OCR
RECOGNISING ACHIEVEMENT

H447 Unit F454: Computing Project

F454: Project

Contents

A2 Project 1
Contents 2
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 15
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 Reson 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
Second User Testing 74

Implementation plan 74
Annotated Listings 77
Section 4 - Documentation 112
Hardware and Software Specification 112
Data Structures 112
Back-up and Recovery Technical Procedures 112
Data Flow Diagram 113
Master File Updated 113
Input 114
Output 115
Annotated Code Listing 115
Error Messages 116
Message Reson 116
Error 116
Section 5 – Evaluation 117
Problems when creating the system 118
Good and Bad Points of the System 121
Possible Extensions: 121

Section 1 – Definition, Investigation and Analysis

Introduction

My name is (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 staff cricket team, M

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?
: 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 firefly (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 scorer from watching the game, it is then written into the scorebook, after the game has finished 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, Maidens (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?
: As a scorer one of the most common errors 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 just an unformatted spreadsheet with data on it. Sometimes it is simply easier just to upload the results straight onto firefly rather then 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 5 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 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 often ends up scoring the match using the score book and after the match n 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 deta 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 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 then 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 bowlers 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 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 played against.

At the start of every innings the user will be asked which team is batting, which 2 batsman are opening (indicating which one is facing first) and which bowler is bowling. From this a database should be created for the BGS 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.

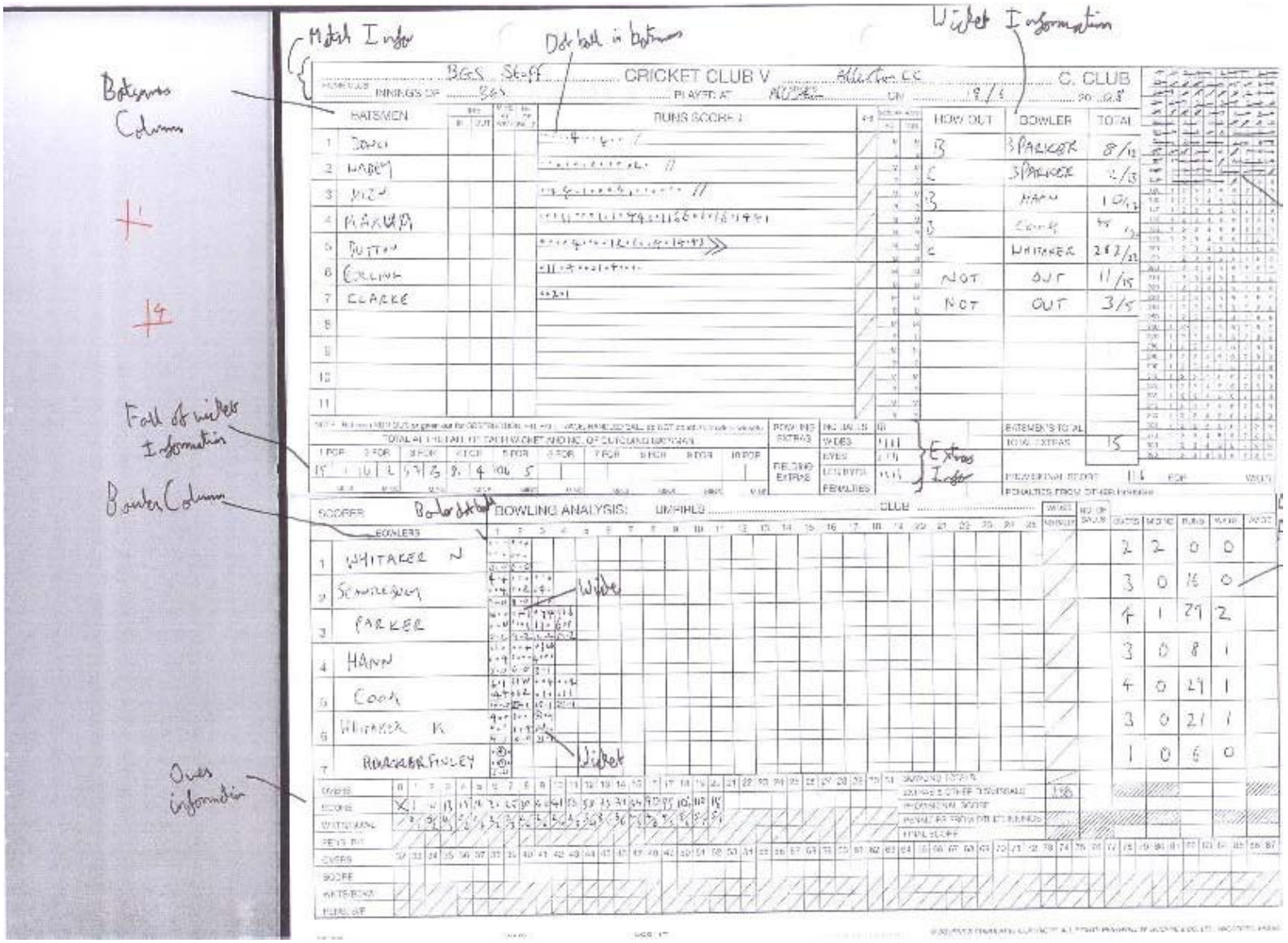
Any outcome which requires extra information (such as a no ball) will cause a menu to pop up with the relevant options for what could happen.

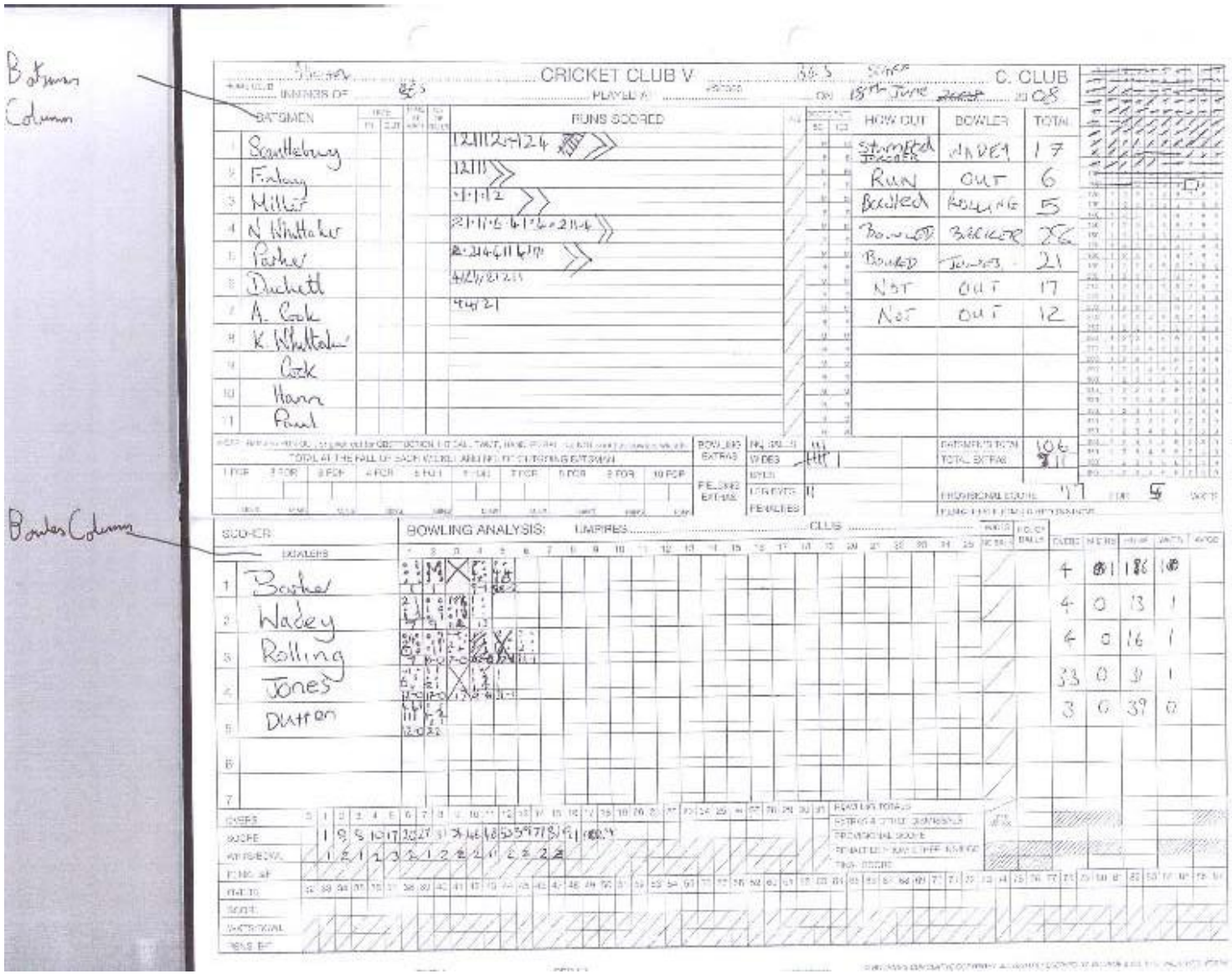
The system will provide the ability to change features such as players and scores, and the user will also be able to override the automatic features of the system (such as when an over finishes).

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.


