Visible = True
_lblOversBowler(i).Caption = """"
_lblBowlerRuns(i).Top = 1320 + 480 * (i)
_lblBowlerRuns(i).Left = 10920
_lblBowlerRuns(i).Visible = True
_lblBowlerRuns(i).Caption = """
_lblBowlerWicket(i).Top = 1320 + 480 * (i)
_lblBowlerWicket(i).Left = 12090
_lblBowlerWicket(i).Visible = True
_lblBowlerWicket(i).Caption = """
_lblMaidens(i).Top = 1320 + 480 * (i)
_lblMaidens(i).Left = 13320
_lblMaidens(i).Visible = True
_lblMaidens(i).Caption = """
End If
Get #1, j, PlayerMatchData
If PlayerMatchData.Listed = False And PlayerMatchData.BowlingOvers <> 0 And
Trim(PlayerMatchData.MatchPlayerTeam) = Trim(BowlingTeam) Then
    lblBowlerlist(i).Caption = PlayerMatchData.MatchPlayerName
    lblOversBowler(i).Caption = PlayerMatchData.BowlingOvers
    lblBowlerRuns(i).Caption = PlayerMatchData.BowlingRuns
    lblBowlerWicket(i).Caption = PlayerMatchData.BowlingWickets
    lblMaidens(i).Caption = PlayerMatchData.BowlingMaidens
    PlayerMatchData.Listed = True
    Put #1, j, PlayerMatchData
End If
j = j + 1
Loop Until lblBowlerlist(i).Caption <> """" Or j = 23
Next i
For i = 1 To $ - 1
    Load lblBowlerDataList(i)
    lblBowlerDataList(i).FontSize = 12
    lblBowlerDataList(i).Top = 8400 + i * 600
This creates a table for all the bowlers.
This fills in the information for all the bowlers.
This list all the bowlers in the lower table, with 5 boxes to show the overs they have bowled.

This fills in the boxes using the statistics file.
This is the Search form.  

1. cmdCancelSearch  
2. cmdMoveLeft  
3. cmdMoveRight  
4. cmdSearchDone  
5. lstDataList  
6. lstSelectedList

Dim PlayerRecord As PlayerData  
Dim SelectedPlayer As String  
Dim PlayerNameSelected  
Dim PlayerIDSelected  
Dim Listed As Boolean  
Dim recordNum As Integer  
Dim StartValue As Integer

Private Sub cmdCancelSearch_Click()  
Me.Hide  
MatchStart.Show  
End Sub

Private Sub cmdMoveLeft_Click()  
lstSelectedList.RemoveItem lstSelectedList.ListIndex  
End Sub

Private Sub cmdMoveRight_Click()  
SelectedPlayer = lstDataList.Text
Private Sub cmdSearchDone_Click()
    i = 0
    If TsamSearch Then
        If lstSelectedlist.ListCount > 1 Then
            MsgBox "You have selected more than one team", vbOKOnly, "Error 101"
        Else
            searchteam = lstSelectedlist.List(0)
            MatchStart.TxtOpposition.Text = searchteam
            OpposingTeam = searchteam
            MatchStart.txtOpposition.Enabled = False
            Teamsearch = False
            MatchStart.cmdOpposeSearch.Enabled = True
            End If
        Else
            StartValue = 0
            If Trim(searchteam) <> "BGS Staff" Then
                Do While MatchStart.txtOpponPlayer(StartValue).Enabled = False
                    StartValue = StartValue + 1
                End While
                Loop Until MatchStart.txtOpponPlayer(StartValue).Enabled = True
            Else
                Do While MatchStart.txtBGSPlayer(StartValue).Enabled = False
                    StartValue = StartValue + 1
                End While
                Loop Until MatchStart.txtBGSPlayer(StartValue).Enabled = True
            End If
            If lstSelectedlist.ListCount + StartValue > 11 Then
                MsgBox "You have selected more than 11 players", vbOKOnly, "Error 102"
            Else
                If Trim(searchteam) <> "BGS Staff" Then
                    awayteam(i + StartValue).PlayerID = lstSelectedlist.ItemData(i)
                    awayteam(i + StartValue).PlayerName = lstSelectedlist.List(i)
                Else
                    homeTeam(i + StartValue).PlayerID = lstSelectedlist.ItemData(i)
                    homeTeam(i + StartValue).PlayerName = lstSelectedlist.List(i)
                End If
                i = i + 1
                Loop Until i = lstSelectedlist.ListCount
                Call PlayerFill
            End If
        End If
    MatchStart.Show
    Me.Hide
End Sub

Private Sub Form_Load()
    MatchStart.Hide
    Open App.Path & "/CricketMasterStats.txt" For Random As #1 Len = Len(PlayerRecord)
If Teamsearch Then
    Search.Caption = "Team Search"
    For recordnum = 1 To LOF(PlayerRecord) / Len(PlayerRecord) / Len(PlayerRecord)
        Get #1, recordnum, PlayerRecord
        Listed = False
        If Trim(PlayerRecord.PlayerTeam) <> "BGS Staff" Then
            For i = 0 To lstDataList.ListCount
                If PlayerRecord.PlayerTeam = lstDataList.List(i) Then
                    Listed = True
            Next i
        End If
        Next recordnum
    Else
        Search.Caption = "Player: Search"
        For recordnum = 1 To LOF(PlayerRecord) / Len(PlayerRecord)
            Get #1, recordnum, PlayerRecord
            If Trim(PlayerRecord.PlayerTeam) = Trim(searchTeam) Then
                lstDataList.AddItem PlayerRecord.PlayerTeam & PlayerRecord.PlayerName & PlayerRecord.PlayerID
            End If
            Next recordnum
        End If
    End If
End Sub

Private Sub PlayerFill()
    For i = StartValue To (StartValue + lstSelectedList.ListCount - 1)
        MatchStart.txtOppoplayer(i).Text = awayteam(i).PlayerName & awayteam(i).PlayerID
        MatchStart.txtOppoplayer(i).Enabled = False
    Next i
    SelectedPlayers = lstSelectedList.ListCount - 1
End Sub

On looking the search window, the data list box will generate a list of teams or a list of names for a particular team from the master side.

This puts players into the match start under.
This is the settings window, and contains the following:

1. Radrunsperrwide(0 – 3)
2. chkExtraBall
3. RadBatSelect(0)
4. RadBatSelect(1)
5. txtLimitedOvers
6. cmdSettingsOK
7. cmdSettingsBack

Private Sub cmdsettingsback_Click()
Me.Hide
MatchStart.Show
End Sub

Private Sub cmdsettingsOK_Click()
  i = 0
  LimitedOvers = txtLimitedOvers.Value
  For i = 0 To 3
    If Radrunsperrwide.Value = True Then
      Runsperrwb = i + 1
    End If
  Next i
  If Runsperrwb = 0 Then
    MsgBox "Select a number of runs for a wide/no ball", vbOKOnly
    i = 1
  End If
  If chkExtraball.Value = 1 Then
    Extraball = True
  Else
    Extraball = False
  End If
  If RadBatSelect(0).Value = True Then
  End If

This sets all the variables to the option selected.
OpeningBatteam = "BGS Staff"
ElseIf RadBatselect(1).Value = True Then
    OpeningBatteam = OpposingTeam
Else
    MsgBox "Select a team to opening the batting", vbOKOnly
End If
End If
If j = 0 Then
    Me.Hide
End Sub

Private Sub Form_Load()
    RadBatselect(1).Caption = OpposingTeam
End Sub

This is the StartPlayers form and contains the following:

1. cmbFacingBat
2. cmbNonFacingBat
3. cmbOpeningBowl
4. cmdSelectdone
6. cmdSelectback

Private Sub cmdSelectback_Click()
    Me.Hide
End Sub

This removes the player from the other batsman list when they are selected from the facing batsman list.

Dim deletedItem As String
Dim Itemadded As Boolean

Private Sub cmdFacingBat_Click()
    Itemadded = False
    For j = 0 To 10
        If deletedItem <> "" And Itemadded = False Then
            cmbNonFacingBat.AddItem deletedItem
            Itemadded = True
        End If
        If cmbNonFacingBat.List() = cmbFacingBat.Text Then
            cmbNonFacingBat.RemoveItem j
            deletedItem = cmbFacingBat.Text
        End If
    Next j
End Sub

Validates a batting team has been selected.

When the form is loaded the opposing team is set as an option to open the batting.
Settings.Show
End Sub

Private Sub cmdSelectNone_Click()
If cmbFacingBat.Text = "" And cmbNonFacingBat.Text = "" And cmbOpeningBowl.Text = "" Then
    OpeningBatsman = cmbFacingBat.Text
    OpeningNonFacing = cmbNonFacingBat.Text
    OpeningBowl = cmbOpeningBowl.Text
    j = 1
    Scoreboard.cmdStartGame.Visible = True
    Me.Hide
    Scoreboard.Show
    Scoreboard.SetFocus
Else
    MsgBox "You have not selected one or more of the option(s)"
End If
End Sub

Private Sub Form_Load()
SummaryPath = App.Path & \"\Summary\" & Trim(MatchStart.txtOpposition.Text) & \"\MatchStart.txtMM.Text & MatchStart.txtYY.Text & "lt"
Open SummaryPath For Random As #1 Len = Len(PlayersMatchData)
If OpeningBatsman = "BGS Staff" Then
    BattingTeam = "BGS Staff"
    BowlingTeam = OpposingTeam
    For i = 1 To 11
        Get #1, i, PlayerMatchData
        cmbFacingBat.AddItem PlayerMatchData.MatchName
        cmbNonFacingBat.AddItem PlayerMatchData.MatchName
        Get #1, i + 11, PlayerMatchData
        cmbOpeningBowl.AddItem PlayerMatchData.MatchName
    Next i
ElseIf OpeningBatsman = OpposingTeam Then
    BattingTeam = OpposingTeam
    BowlingTeam = "BGS Staff"
    For i = 1 To 11
        Get #1, i + 11, PlayerMatchData
        cmbFacingBat.AddItem PlayerMatchData.MatchName
        cmbNonFacingBat.AddItem PlayerMatchData.MatchName
        Get #1, i, PlayerMatchData
        cmbOpeningBowl.AddItem PlayerMatchData.MatchName
    Next i
End If
Close #1
If i = 1 Then
    cmdSelectback.Enabled = False
End If
End Sub

This is the StartGame window and contains just one command button:

cmdStart

Press Start Game to start a new game.

When the game loads a list is generated from the Summary file to create the combo box lists.
Private Sub cmdStartStart_Click()
    Me.Hide
    MatchStart.Show
End Sub

This is the Wicket form and contains the following:
1. optWicket(0 – 10)
2. chkBatCrossed
3. cmdWicketDone
4. cmdWicketCancel

Private Sub cmdWicketCancel_Click()
    Me.Hide
End Sub

Private Sub cmdWicketDone_Click()
    Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
    If optWicket(4).Value = True Then
        RunOut.Show
    Else
        Hide the window
    End If
LastBatName = Scoreboard.IblLastManName
LastBatFow = Scoreboard.IblLastManFow
LastBatTotal = Scoreboard.IblLastManTotal
LastBatHowOut = Scoreboard.IblLastManHowOut
BatOut = CurBat

For i = 0 To 10
   If optWicket(i).Value = True Then
      Scoreboard.IblLastManHowOut = optWicket(i).Caption
      WicketType = optWicket(i).Caption
   End If
Next i

If BatOut = BatA Then
   Scoreboard.IblLastManTotal = Scoreboard.IblBatsmanA.Caption
   Scoreboard.IblBatsmanA.Caption = "0"
Else
   Scoreboard.IblLastManTotal = Scoreboard.IblBatsmanB.Caption
   Scoreboard.IblBatsmanB.Caption = "0"
End If

If WicketType <> "Retired" Then
   Scoreboard.IblWickets.Caption = Scoreboard.IblWickets + 1
End If

Scoreboard.IblLastManName = CurBat
Scoreboard.IblLastManFow = Scoreboard.IblRunTotal

If Wicket.chBatsmanCrossed.Value = 1 Then
   If optWicket(0).Value = False Then
      MsgBox "The batsman cannot cross unless the wicket was caught"
   Else
      CurBat = OtherBat
   End If
End If

For i = 1 To LOF(PlayerMatchData)
   Get #1, i, PlayerMatchData
   If PlayerMatchData.MatchPlayerName = BatOut Then
      PlayerMatchData.BattingWktBowler = CurBowl
   End If
   If PlayerMatchData.MatchPlayerName <> CurBowl And optWicket(3).Value = False And optWicket(4).Value = False And optWicket(10).Value = False Then
      PlayerMatchData.Bowlingwkt = PlayerMatchData.Bowlingwkt + 1
   End If
   Put #1, i, PlayerMatchData
Next i