Match Infor
Batsmans Column
Fall of wicket Information
Bowler Column
Overs information
Wicket Information
BGS Staff
CRICKET CLUB V
Allerton CC
C. CLUB
BATSMEN
RUNS SCORED
HOW OUT
BOWLER
TOTAL
Extras Infor
Wide
Wicket



Batsmen Column

Bowler Column

CRICKET CLUB V C. CLUB

INNINGS OF BES PLAYED AT ON 18th June 2008

	BATSMEN	RUNS SCORED	HOW OUT	BOWLER	TOTAL
1	Southebury	12111214124	Stumped Jones	WADEY	17
2	Finlay	1211	RUN	OUT	6
3	Miller	1·1·12	Bowled	ROLLING	5
4	N Whittaker	21·11·5·4·1·4·2·11·4	Bowled	BARKER	26
5	Parker	2·2·4·4·11·4·1·1	Bowled	JONES	21
6	Duckett	4·2·1·2·1·2·1	NOT	OUT	17
7	A. Cook	4·4·21	NOT	OUT	12
8	K. Whittaker				
9	Cook				
10	Hann				
11	Paul				

TOTAL AT THE FALL OF EACH WICKET AND NO. OF OUTGOING BATSMAN

1 FOR 2 FOR 3 FOR 4 FOR 5 FOR 6 FOR 7 FOR 8 FOR 9 FOR 10 FOR

BOWLING EXTRAS: NO. BALLS 11; WIDES 1111; BYES; LEG BYES 11; PENALTIES

FIELDING EXTRAS

BATSMEN'S TOTAL 106; TOTAL EXTRAS 11

PROVISIONAL SCORE 117 FOR 5 WKTS

SCORER

BOWLING ANALYSIS: UMPIRES CLUB

	BOWLERS	OVERS	MIDENS	RUNS	WKTS	AVGS
1	Barker	4	1	186	10	
2	Wadey	4	0	13	1	
3	Rolling	4	0	16	1	
4	Jones	33	0	31	1	
5	Dutton	3	0	39	0	
6						
7						

OVERS: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

SCORE: 1 8 5 10 17 20 27 31 34 46 48 50 59 71 81 91 100 104

WKTS/BOWL: 1 2 1 2 3 2 1 2 2 2 1 2 2 2 2

FIELDING SIDE

OVERS

SCORE

WKTS/BOWL

RENS BIT

READING TOTALS

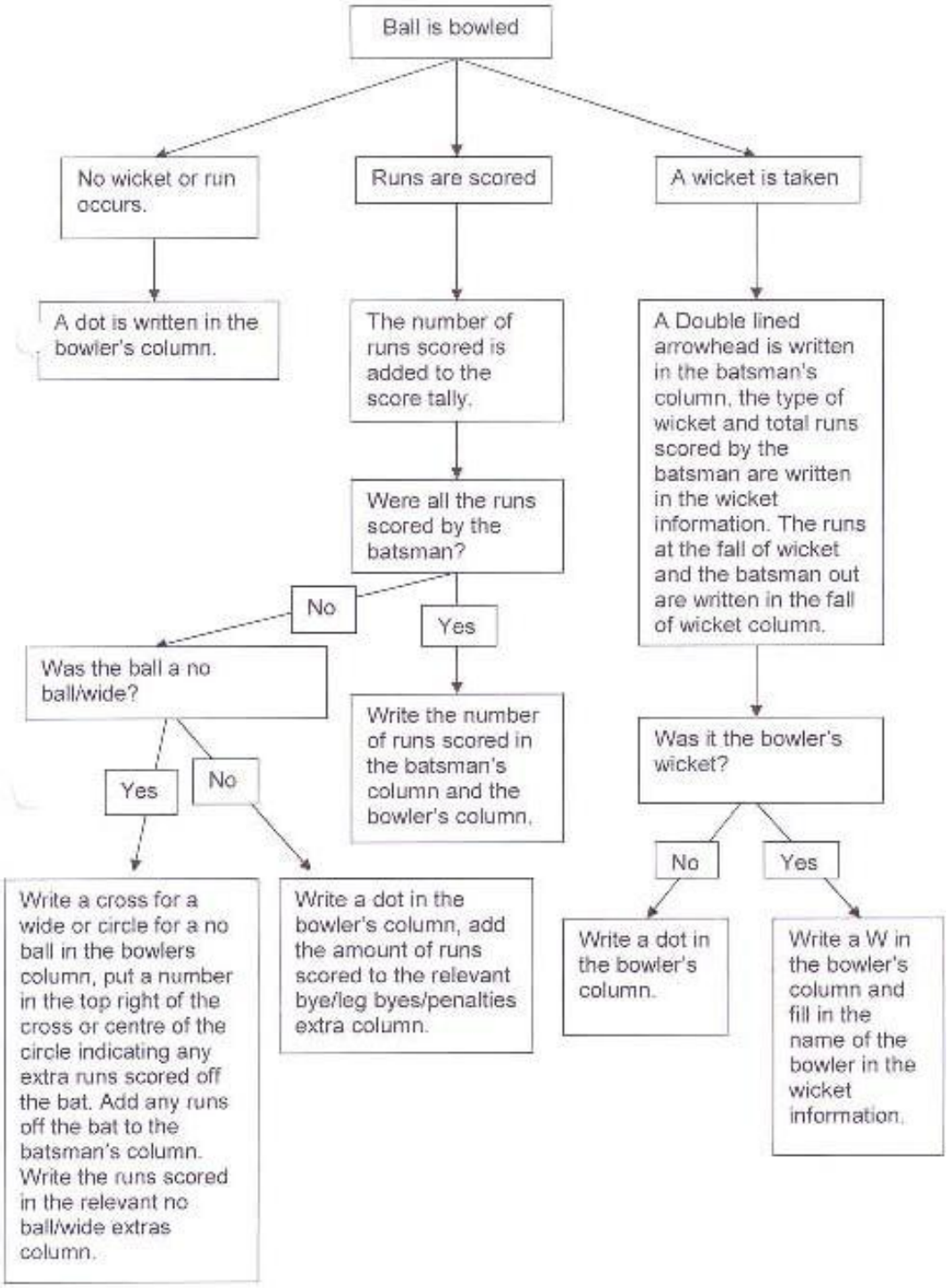
EXTRAS & OTHER DISMISSALS

PROVISIONAL SCORE

RENAULT LEFT BAY OTHER INVOICE

FINAL SCORE

This is a flowchart for the scorebook at a cricket match



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 then 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.

Data Flow of Current System

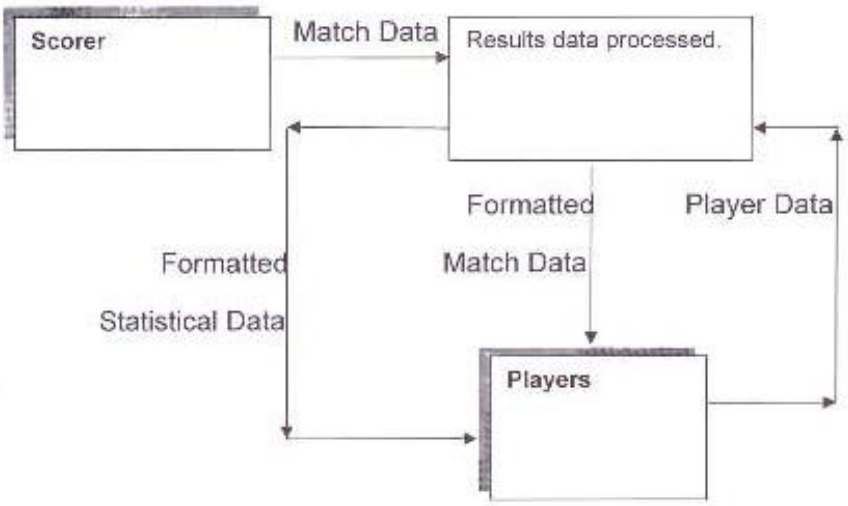
Key:

Shaded background box = Entity

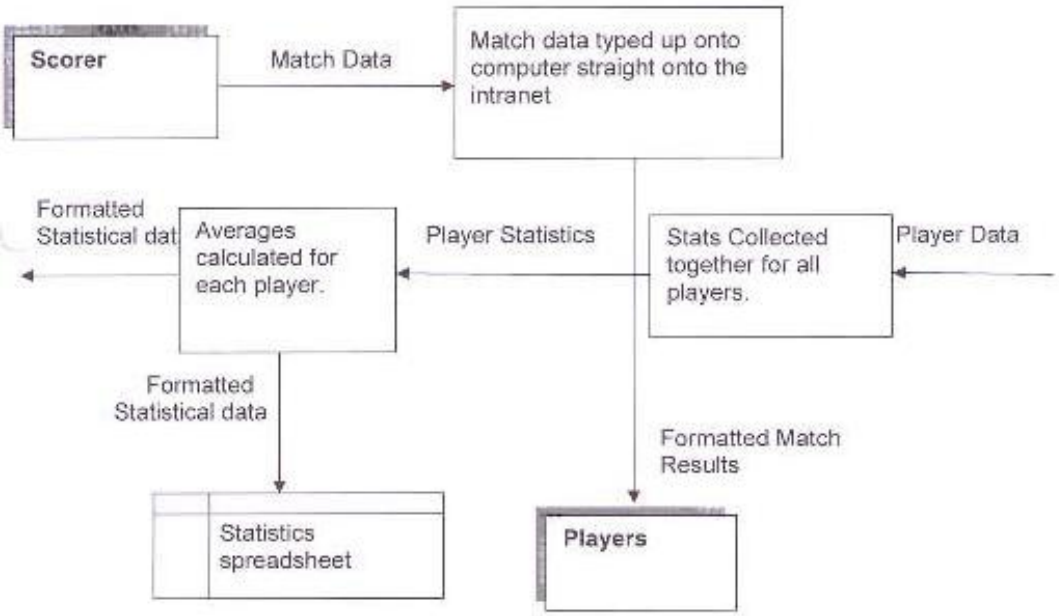
Square box = process

Sectioned box = data store

Main System



Results data processed



Origins and Forms of data

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

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

Fact Recording Justification

My main method of fact finding here has been the interviews, due to the fact that the cricket scorebook 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 s 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 reason: 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 batsman 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

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

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

Batting table:

15x11 batsman	165
Total Extras	3
Total runs/wickets	5
3x11 batsman totals	30
20x11 Wicket information	200
Overs bowled	2
Total bytes:	405

Bowling Table:

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

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 required. So the system will need 405 + 405 + 210 + 210 + 3300 + 3300 bytes or 8.43 KB

Assuming 12 matches a season are played, the storage of match data w amount considering most modern Laptops which this system will be use 60GD

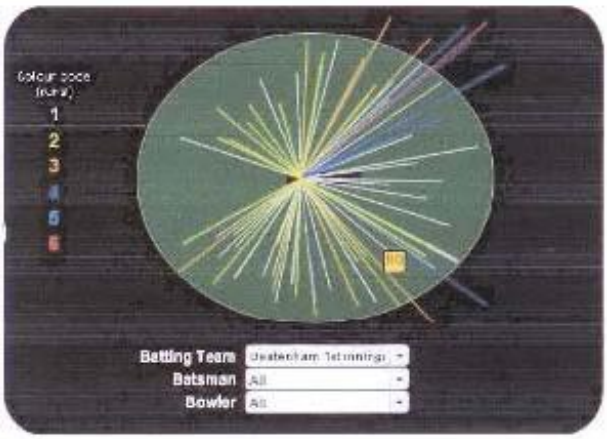
Alternative Approaches

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

There are other options which I have considered: one of these is off the shelf packages that already exist that use a spreadsheet such as spreadsheet based system. Although it follows all the requirements it w create and there are limitations to the interface due to the grid aspect of also contains features that are not needed for low standard cricket gam

Below is a screenshot of the L&W system, due to the many featu very compact with a lot of information being displayed on one window. A would be ideal, for a simple club game it is not needed and much of the So I intend to make an interface which is simpler and easier to use than such as this one, so that it requires less training to use and reduces the information about the match.