For i = 1 To LOF(PlayerMatchData)
   Get #1, i, PlayerMatchData
   If PlayerMatchData.MatchPlayerName <> OtherBat And PlayerMatchData.MatchPlayerName <> CurBat And Trn(PlayerMatchData.BattingHowOut) = "" And Trn(PlayerMatchData.MatchPlayerName) = Trn(BattingTeam) Then
      NextPlayer.cmbNextPlayer.AddItem PlayerMatchData.MatchPlayerName
   End If
Next i

Call EndBat(0)
If Scoreboard.IblWickets = 10 And Secondinnings = False Then
   Call EndInnings
ElseIf Scoreboard.IblWickets = 10 And Secondinnings = True Then
   If BowlingTeam = "BGS Staff" Then
      Generate the next player list.
      Call end innings or end match if necessary.
k = 1
Else
  k = 2
End If
Call EndMatch(k)
Else
  NextPlayer.lblNextPlayerText.Caption = "Select the next batsman"
  NextPlayer.cmdProlong.Visible = False
  j = 1
  NextPlayer.Show
End If
Me.Hide
End Sub

In addition to the forms, there is also the Module:

Public Type PlayerData
  PlayerID As Integer
  PlayerName As String * 30
  PlayerTeam As String * 20
  Innings As Integer
  AverageRuns As Integer
  Highscore As Integer
  NotOuts As Integer
  Overs As Integer
  Runs As Integer
  Wickets As Integer
  Maidens As Integer
  econrate As Single
  strikerate As Single
  bowlingruns As Integer
End Type

Public Type MatchSummary
  MatchPlayerID As Integer
  MatchPlayerName As String * 30
  MatchPlayerTeam As String * 20
  Battingruns As Integer
  BattingBallsFaced As Integer
  BattingHowOut As String * 10
  BattingWktBowler As String * 30
  Bowlingovers As Integer
  bowlingruns As Integer
  Bowlingwickets As Integer
  BowlingmaidenRuns As Integer
  Listed As Boolean
End Type

Public Type TeamSummary
Public Type TeamSummary
    TeamName As String * 20
    Battingruns As Integer
    Wickets As Integer
    Overs As Integer
End Type

Public Type MatchStatistic
    Ballnumber As integer
    Runs As Integer
    Batsman As String * 30
    Bowler As String * 30
    Extra As String * 8
    Wicket As String * 10
    BatsmanOut As Integer
End Type

Public PlayerRecord As PlayerData
Public MatchData As MatchStatistic
Public TeamData As TeamSummary
Public searchteam As String
Public Battingorder(0 To 10) As String
Public homeTeam(0 To 10) As PlayerData
Public awayTeam(0 To 10) As PlayerData
Public SelectedPlayers As Integer
Public SearchDonc As Boolean
Public Teamsearch As Boolean
Public OpposingTeam As String
Public MatchMonth As String
Public MatchDay As Integer
Public i As Integer
Public Runsperswb As Integer
Public Extraball As Boolean
Public LimitedOvers As Integer
Public OpeningBatteam As String
Public j As Integer
Public OpeningBatsman As String
Public Bowled As Integer
Public OpeningNonfaceBat As String
Public OpeningBowl As String
Public CurBat As String
Public Otherbat As String
Public CurBowl As String
Public PlayerMatchData As MatchSummary
Public SummaryPath As String
Public BowlingTeam As String
Public BattingTeam As String
Public BatOut As String
Public BatA As String
Public BatB As String
Public k As Integer
Public l As Integer
Public Ballnum As Integer
Public Maiden As Boolean
Public StatisticPath As String
Public Extratype As String
Public WicketType As String
Public Chosen As Boolean
Public Secondinnings As Boolean
Public LastAction As String
Public LastBatName As String
Public LastBatFow As Integer
Public LastBatTotal As Integer
Public LastBatHowOut As String
Public objExcel As Object
Public Tempbat As String
Public TempBowl As String

Public Sub EndOver()
    Close #1
    Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
    For i = 1 To LOF(1) / Len(PlayerMatchData)
        Get #1, i, PlayerMatchData
        If PlayerMatchData.MatchPlayerName = TempBowl Then
            PlayerMatchData.Bowlingovers = PlayerMatchData.Bowlingovers + 1
            If Maiden = True Then
                PlayerMatchData.Bowlingmaidens = PlayerMatchData.Bowlingmaidens + 1
            End If
        End If
    Next i
    Tempbat = CurBat
    CurBat = Otherbat
    Otherbat = TempBat
    scoreboard.lblFacingBatsman = CurBat
    Ballnum = Int(scoreboard.lblOversBowled) * 20
    End Sub

Public Sub EndBall(i As Integer)
    Close #1
    Open StatisticPath For Random As #1 Len = Len(MatchData)
    I = LOF(1) / Len(MatchData) + 1
    With MatchData
        Ballnumber = Ballnum
        Bowler = CurBowl
        Batsman = CurBat
        If i > .1 And i < 7 Then
            Runs = i
        End If
        Extra = Exratype
        If WicketType <> "" Then
            Wicket = WicketType
            Open SummaryPath For Random As #2 Len = Len(PlayerMatchData)
            For j = 1 To 22
                Get #2, j, PlayerMatchData
                If PlayerMatchData.MatchPlayerName = BatOut Then
                    BatsmanOut = PlayerMatchData.MatchPlayerID
                End If
            Next j
            Close #2
        End If
    End With
End Sub
Else
    Wicket = ""
End If
End With
Put #1, 1, MatchData
Open SummaryPath For Random As #2 Len = Len(PlayerMatchData)
For j = 1 To 22
    Get #2, j, PlayerMatchData
    If PlayerMatchData.MatchPlayerName = CurBat Then
        PlayerMatchData.BattingBallsFaced = PlayerMatchData.BattingBallsFaced + 1
        Put #2, j, PlayerMatchData
    End If
Next j
Close #1
Close #2
Ballnum = Ballnum + 1
ExtraType = ""
WicketType = ""
End Sub

Public Sub EndInnings()
Close #1
Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
MsgBox "The innings has finished, click Ok to start the next innings", vbOKCancel
TeamData.TeamName = BowlingTeam
TeamData.BattingRuns = Scoreboard.lblRunsTotal.Caption
TeamData.Wickets = Scoreboard.lblWickets.Caption
TeamData.Over = Scoreboard.lblOversBowed.Caption
Put #1, 50, TeamData
Unload Scoreboard
OpeningBatsman = BowlingTeam
Scoreboard.cmdStartGame.Caption = "Start next Innings"
Scoreboard.lblLastInnings.Caption = TeamData.BattingRuns
Bowled = 0
Close #1
Unload StartPlayers
StartPlayers.Show
StartPlayers.cmdSelectback.Enabled = False
i = 1
SecondInnings = True
For i = 0 To 10
    BattingOrder(i) = ""
Next i
Ballnum = Int(Scoreboard.lblOversBowed) * 20
End Sub

Public Sub EndMatch(k As Integer)
If k = 1 Then
    MsgBox "BGS Staff have won the match!"
Else
    MsgBox OpposingTeam & ' have won the match!
End If
Open SummaryPath For Random As #2 Len = Len(PlayerMatchData)
Open App.Path & "\Cricketmasterfile.txt" For Random As #3 Len = Len(PlayerRecord)
For i = 1 To 22
  Get #2, 1, PlayerMatchData
  For j = 1 To LOF(3) / Len(PlayerRecord)
    Get #3, j, PlayerRecord
    If PlayerMatchData.MatchPlayerName = PlayerRecord.PlayerName Then
      With PlayerRecord
        .Runs = Runs + PlayerMatchData.BattingRuns
        If PlayerMatchData.BattingRuns > .Highscore Then
          .Highscore = .Runs
        End If
      End With
      If PlayerMatchData.BattingBallsFaced <> 0 Then
        .Innings = Innings + 1
      End If
      If PlayerMatchData.BattingBallsFaced <> 0 And Trim(PlayerMatchData.BattingHowOut) = "" Then
        .NotOuts = .NotOuts + 1
      End If
      .Overs = .Overs + PlayerMatchData.BowlingOvers
      .Wickets = .Wickets + PlayerMatchData.BowlingWickets
      .strikeRate = BowlingRuns / .Overs
      Put #3, j, PlayerRecord
    End If
  Next j
Next i
Close #2
Close #3
Scoreboard Show
Call UpdateFile
End Sub

Public Sub PreEndOver()
If Scoreboard.OversBowled + 1 >= LimitedOvers Then
  If SecondInnings = False Then
    Call EndInnings
  Else
    If BowlingTeam = "BGS Staff" Then
      k = 1
    Else
      k = 2
    End If
    If CurBowler <> BowlingTeam(k) Then
      CurBowler = BowlingTeam(k)
    End If
  End If
Else
NextPlayer.Caption = "Select the next Bowler"
Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
  For i = 1 To LOF(1) / Len(PlayerMatchData)
    Get #1, i, PlayerMatchData
    If PlayerMatchData.MatchPlayerName <> CurBowler And
       Trim(PlayerMatchData.MatchPlayerTeam) = Trim(BowlingTeam) Then
      NextPlayer.cmbNextPlayer.AddItem PlayerMatchData.MatchPlayerName
    End If
  Next i
End If
If SecondInnings = True Then
  Call Scoreboard.ShowScoreboard
Else
  Call Scoreboard.CloseScoreboard
End If
End If
Next i
NextPlayer.cmdProlong.Visible = True
j = 0
NextPlayer.Show
End If
End Sub

Public Sub UpdateFile()
If Dir(App.Path & "\"Spreadsheet\MasterFile.xlsx") <> "" Then
Kill App.Path & "\"Spreadsheet\MasterFile.xlsx"
End If
Set objExcel = CreateObject("Excel.sheet")
Open App.Path & "\"CricketMasterFile.txt" For Random As #1 Len = Len(PlayersRecord)
For i = 1 To LOF(#1) / Len(PlayersRecord)
Get #1, i, PlayersRecord
objExcel.application.UpDown(4, 2) = PlayersRecord.PlayerID
objExcel.application.UpDown(4, 3) = PlayersRecord.PlayerName
objExcel.application.UpDown(4, 4) = PlayersRecord.PlayerTeam
objExcel.application.UpDown(4, 5) = PlayersRecord.Innings
objExcel.application.UpDown(4, 6) = PlayersRecord.AverageRuns
objExcel.application.UpDown(4, 7) = PlayersRecord.Highscore
objExcel.application.UpDown(4, 8) = PlayersRecord.NotOuts
objExcel.application.UpDown(4, 9) = PlayersRecord.Overs
objExcel.application.UpDown(4, 10) = PlayersRecord.Runs
objExcel.application.UpDown(4, 11) = PlayersRecord.Wickets
objExcel.application.UpDown(4, 12) = PlayersRecord.Maidens
objExcel.application.UpDown(4, 13) = PlayersRecord.EconRate
objExcel.application.UpDown(4, 14) = PlayersRecord.strikerrate
Next i
objExcel.application.UpDown(4, 2) = "Player ID"
objExcel.application.UpDown(4, 3) = "Name"
objExcel.application.UpDown(4, 4) = "Team"
objExcel.application.UpDown(4, 5) = "Inns"
objExcel.application.UpDown(4, 6) = "Avo"
objExcel.application.UpDown(4, 7) = "High Score"
objExcel.application.UpDown(4, 8) = "Not Outs"
objExcel.application.UpDown(4, 9) = "Overs"
objExcel.application.UpDown(4, 10) = "Runs"
objExcel.application.UpDown(4, 11) = "Wickets"
objExcel.application.UpDown(4, 12) = "Maidens"
objExcel.application.UpDown(4, 13) = "Econ. Rate"
objExcel.application.UpDown(4, 14) = "Strike Rate"
objExcel.SaveAs App.Path & "\"Spreadsheet\MasterFile.xlsx"
objExcel.application.quit
Close #1
Call UpdateFile2()
End Sub