There are other off the shelf packages that could be used that fill all the requirement specification such as PitchPad, this system is far more complex then 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 wagon wheel diagrams 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 that 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

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

Section 2 – Design

Design Objectives

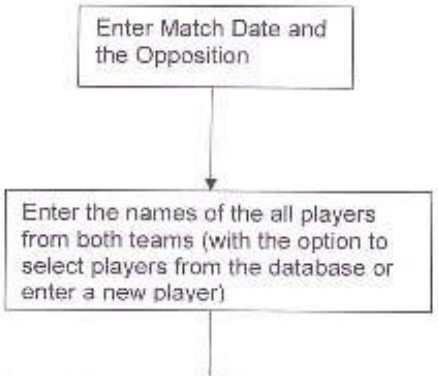
The system will need to achieve the following objectives:

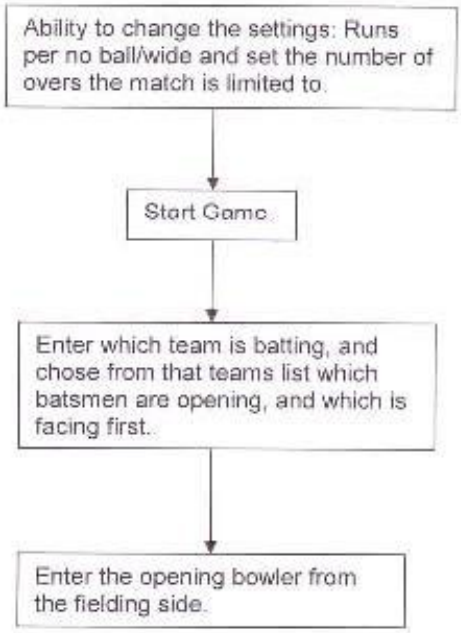
- On starting a game the user will be asked to fill in the names of all the players playing in the match, either by entering a new name, or choosing from the database.
- The user will be able to change features about the game prior to the start such as number of overs and number of runs for a wide.
- 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 batsman who is facing, the current bowler, 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:

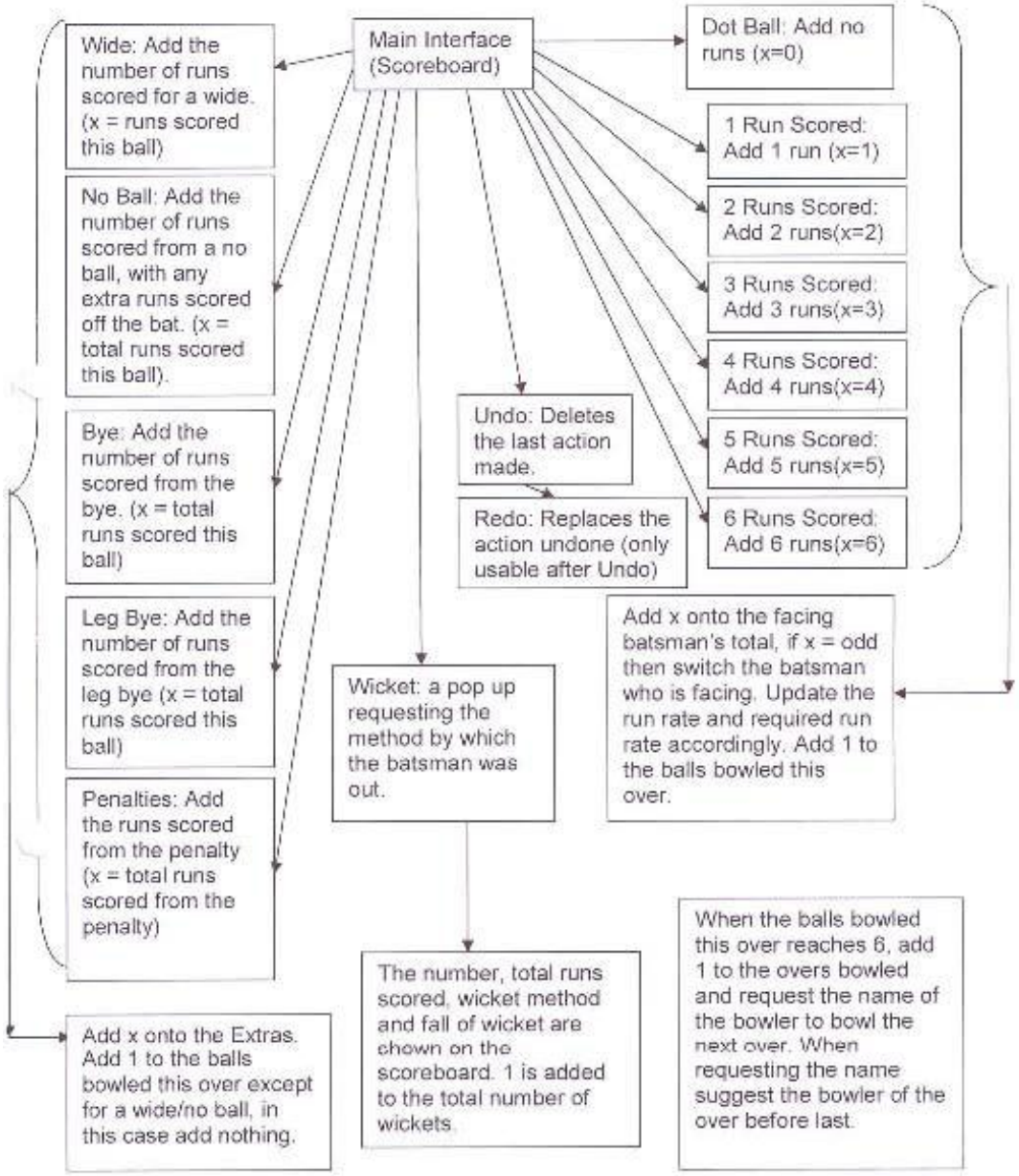
Input





Once this process has been done the game will start and the main scoreboard will be shown, the flow cha for this window is on the next page.

Input, Processing & Output

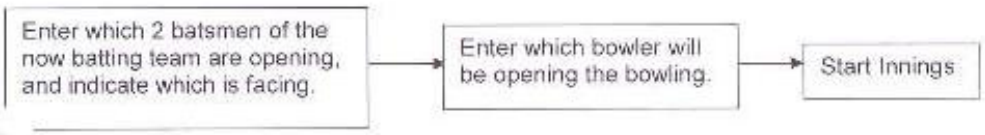


In addition, when the overs bowled reaches the number of limited overs or the number of wickets reaches 10 then the innings is over. The scorer gets a message saying the innings is complete; they can then close down the program to have a break and then re-open the program to carry on with the match when 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.

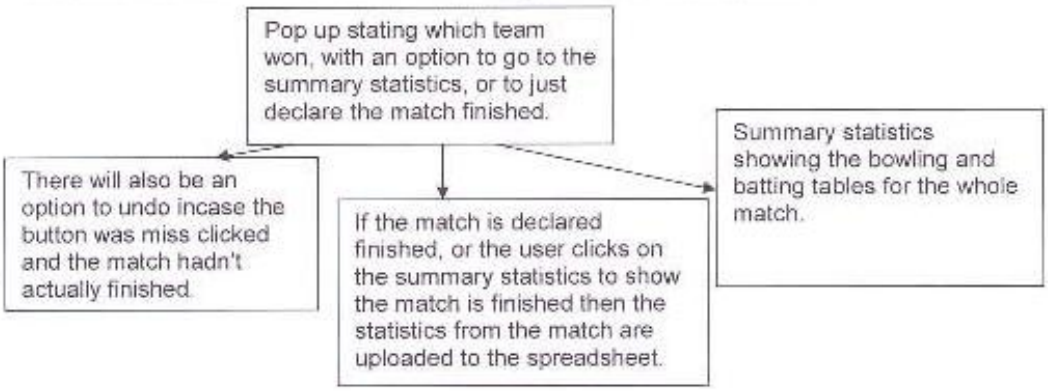
When the user indicates that the second innings of a match is about to start they will be prompted with the following flowchart:



Once the second innings has started the main scoreboard will display the following additional information:

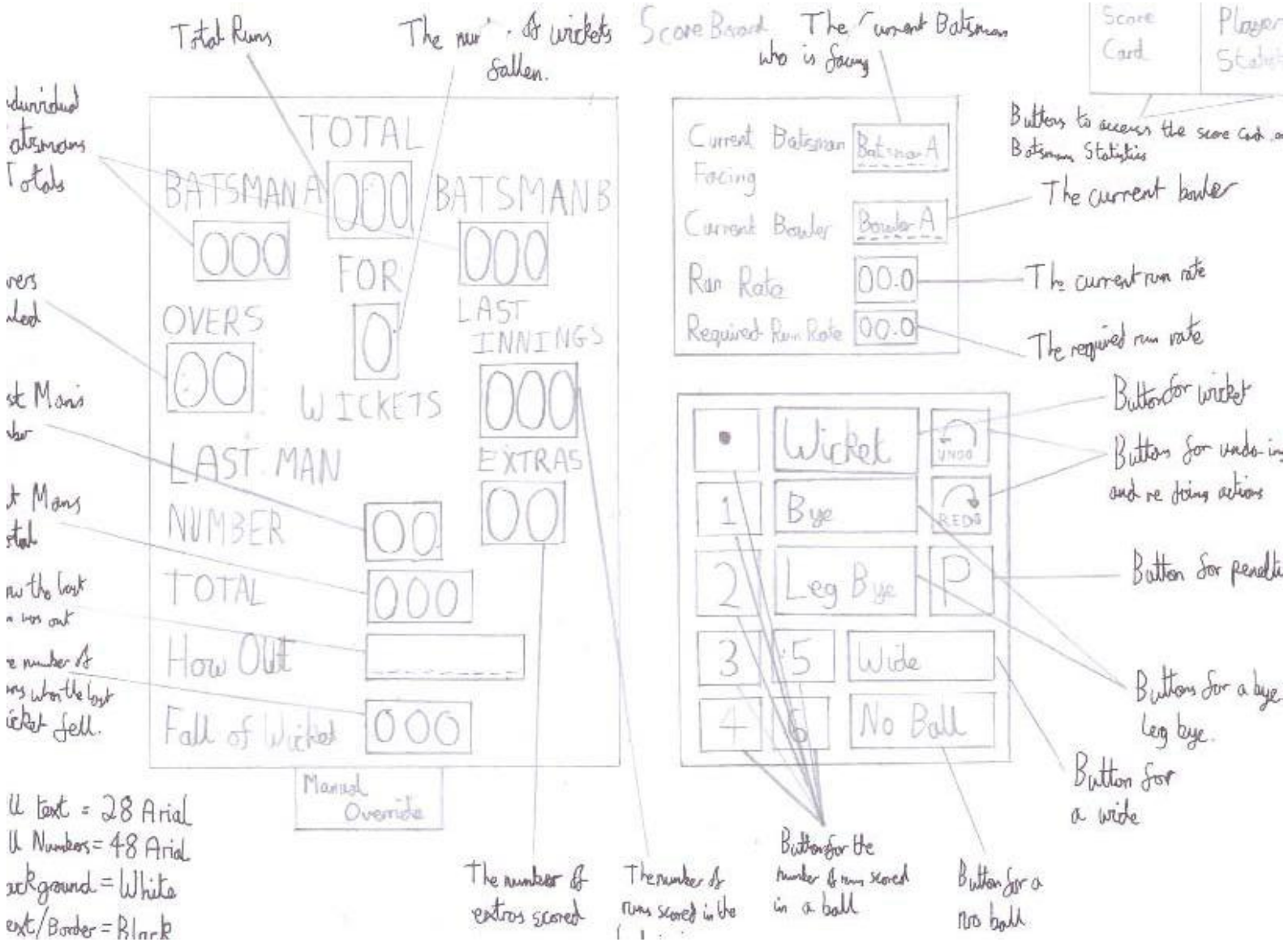
- The total runs scored in the first innings
- The run rate required by the batting team in the second innings

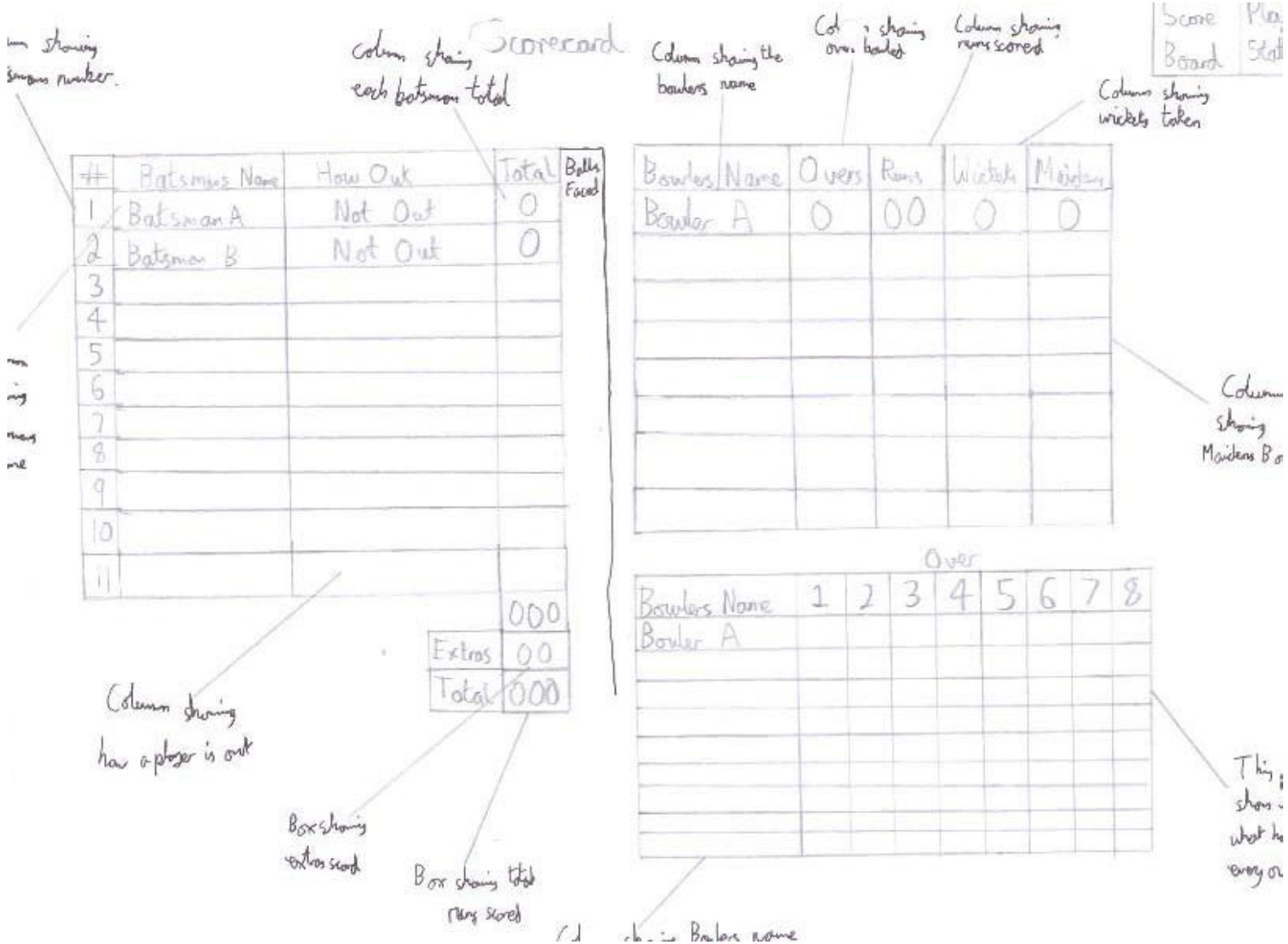
Once either team has won the match, the following flowchart will take place:

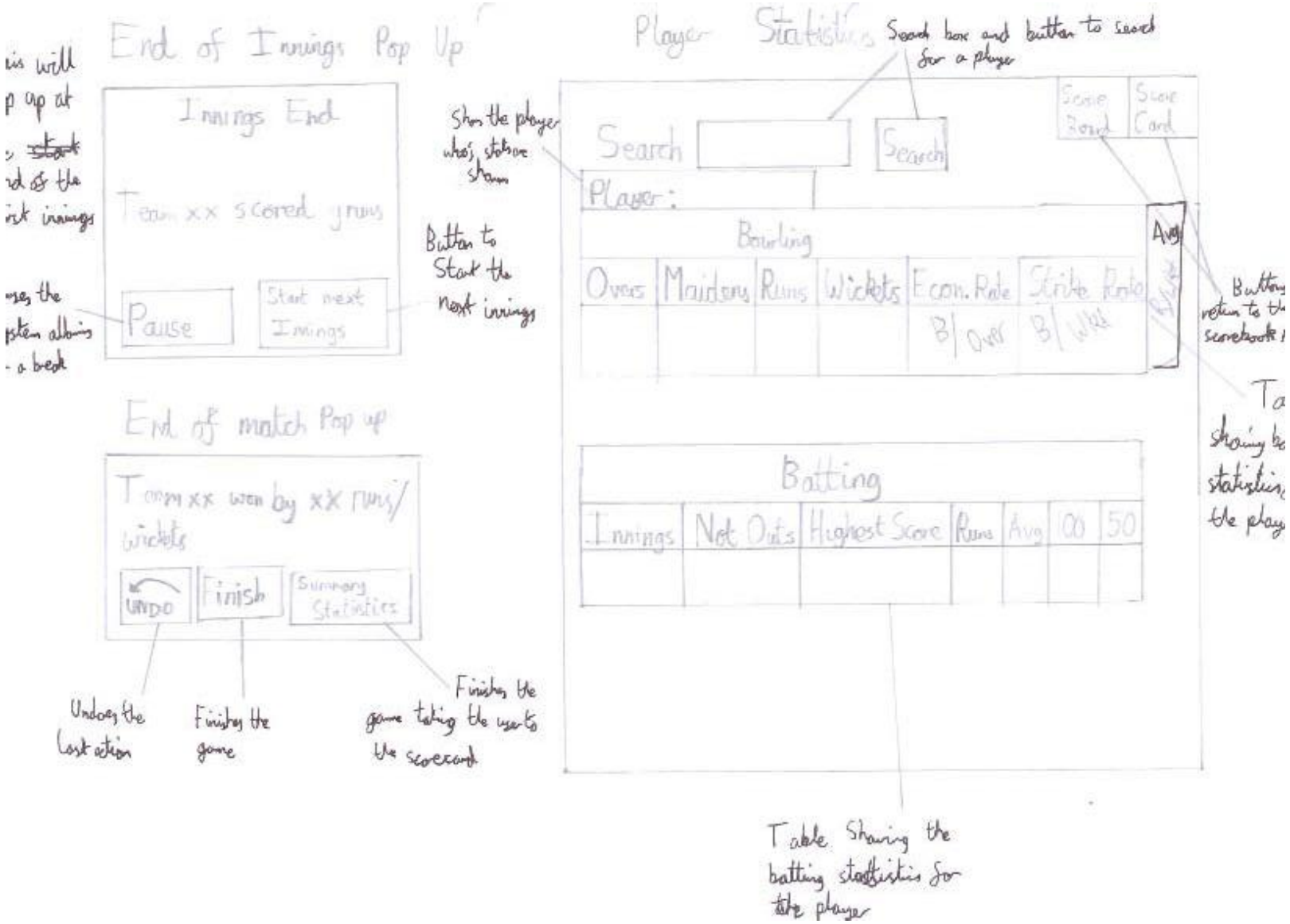


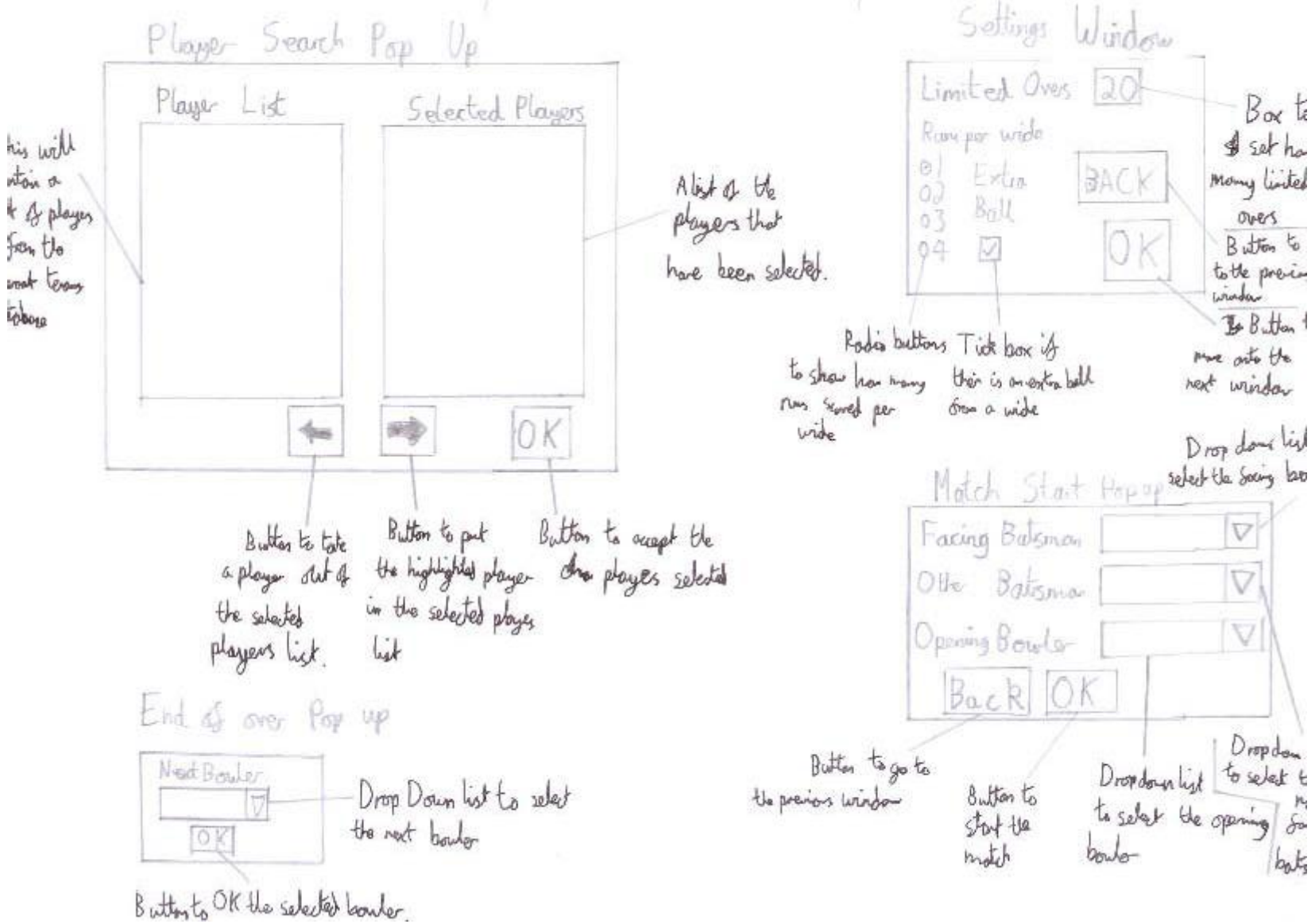
User Interface

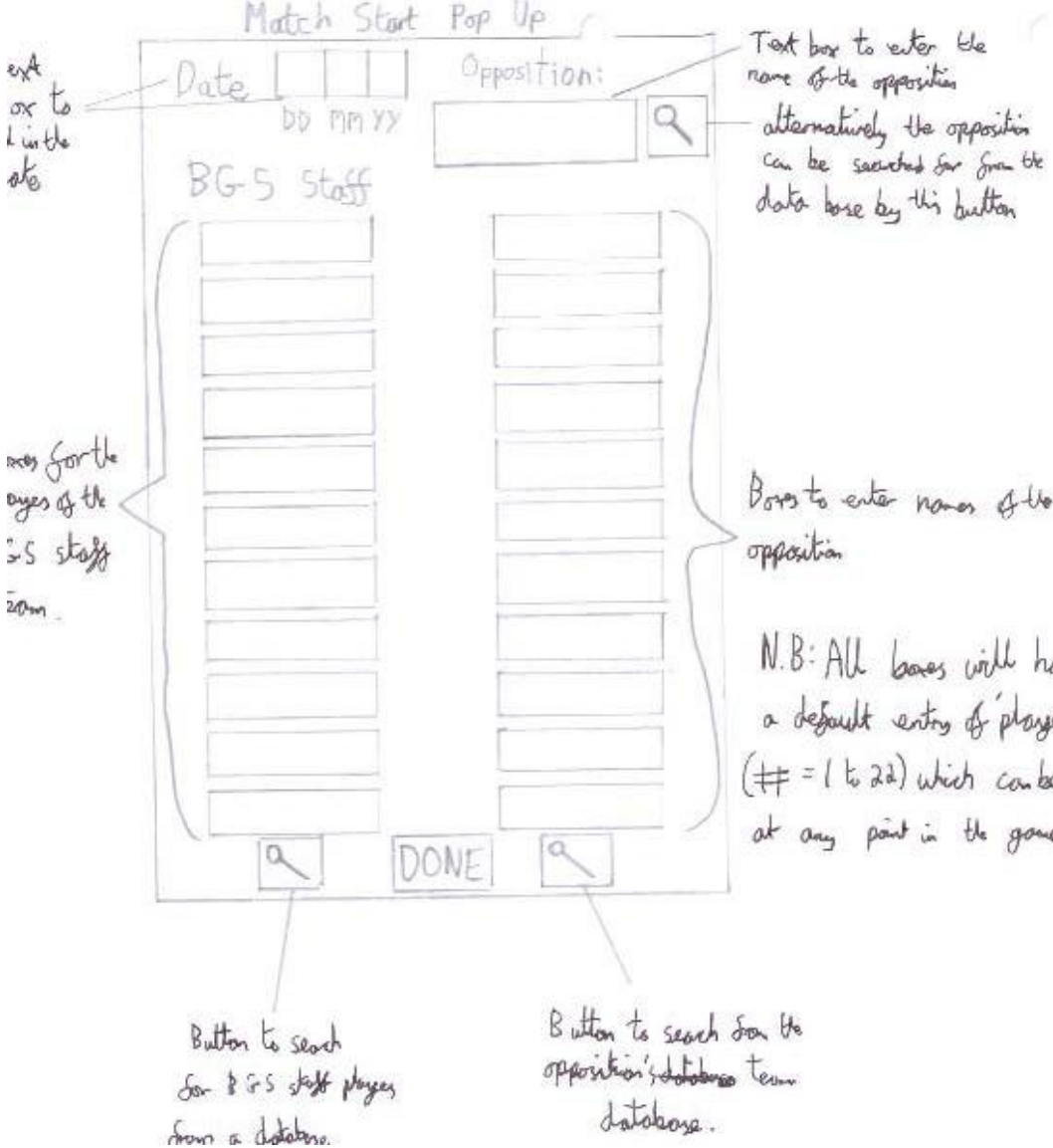
The next pages contain my hand drawn interface designs for the system.


Total Runs
The number of wickets fallen.
Score Board
The current Batsman who is facing
Score Card
Player Stats
Individual extras Totals
TOTAL
BATSMAN A
000
BATSMAN B
000
000
FOR
OVERS
0
LAST INNINGS
00
WICKETS
000
LAST MAN
EXTRAS
NUMBER
00
00
TOTAL
000
How Out
Fall of Wicket
000
Manual Override
Current Batsman Facing
Batsman A
Current Bowler
Bowler A
Run Rate
00.0
Required Run Rate
00.0
Buttons to access the score card & Batsman Statistics
The current bowler
The current run rate
The required run rate
Wicket
UNDO
1
Bye
REDO
2
Leg Bye
P
3
5
Wide
4
6
No Ball
Button for wicket
Buttons for undo-ing and re doing actions
Button for penalty
Buttons for a bye Leg bye.
Button for a wide
The number of extras scored
The number of runs scored in the
Button for the number of runs scored in a ball
Button for a no ball
ll text = 28 Arial
ll Numbers = 48 Arial
ckground = White
ext/Border = Black