Public Sub UpdateFile2()
If Dir(App.Path & "\"Spreadsheet\"" & OpposingTeam & MatchStart.cmbYY.Text & ".xlsx") <> "" Then
Kill App.Path & "\"Spreadsheet\"" & OpposingTeam & MatchStart.cmbYY.Text & ".xlsx"
End If
Set objExcel = CreateObject("Excel.sheet")
Open SummaryPath For Random As #1 Len = Len(PlayersMatchDate)
For i = 1 To 22
Get #1, 1, PlayerMatchData
With PlayerMatchData
  objExcel2.application.cells(i + 4, 2) = .MatchPlayerID
  objExcel2.application.cells(i + 4, 3) = .MatchPlayerName
  objExcel2.application.cells(i + 4, 4) = .MatchPlayerTeam
  objExcel2.application.cells(i + 4, 5) = .BattingRuns
  objExcel2.application.cells(i + 4, 6) = .BattingBallsFaced
  objExcel2.application.cells(i + 4, 7) = .BattingHowOut
  objExcel2.application.cells(i + 4, 8) = .BattingWktBowler
  objExcel2.application.cells(i + 4, 9) = .BowlingOvers
  objExcel2.application.cells(i + 4, 10) = .bowlingruns
  objExcel2.application.cells(i + 4, 11) = .Bowlingwickets
  objExcel2.application.cells(i + 4, 12) = .Bowlingmaidens
End With
Next i
With PlayerMatchData
  objExcel2.application.cells(3, 2) = MatchStart.cmbDD.Text & "/" & MatchStart.cmbMM.Text & "/" & MatchStart.cmbYYYY.Text
  objExcel2.application.cells(4, 2) = "Player ID"
  objExcel2.application.cells(4, 3) = "Name"
  objExcel2.application.cells(4, 4) = "Team"
  objExcel2.application.cells(4, 5) = "Runs Scored"
  objExcel2.application.cells(4, 6) = "Balls Faced"
  objExcel2.application.cells(4, 7) = "How Out"
  objExcel2.application.cells(4, 8) = "Bowler"
  objExcel2.application.cells(4, 9) = "Overs"
  objExcel2.application.cells(4, 10) = "Runs Against"
  objExcel2.application.cells(4, 11) = "Wickets"
  objExcel2.application.cells(4, 12) = "Maidens"
End With
objExcel2.SaveAs App.Path & "/\Spreadsheets\" & OpposingTeam & MatchStart.cmbYYYY.Text & ".xlsx"
Close #1
End Sub
Error Messages

Due to the nature of the system there is not a lot of validation needed. Where validation is needed I have included the following errors:

<table>
<thead>
<tr>
<th>Message Reason</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selecting more than 1 team when searching</td>
<td>More than 1 team has been selected</td>
</tr>
<tr>
<td>Selecting more than 11 players in the search</td>
<td>More than 11 players have been chosen</td>
</tr>
<tr>
<td>Not selecting the opening batting team in the</td>
<td>Select a team to open the batting</td>
</tr>
<tr>
<td>settings</td>
<td></td>
</tr>
<tr>
<td>Not selecting one or more of the players to open</td>
<td>You have not selected one or more of the option(s)</td>
</tr>
<tr>
<td>the batting and bowling</td>
<td></td>
</tr>
<tr>
<td>Attempting to view the scoreboard before a complete over</td>
<td>The scoreboard cannot be shown until a full over is bowled.</td>
</tr>
</tbody>
</table>
Section 5 – Evaluation

Due to the time restrictions in making the system, there were several requirements I was unable ensured my users were aware of all areas of the specification I was unable to complete.

All the screenshots (SS’s) listed here are from the original test on pages 44 to 64.

<table>
<thead>
<tr>
<th>Requirement Spec</th>
<th>Met?</th>
<th>Proof</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system will ask for the names of all the players at the start of the match for at least the BGS staff team.</td>
<td>Yes</td>
<td>SS 5 &amp; SS 6</td>
<td>The system will also ask for the opponents names straight away.</td>
</tr>
<tr>
<td>The names of the opposition team can be entered separately when that relevant player comes into play during the game or at the end of the match</td>
<td>No</td>
<td>None</td>
<td>I was not able to implement a feature to allow the user to change the name of players during the match.</td>
</tr>
<tr>
<td>The system will always know which batsman is facing so that if runs are scored off the bat then the system will add the runs to the correct batsman.</td>
<td>Yes</td>
<td>SS 17 - 22</td>
<td>This requirement was fully met.</td>
</tr>
<tr>
<td>The system will add runs to the total whenever they are inputted</td>
<td>Yes</td>
<td>SS 17 - 22</td>
<td>This requirement was fully met.</td>
</tr>
<tr>
<td>The system will add runs to the correct section of the scoreboard (either batsman or the extras total)</td>
<td>Yes</td>
<td>SS 17 - 22, SS 29 &amp; 30</td>
<td>This requirement was fully met.</td>
</tr>
<tr>
<td>When a wicket is taken the scoreboard will add that wicket on to the total number of wickets</td>
<td>Yes</td>
<td>SS 39 - 41</td>
<td>This requirement was fully met.</td>
</tr>
<tr>
<td>When a wicket is taken the system will ask for the name of the new batsman which can be selected from a list of batsman or entered by the user</td>
<td>Yes</td>
<td>SS 40</td>
<td>The user has to choose from the list of players created at the start of the match.</td>
</tr>
<tr>
<td>The system will have a separate window with a full scorecard of the whole match</td>
<td>Yes</td>
<td>SS 45</td>
<td>This requirement was fully met.</td>
</tr>
<tr>
<td>The scorecard will display 2 tables per innings: one table containing the batting figures and the other the bowling figures.</td>
<td>Yes</td>
<td>SS 45</td>
<td>This requirement was fully met.</td>
</tr>
<tr>
<td>The batting table will show the runs scored by a batsman, and if that batsman is out: the method by which they were out.</td>
<td>Yes</td>
<td>SS 45</td>
<td>This requirement was fully met.</td>
</tr>
<tr>
<td>The bowling table will show each bowler which has bowled, with the number of runs they have had scored against them, the number of overs they have bowled, the number of wickets they have taken, and the number of maidens they have bowled.</td>
<td>Yes</td>
<td>SS 45</td>
<td>This requirement was fully met.</td>
</tr>
</tbody>
</table>
At the end of the match the system will produce 4 tables, a batting and bowling table from each innings, along with a summary statement.

This requirement wasn't fully met, as there is no summary statement created. Also the figures for all players batting and bowling is merged into 1 table, rather than split into 4.

### Problems when creating the system

The main problem that came about relatively soon after I started making the program was the scale of the project. I soon realised that I would be unable to create everything I had been asked to in the requirement specification. Having realised this, my next task came to prioritise the requirements. Using information I had gathered from the interviews, I found which requirements were merely extra features, and which were fundamental to the system itself. From this I decided to not include the following features.

- The interface to look up the season statistics for any players.
  This feature doesn't add much to the system. It simply allows users to view statistics using the program. However these statistics are also easily accessible from the master spreadsheet.

- The ability to edit the information of any ball.
  Although this was a main feature requested by the users, when thinking about programming this I could see that it would be a very tough feature to add in. This is mainly because it has the potential to create many bugs from the multiple changes in data that would need to occur. I also felt this feature would grant the user too much power to merely edit any part of the score they wish, and would lose the element of rigidity in the score shown by the system.