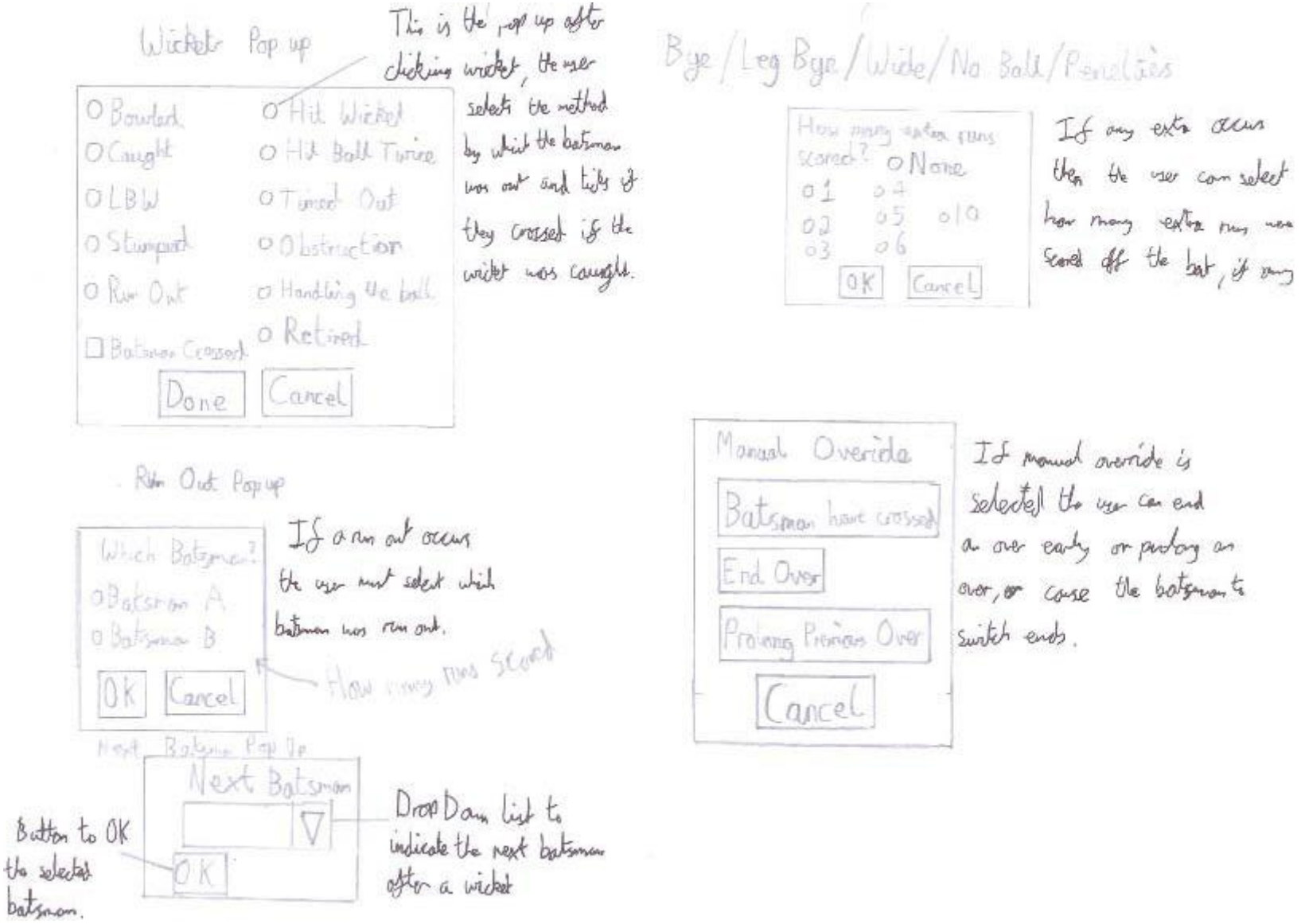

Scorecard
Column showing batsmans number.
Column showing each batsman total
Column showing the bowlers name
Col showing overs bowled
Column showing runs scored
Column showing wickets taken
Score Board
| Batsmans Name | How Out | Total | Balls Faced
1 | Batsman A | Not Out | 0
2 | Batsman B | Not Out | 0
3
4
5
6
7
8
9
10
11
000
Extras | 00
Total | 000
Bowlers Name | Overs | Runs | Wickets | Maidens
Bowler A | 0 | 00 | 0 | 0
Column showing Maidens Bo
Over
Bowlers Name | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
Bowler A
Column showing how a player is out
Box showing extras scored
Box showing total runs scored
Thi shows what ha every ov


End of Innings Pop Up
Innings End
Team xx scored yyyy runs
Pause
Start next Innings
Button to Start the next innings
End of match Pop up
Team xx won by xx runs/ wickets
UNDO
Finish
Summary Statistics
Undoes the last action
Finishes the game
Finishes the game taking the user to the scorecard
Player Statistics
Search box and button to search for a player
Shows the player who's statistics shown
Score Board
Score Card
Search
Search
Player:
Bowling
Overs
Maidens
Runs
Wickets
Econ. Rate
Strike Rate
Avg
B/ Over
B/ Wkt
Batting
Innings
Not Outs
Highest Score
Runs
Avg
00
50
Table Showing the batting statistics for the player


Player Search Pop Up
Player List
Selected Players
OK
A list of the players that have been selected.
Button to take a player out of the selected players list.
Button to put the highlighted player in the selected plays list
Button to accept the two players selected
Settings Window
Limited Overs
20
Runs per wide
01 Extra
02 Ball
03
04
BACK
OK
Radio buttons to show how many runs scored per wide
Tick box if their is an extra ball from a wide
Match Start Popup
Facing Batsman
Offe Batsman
Opening Bowler
Back
OK
Button to go to the previous window
Button to start the match
Dropdown list to select the opening bowler
End of over Pop up
Next Bowler
OK
Drop Down list to select the next bowler
Button to OK the selected bowler.



This is the error message which will occ where there is an error


Wicket Pop up
This is the pop up after clicking wicket, the user selects the method by which the batsman was out and ticks if they crossed if the wicket was caught.
Bowled
Caught
LBW
Stumped
Run Out
Batsman Crossed
Hit Wicket
Hit Ball Twice
Timed Out
Obstruction
Handling the ball
Retired
Done
Cancel
Bye / Leg Bye / Wide / No Ball / Penalties
How many extra runs scored?
None
1
2
3
4
5
6
10
OK
Cancel
If any extra occurs then the user can select how many extra runs were scored off the bat, if any
Run Out Pop up
Which Batsman?
Batsman A
Batsman B
OK
Cancel
If a run out occurs the user must select which batsman was run out.
How many runs scored
Manual Overide
Batsman have crossed
End Over
Proceed Previous Over
Cancel
If manual override is selected the user can end an over early or prolong an over, or cause the batsmen to switch ends.
Next Batsman Pop Up
Next Batsman
OK
Button to OK the selected batsman.
Drop Down list to indicate the next batsman after a wicket

Interview

After another interview with [redacted] 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 Batting 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

Field Name	Description of Contents	Data type	Length	Sample values / range / set	Type of validation expected
Date	The Date of the Match	Date	6	21/06/09	NA
Opposition	The Opposing team	Text	20	QEH Staff	NA
PlayerID	Primary Key	AutoNumber	3	21	Auto-Generated
PlayerName	The name of the player	Text	30	John Smith	NA
PlayerTeam	The players team	Text	20	BGS Staff	NA
Battingruns	The number of runs scored	Integer	3	58	NA
BattingBallsFaced	Number of balls faced	integer	3	105	NA
BattingHowOut	How the player was out whilst batting.	Text	10	Bowled	NA
BattingWktBowler	The name of the bowler who took the wicket	Text	30	John Smith	NA
Bowlingovers	The number of over bowled by a player	Integer	2	5	NA
Bowlingruns	The number of runs scored against a player	Integer	2	16	NA
Bowlingwkts	The number of wickets taken by a player	Integer	1	3	NA
Bowlingmaidens	The number of maidens bowled by a player	Integer	1	2	NA

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

File Name: [date]scorecard
Purpose: To provide a ball by ball description of the match

Fieldname	Description of contents	Data Type	Length	Sample values/range/set	Type of Validation
Ballno	The number of the ball	Integer	3	14	NA
Runs	Number of runs scored off the ball	Integer	1	2	NA
Batsman	The number of the batsman who scored the runs	Integer	2	8 (0 = neither batsman scored the runs)	NA
Bowler	The number of the bowler who bowled the ball	Integer	2	4	NA
Extra	The type of extra scored	Integer	1	0 = no extra, 1 = wide, 2 = no ball, 3 = bye, 4 = leg bye, 5 = penalties	NA
Wicket	The type of wicket off the ball	Integer	2	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	NA
Batsmanout	The number of the batsman who was out from the wicket	Integer	2	5 (0 = no batsman out)	NA

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 6 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.

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

Size of Individual Record: 80
Maximum Records Stored: 1,000
Total size required for data: 80,000
Overheads (10%): 88,000
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.

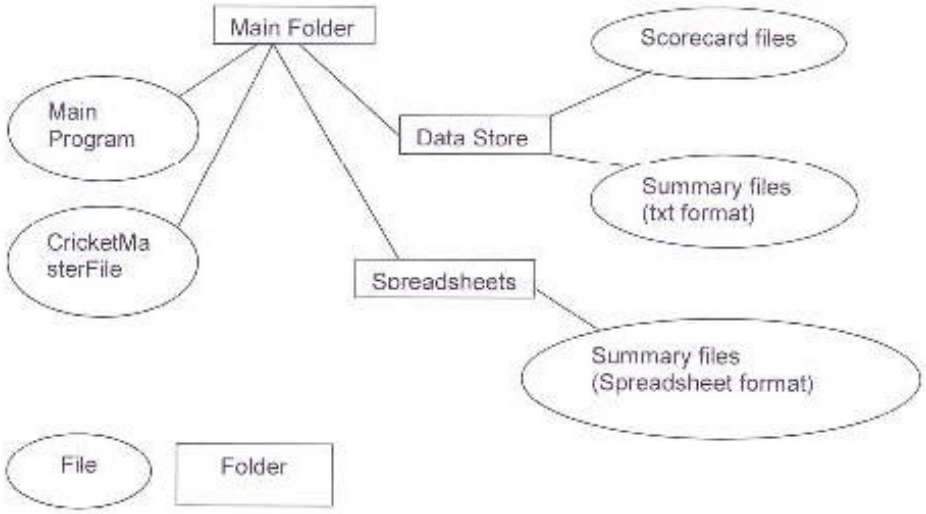
Data Structure

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

Field Name	Description	Data Type	Length	Sample Value
TotalRuns	The total runs scored this innings	Integer	3	123
TotalWkts	The total wickets this innings	Integer	1	4
BatAtotal	Runs scored by Batsman A	Integer	3	54
BatBtotal	Runs scored by Batsman B	Integer	3	14
Overs	Total Overs bowled	Integer	2	15
LastInns	Runs scored last innings	Integer	3	160
Extras	Total extras scored	Integer	2	17
LMruns	Runs scored by the last man in	Integer	3	103
LMnum	Number of the last man in	Integer	2	6
LMout	How the last man in was out	Text	17	Bowled
LMFoW	Number of runs when last wicket fell	Integer	3	23
CurBat	Name of the current batsman facing	Text	30	Smith
CurBowl	Name of the current bowler	Text	30	Jones
RunRate	Current run rate	Single	4	4.26
RqRunRate	Required run rate to win	Single	4	8.13
Bowled	Number of balls bowled in the over	Integer	1	2

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:



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:

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

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 file sizes, using the assumption of 12 matches per year this would be the expected file size required:

	Year 1		Year 2		Year 3	
Filename	No of Records	File Size	No of Records	File Size	No of Records	File Size
CricketMasterFile	1,000	88,000	1,000	88,000	1,000	88,000
Summary	264	32,232	528	64,464	792	96,696
Statistics	24,000	315,120	48,000	630,240	72,000	945,360

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.