- The ability to edit player names or add players when they come into play.
  This feature again isn't essential. It will force the user to be organised before the match to obtain the names of all 22 players who are playing. But due to time restrictions again this was a feature that had to be left out of the system.

Given more time I would implement all these features into the system, and they certainly create scope for improvement.

When creating the system itself, I had many coding problems, small and large. This is a summary of the more significant problems I came across.

- The first part of the program I created was the file structure. I myself had done very little work with random access files in Visual Basic in the past, so using these files created several small syntax errors in the code due to inexperience, however I am now more confident at programming using random access files as these errors are rarer.

- It took a long time to get the Matchstart form fully working in conjunction with the search form. Firstly to get the search to display the correct lists with which to search from, then also to transfer the selected options back onto the Matchstart form.

- The settings form was relatively straightforward to create, there were a few interface problems as I was using 2 different sets of radio buttons on the same window. This was sorted by placing forms round both sets of radio buttons.

- Once the game had started, problems developed due to the number of different routes that can be taken to complete a ball. I found that processes were overlapping in certain occasions or weren't
being run at all. To sort this I ended up making several different modules to each carry out specific portions of the tasks required depending on what had happened that ball.

- A problem I had testing the system is that it would often become quite laborious to test the system multiple times as the user is required to go through several options before they can reach the situation where they are playing a match. To solve this I created test scenarios (as used in subsidiary tests in response to user testing), to do this I created another command button on the Matchstart form which took the user straight to the startplayers form with a pre set 22 players. This code to do this is as follows:

```vba
Private Sub cmdTestSetup_Click()
    OpposingTeam = "Test Opp"
    Open App.Path & "\" & txtOpposition.Text & cmbDD.Text & cmbMM.Text & cmbYY.Text & ".txt" For Random As #1
    Len = Len(PlayerMatchData)
    For i = 0 To Len
        With PlayerMatchData
            .MatchPlayerID = i + 1
            .MatchPlayerName = "BGS" & i
            .MatchPlayerTeam = "BGS Staff"
            .Battingruns = 0
            .BattingBallsFaced = 0
            .BattingHowOut = ""
            .BattingWktBowler = ""
            .Bowlingovers = 0
            .Bowlingruns = 0
            .Bowlingwickets = 0
            .Bowlingmaidens = 0
        End With
        Put #1, i + 1, PlayerMatchData
    Next i
    For i = 0 To Len
        With PlayerMatchData
            .MatchPlayerID = i + 12
            .MatchPlayerName = "Opp" & i + 1
            .MatchPlayerTeam = "Toss Opp"
            .Battingruns = 0
            .BattingBallsFaced = 0
            .BattingHowOut = ""
            .BattingWktBowler = ""
            .Bowlingovers = 0
            .Bowlingruns = 0
            .Bowlingwickets = 0
            .Bowlingmaidens = 0
        End With
        Put #1, i + 12, PlayerMatchData
    Next i
    Close #1
    ExtraBall = True
    KupserRun = 1
    LimitedOvers = 20
    OpeningBatTeam = "BGS Staff"
    Me.Hide
    StartPlayers.Show
End Sub
```
In addition to this, I also created buttons on the scoreboard to set the wickets to 9 and the overs to 19 (so the end of innings could be tested easily):

Private Sub cmd9wicks_Click()
lblWickets.Caption = "9"
lblOversBowled.Caption = "19"
End Sub

I also put in a button that allowed me to update the files to spreadsheets straight away

Private Sub Command1_Click()
Call UpdateFile
End Sub

- However using this predesigned setup also created a problem which I didn't notice for quite a while. Because the name of the opposition and the date was exactly the same for every test, the system didn't make a new file, but simply added data onto the existing file, which created very confusing results when testing the scoreboard window.

- The scoreboard window itself created many problems to create. Firstly when the array of labels was created, their visible property was set to false. In addition there width and height was often not sufficient enough to fit the data entered into them. It also took a long time to figure out how to configure arrays and loops to sort out the bottom table which shows what happens every ball, for every over, labelled next to each different bowler.

- The final part in creating the system was the spreadsheets, this didn't cause many problems, however one problem was that of trying create/edit two separate spreadsheets in the same subroutine. This was fixed simply by separating the two bits of code into separate modules.

- Another problem that came more significant as the project went on was the use of variables. I ended up with 5 different variables purely for random use (i.e to store the number of runs scored or other parameters), this lead to some variables being overlapped and changed unnecessarily. This problem occurred more often as the project went on as I lost track of which variables shouldn't be changed as they were storing important data.

- To help this problem I could have used more variables with meaningful names to help keep track of what was being stored in each variable. I ended up using the letters 1, j, k, l and p. Using I was a bad choice as at a glance it looks similar to 1.
Good and Bad Points of the System

Some of the good points about the system are:

From looking back through my initial specification and user response, the system does carry out the task it was designed to do efficiently and effectively. One of the main good things about the system is the main scoreboard interface. How the visual display of the scoreboard represents a real life scoreboard clearly, so any cricketer could look at the scoreboard and easily recognise the score. Another feature that makes the system good is the scorecard, previously using a scorebook the scorer would have to manually calculate any statistics the players wanted to know. Now the user can simply click on the scorecard and all the statistics for the players will be there.

The system also accurately keeps track of things like balls faced by each batsman, and will also calculate economy and strike rate for bowlers, and averages for batsmen. These statistics are not just recorded each match, but they are kept throughout the season for every player.

Opposition players are also kept track of, so an opposition player can easily see how they have done in previous years against the BGS Staff, this would have been very difficult before using a scorebook.

As for scoring a game of cricket itself, the system makes it much easier than using a scorebook, the majority of balls will take 3 key strokes or mouse clicks or less to input. Once the game is being played the system is very easy to use and any player could easily take over scoring, whilst before they would need to know the notion of the scorebook, in addition several different users could score using this system and the output would remain the same.

Some bad points or limitations include:

Unfortunately the game of cricket has a number of exceptionally rare outcomes from a ball. To include all of these into the system would simply make it much more complex than needed. This is a possible extension to create a feature that brings up a new window for rare outcomes. These would include 7 runs from one ball, a player returning to bat after retiring or a 12th man coming on to replace another player.

Another limitation is the rigidity of the system. There is a lot of data that once entered cannot be changed, such as player names or the settings. As I discussed earlier in my evaluation the system needed to maintain a good balance between rigidity and flexibility. If I included too many features to allow the user to edit anything, then the system would lose the ability to be fully trusted by a scorer, as it would be easy to tamper with the score. However mistakes are made and so I have included some editing features, such as the undo button.

Along with this would be a feature to allow the user to change the names of players, this would be a useful extension as the user may often not know all of the opposition names at the start of the match, so to be able to change them mid way through the match would ultimately be useful.

Possible Extensions:

So in summary these are the possible extensions:

Editing a ball

This would be achieved by altering the scorecard window. I would implement a feature that allowed the user to click on any over to select a ball which they wish to edit. From this they would be able to edit all the information about the ball. After the changes the statistics file would be updated accordingly for the edited ball. The system would then recalculate the current score from the changes made, and update the scoreboard appropriately.
Editing a name

This feature would be created by the user clicking a separate button which would bring up a window similar to the match start window. This would list all 22 players who are playing and the name of the opposition team. The user would be able to edit any of the names including the opposing team. Once they press Done the system would search through the master file and change the name, and the summary file to also change the name. The statistics file would then be searched and again the old name replaced with the new name.

Exceptional circumstances

This would be a feature that is simply added onto manual override. Extra buttons would be created for more than 6 runs being scored, this would bring up an extra window allowing the user to enter how many runs are scored into a text box. A button for a retired player coming back in which would set any players who are retired to a status where they haven’t batted yet. A button to end the innings early, this would allow the user to set a new number of limited overs for which the second innings is played to.
Table of Contents

User Guide .................................................................
Introduction or Preface .................................................................
1 Hardware and Software Requirements .................................................................
2 Instructions .................................................................
  2.1 Installation .................................................................
  2.2 The Interface .................................................................
3 Runs .................................................................
4 Extras .................................................................
5 Wicket .................................................................
6 End of Over .................................................................
7 Short Run & Undo .................................................................
8 End of Innings .................................................................
9 End of the Match .................................................................
10 Spreadsheets .................................................................
11 Scorecard .................................................................
12 Problem Solving .................................................................
  3.1 Error Message .................................................................
  3.2 Troubleshooting .................................................................
  3.3 Contact Details .................................................................
13 Back-up Routines .................................................................
14 Glossary .................................................................
15 Index .................................................................

Introduction or Preface
Scoreboard is a program that allows you to score a cricket match from the sideline of a cricket pitch on a computer. This manual will show you how to operate the system to make the most of the features provided.

1 Hardware and Software Requirements
To run Scoreboard you will need the following Hardware/Software:
  Windows 98 or above
  Computer With keyboard and mouse
  If using a laptop a means of keeping power for the entire match (either by a spare battery or a mains connection)
  Network connection (to upload the stats once the game has finished)

2 Instructions

2.1 Installation
To install Scoreboard, simply unzip the Scoreboard.zip file and run Scoreboard.exe.

2.2 The Interface
When opening Scoreboard you will view the following interface:

Simply click Start Game to start a new game.
The following window will then come up:

**Select Players**

Match Start - Enter teams

**Date:**
- Select Month
- Day
- Year

**Opposition:**
- Opposition
- Search

<table>
<thead>
<tr>
<th>BGS Staff</th>
<th>Opposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player 1</td>
<td>Player 12</td>
</tr>
<tr>
<td>Player 2</td>
<td>Player 13</td>
</tr>
<tr>
<td>Player 3</td>
<td>Player 14</td>
</tr>
<tr>
<td>Player 4</td>
<td>Player 15</td>
</tr>
<tr>
<td>Player 5</td>
<td>Player 16</td>
</tr>
<tr>
<td>Player 6</td>
<td>Player 17</td>
</tr>
<tr>
<td>Player 7</td>
<td>Player 18</td>
</tr>
<tr>
<td>Player 8</td>
<td>Player 19</td>
</tr>
<tr>
<td>Player 9</td>
<td>Player 20</td>
</tr>
<tr>
<td>Player 10</td>
<td>Player 21</td>
</tr>
<tr>
<td>Player 11</td>
<td>Player 22</td>
</tr>
</tbody>
</table>

**Search BGS Players**

**Search Opposition Players**

Firstly, enter the date at the top, select the Month first to make a list of days appear.

Then you will need to select your opposition. If you have played the opposition before using this system, you can click search, and the search window on the next page will come up. If you haven't played the opposition before, simply type the name of the team into the text box.

You will then need to select your players. For the BGS side, press the Search BGS Players button. This will bring up the search window again listing players who have played for BGS through this system previously. Select these players, and type in the names of any remaining players in the remaining boxes.

Likewise for the opposing team. If you have selected a team from the search, then you will be able to search for players who previously played for that team, then fill in any remaining boxes with new players. Otherwise, if the opposition team is new, then all new players will need to be typed in.
Once both teams have all players filled, press Done. The settings window will then come up:

On this window, select the number of runs per wide/no ball, if an extra run is given for a wide/no ball, how many overs the game is limited to and which team is batting first. Once you have selected all these options press OK.

On this window, select the next window will appear to select the starting players:

Once you have done that, the next window will appear to select the starting players:

On this window simply select the facing batsman, non facing batsman and opening bowler from the lists. Then press Done.

(Note: it is not required to define the batting order before the game).

The game can now start, and the scoreboard window will show up.

This window is the Scoreboard window. This will be the window used most of the time when scoring a match.

The Scoreboard more or less represents a real life scoreboard; it is real time meaning that it will be updated as soon as a ball has taken place. The buttons in the bottom right box give the options for what can happen every ball.

 Runs 

On a ball where a number of runs is scored, simply press the corresponding number to the number of runs scored.
**Extras**

For any extra press the relevant extra button. Once you press any of these extras, the following window will appear:

![Extra Scored Window]

On this window simply select how many additional runs were scored from the extra.
(Note for a wide/no ball enter the additional runs scored by the batsman, not the initial runs scored from the wide/no ball itself).

Once the number of runs is selected, press Ok
(Note if a no ball you will be asked if the extra runs scored are byes or runs off the bat).

**Wicket**

On the event of a wicket, press the wicket button and the following window will come up:

![Wicket Window]

Select the method by which the batsman was out. If the batsman was caught and the batsman crossed in the process, check the batsman crossed box.

Once you have done this, press Done. If a Run Out has occurred then an extra window will come up:
Short Run & Undo

To undo an action, simply press the undo button and the application will return to the state of the previous ball.

To put a short run in, simply press short run after any number of runs has been scored in a ball.

End of Innings

When the end of the first innings is reached, you will receive this pop up.