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

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 Wadey, Andy Keen, Andy Barker, Graham Clark, Justin Harford** (chosen from the database) **Ben Scott, Kevin Blackburn, Andrew Flintoff, Kevin Peterson, Steve Harmason** (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**
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(6), 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, Batsman 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 Wadey), 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 5 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 batsman: **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(Handling 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

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

			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.		
20	Third Over data Inputted	Bye(1), Bye(6), Leg Bye(1), Leg Bye(6), Penalty(1), Penalty(6)	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	SS 23 – 28	The Penalty window did not have the None button greyed out.
22	Fourth Over data inputted	Wide(0), No ball(0)	Each of these balls should add 1 to the total and 1 to the extras	SS 29 – 31	As expected
22	Fourth Over data Continued	2 Runs [Undo]	The 2 runs added should be taken away from the total and the batsman scores	SS 32 – 34	As Expected
22	Fourth Over data continued	[Batsman Swap Ends]	Swap the facing batsman without changing anything else.	SS 35 & SS 36	As Expected
24	Fifth Over: Batsman bowled out	Wicket(Bowled), next batsman Colin Wadey selected	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.	SS 39 - 41	As Expected
24	Fifth Over: 2 more batsman out	Wicket(Caught) and Wicket(LBW)	The same process should occur for the wickets, with the Last man information being updated correctly with the type of wicket		As Expected
24	Fifth Over: Check Scorecard	Click the Scorecard button	See Below	SS 45A & 45	At first the scorecard bugged as shown in SS 45A. but was then fixed in SS 45

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:

Batsman	How Out	Bowler	Total	Balls Faced
Ian Rolling	Bowled	Opp Five	13	15
Andy Barker	Not	Out	25	24
Colin Wadey	Caught	Opp Five	0	1
Andy Keen	LBW	Opp Five	0	1
Graham Clark	Not	Out	3	1

Bowler	Overs	Runs	Wickets	Maidens
Opp one	1	8	3	1
Opp two	1	12	0	0
Opp three	1	21	0	0
New Opp eight	1	0	0	1

Bowler	1	2	3	4	5
Opp one	0 0 0 0 0 0	W W W 3 5			
Opp two	+ + 0 +6 0 0 6 4 0				
Opp three	1 2 3 4 5 6				
New Opp eight	0 0 0 0 0 0				

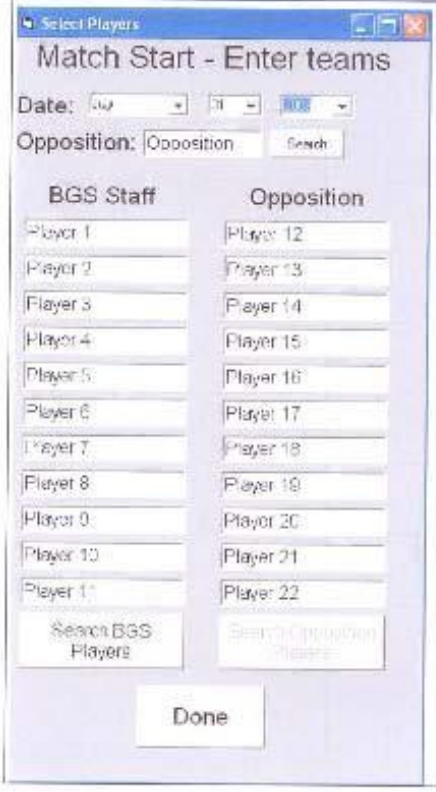
Test #	Test name	Input Data	Expected Result	Output	Actual Result
27	Innings finishes	The last ball of the innings	Pop up saying the innings has ended, followed by player selection for the next innings	SS 44	As expected
32	Start second innings	Confirm the starting players	The scoreboard comes up with a start next innings button	SS 46	As Expected
33	Over 1 data: Run Out	Wicket: Run Out	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.	SS 47	As Expected
35	Over 2 data	Various different wicket data is entered, the over is also prolonged	All the different wickets should update the scoreboard correctly. When the over finishes the prolong option should allow another ball to take place.		As Expected
37	Over 3 data, end of match	The final wicket of the match is inputted	A message box should come up displaying the winning team and then showing the scorecard for the second innings. The spreadsheets showing match data are created.	SS 49 (summary file), SS 50 (master file)	As Expected

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 rema possible batsman rather than the whole team.

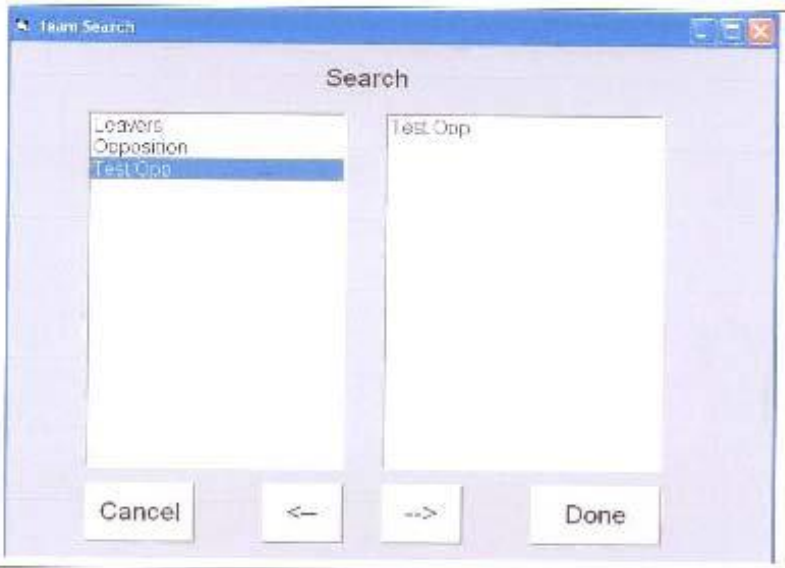
1

Press Start Game to start a new game
Start Game

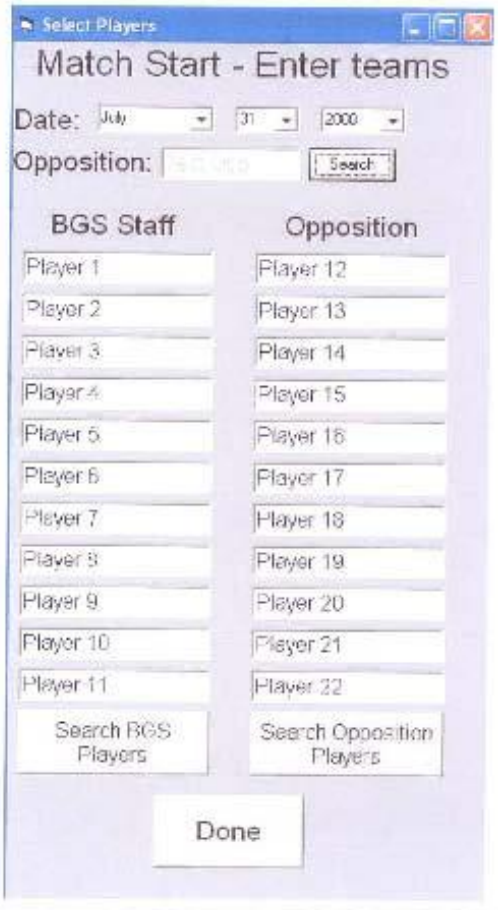
2


Select Players
Match Start - Enter teams
Date:
Opposition:
Opposition
Search
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
Player 12
Player 13
Player 14
Player 15
Player 16
Player 17
Player 18
Player 19
Player 20
Player 21
Player 22
Search BGS Players
Done