Scoreboard

The innings has finished, click OK to start the next innings

OK Cancel

Press OK to this pop up, and you will then need to select the Opening batsman and bowler for the second innings. Once you have done this the second innings will start in the same way the first innings did.

End of the Match

When either team has won the match, a pop up will appear, displaying which team has won the match. Once this has happened a spreadsheet for the match itself will be created, and the “Master Spreadsheet” will be updated.

Spreadsheets

Once a match has finished, a spreadsheet showing statistical data from that match will be created. To access this spreadsheet simply go to the folder in which the program is saved, go into the Spreadsheets folder and the spreadsheet for each game will be titled with the name of the opposition followed by the year.
Scorecard
At any time in the game you may view the scorecard for the innings being played at that time. This will show batting and bowling figures for all players who have batted/bowled, and will also show each over split down into each ball.

(Note you cannot view the scorecard before one over has been completed in the innings, also a bowler who has not completed a full over may not appear on the scorecard)

3 Problem Solving

3.1 Error Message
All error messages will come up in a pop up box whenever an incorrect action is made. Due to the nature of the system there are not many errors that can be made.

3.2 Troubleshooting
When selecting players I selected the wrong player by accident.

If you have put a player on the selection list, simply highlight them and press the left arrow to take them away from the list. If the player is now selected in a greyed out box, you must restart the program, so double check your selection of players before press Done.

3.3 Contact Details
For major problems:

Email gliver.taylor@blueyonder.co.uk

4 Back-up Routines
After every ball the data from that ball is loaded into the statistics file for the game. So this file is kept up to date at all times.

For manual backup, copy the Scorefile folder and the CricketMasterFile.txt to another medium (such as a USB stick).
The BGS IT system does backup all files over the weekend, so files will be backed up externally on a weekly basis.

5. Glossary

Hardware: The physical components of a computer

Software: The virtual components of a computer

USB Stick: A device external to the computer that can store data.

Window: An individual screen within the program

6. Index
Back-up 12
End 10
Innings 11
Match 11
Errors 12
Extras 8
Installation 3
Player Selection 4,5
Requirements 3
Runs 7
Scorecard 12
Settings 6
Spreadsheets 11
Undo 11
Wicket 9
F454: Project Commentary
Project commentary:

This is a project submitted for the legacy specification, but it includes some excellent examples of what is required for the new specification, though there will inevitably be some omissions, some extra information and the organisation of the report may differ from that expected for the new specification.

Some elements of this project have been removed because they are no longer required and would give the wrong impression.

Section a, Definition, Investigation and Analysis.

a(i) Section a(i) for the old and new specifications are very similar and this candidate identifies the end users, what they do, what roles they play in the organisation, the nature of the problem to be investigated and the sort of data involved. [3/3]

a(ii) Once again the two specifications are very similar and this candidate has clearly worked with the end users to investigate what is required of the system. Evidence includes DFD’s for the existing system, original documents from the existing system, planning for and transcripts of interviews (original interview notes taken during the interview would be useful), the data is identified and hardware and software specifications. The candidate has also looked at commercially available software to inform decisions. Overall there is good/excellent evidence of end user involvement, all aspects have been covered fully or at least very well and this clearly belongs in the top range of marks for this section. [9/11]

Section b, Design

b(i) this section of the new specification is once again similar to the equivalent section of the old specification but with new sections for Algorithms and test strategies the work from this candidate will not be in the order expected for the new specification, however, much of what is required can be identified and credited accordingly. In b(i) we are looking for the end users’ requirements being developed into a workable design that could be implemented. This section should include measureable objectives, designs for the user interface, data capture forms and reports, processes, variables with data type and validation and data structures. The overall design needs to be agreed with the end user. Most of these items can be found within the candidate’s design section, including some evidence of end user agreement, though a signature, and/or some comments on the designs might be useful. Overall there is good evidence of design in this candidate’s work and it clearly belongs in the top range of marks. [5/6]

b(ii) requires the candidate to develop and test suitable algorithms for their solution to the problem. This candidate has produced some outline algorithms that are based on the analysis performed, but does not show that these have been tested. Typically the candidate should be showing that the set of algorithms provides a complete solution to the problem works together and perform as expected. This might be shown using a
simple trace and compared to the original requirements to show that all required outcomes are achieved. However there are algorithms, they do relate to the task and consequently some credit needs to be given in the second category of marks in the marking guidance.  

b(iii) requires a test strategy to be identified. The strategy should identify how the system will be tested to show how it achieves the desired outcomes. This candidate has clearly thought carefully about how the system should be tested, has identified test data, and expected results. Hence a top range mark

Section c Software Development and Testing

This section has been allocated a significantly higher proportion of the marks than in the previous specification and consequently we are looking to the candidate to explain how the system developed including testing of the system during development and consequent changes. While this candidate was not asked specifically to do this there is excellent evidence to show the system being modified in light of testing, both by the candidate and by the end users. The code produced is annotated reasonably clearly and it is fairly clear how the sections of code relate to the solution produced and are interrelated. The overall mark for this section must therefore fall within the top band but perhaps, due to some limitation in the annotation and evidence of modularity at the lower end of the range.

The testing is extensive and we are left with little doubt that this system works, there is extensive evidence for end user involvement at this stage and a top mark is appropriate.

Section d, Documentation

This is a much slimmed down section of the project from the previous specification and the technical documentation is taken from the preceding sections. We also require good on screen help rather than extensive user guides. Paper documentation may also be required to include basic installation, use and troubleshooting. Evidence for good on screen help may be seen in the development section, there is obvious consideration for error messages and there is a basic user guide with all the other information required to make sue of the solution.

Section e, Evaluation

e(i) requires the candidate to go back to the system objectives and show how each of these has been met, or if not explain why not. There are no significant omissions in this candidate’s report for this section and evidence is clearly identified.

e(ii) is slightly different to the equivalent section of the previous specification and requires the candidate to discuss the end user’s response rather than simply provide a letter saying it worked perfectly. While the system may work perfectly, it is more likely
for the system to have some minor flaws or not quite match the original specification, this is reasonable and candidates need to be able to respond effectively to these issues if they occur. If the end user writes a response indicating that there are some issues, full marks for this section can still be obtained if the candidate comments on how these might be rectified or dealt with. In this case the system has been tested, some minor faults identified and the candidate has dealt with them. [3/3]

e(iii) there is an extensive discussion about the system it’s good and bad points, any extensions and how these might be dealt with. [3/3]

Overall [70/80]