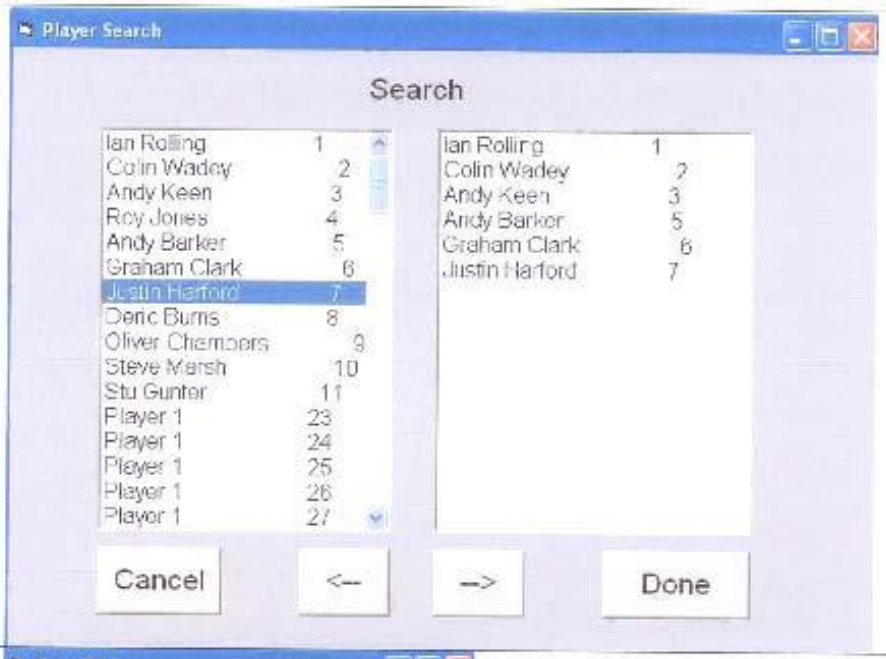
3


Team Search
Search
Leavers
Opposition
Test Opp
Test Opp
Cancel
<--
-->
Done

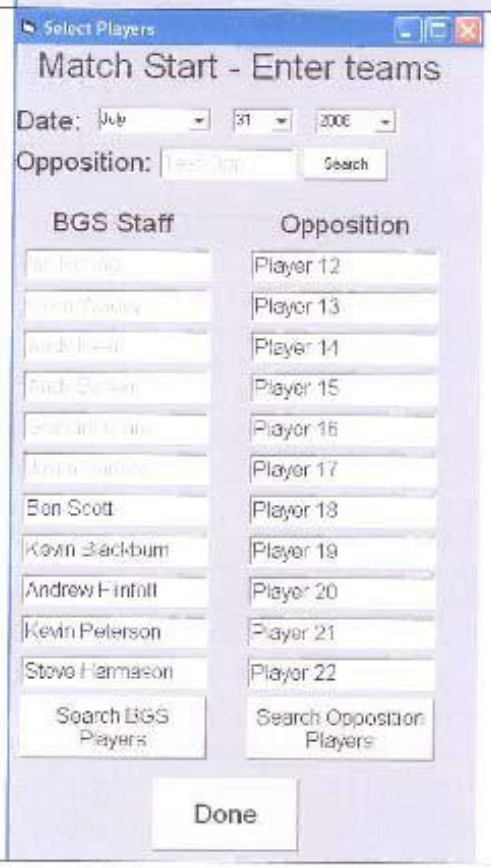
4


Select Players
Match Start - Enter teams
Date:
July
31
2000
Opposition:
Search
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
Player 12
Player 13
Player 14
Player 15
Player 16
Player 17
Player 18
Player 19
Player 20
Player 21
Player 22
Search BGS Players
Search Opposition Players
Done

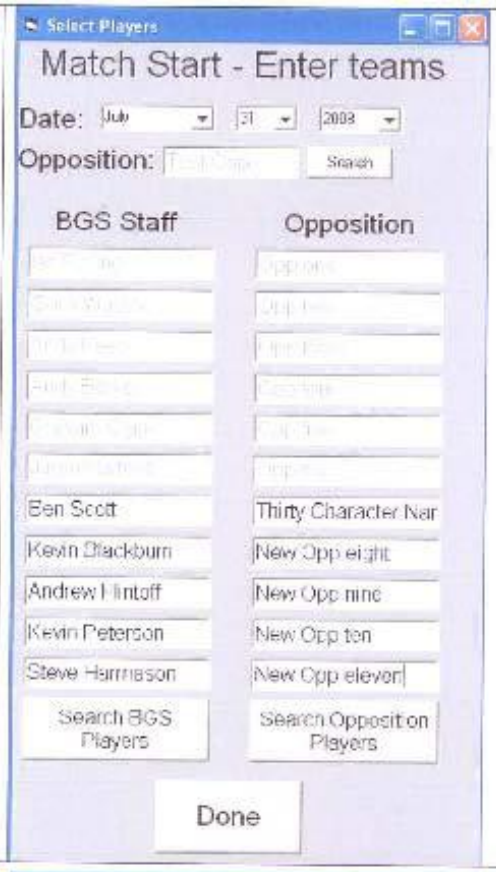
5


Player Search
Search
Ian Rolling 1
Colin Wadey 2
Andy Keen 3
Roy Jones 4
Andy Barker 5
Graham Clark 6
Justin Harford 7
Deric Burns 8
Oliver Chambers 9
Steve Marsh 10
Stu Gunter 11
Player 1 23
Player 1 24
Player 1 25
Player 1 26
Player 1 27
Ian Rolling 1
Colin Wadey 2
Andy Keen 3
Andy Barker 5
Graham Clark 6
Justin Harford 7
Cancel
<--
-->
Done

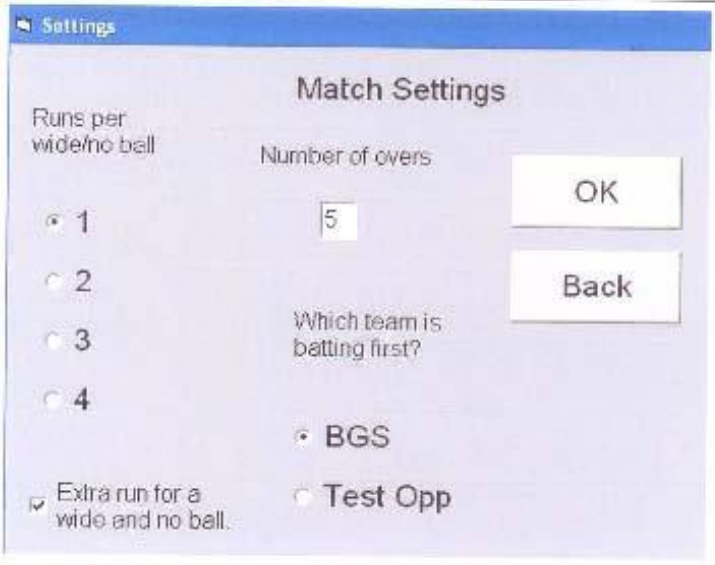
6


Select Players
Match Start - Enter teams
Date: July 31 2008
Opposition:
Search
BGS Staff
Opposition
Player 12
Player 13
Player 14
Player 15
Player 16
Player 17
Ben Scott
Player 18
Kevin Sackburn
Player 19
Andrew Flintoff
Player 20
Kevin Peterson
Player 21
Steve Harmason
Player 22
Search BGS Players
Search Opposition Players
Done

7


Select Players
Match Start - Enter teams
Date:
July
31
2008
Opposition:
Search
BGS Staff
Opposition
Ben Scott
Kevin Blackburn
Andrew Flintoff
Kevin Peterson
Steve Harmison
Thirty Character Nar
New Opp eight
New Opp nine
New Opp ten
New Opp eleven
Search BGS Players
Search Opposition Players
Done

8


Settings
Match Settings
Runs per wide/no ball
1
2
3
4
Extra run for a wide and no ball.
Number of overs
5
Which team is batting first?
BGS
Test Opp
OK
Back

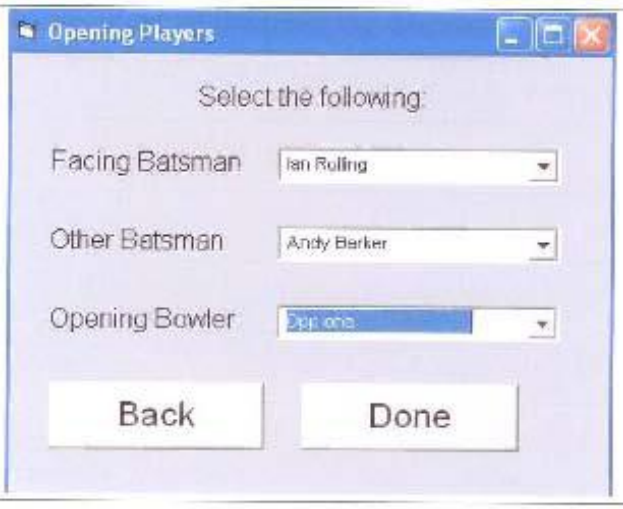
9

Opening Players
Select the following:
Facing Batsman
Ian Rolling
Other Batsman
Andy Barker
Opening Bowler
Colin Wesley
Andy Keen
Andy Barker
Graham Clark
Justin Harford
Ben Scull
Kevin Blackburn
Andrew Flintoff
Back

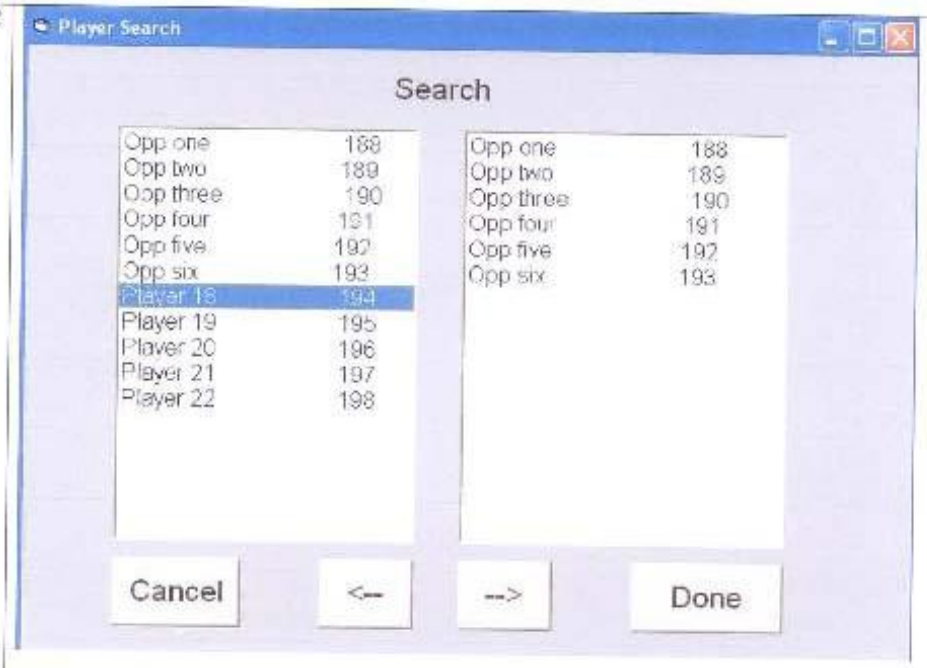
10

Opening Players
Select the following:
Facing Batsman
Ian Rolling
Other Batsman
Andy Barker
Opening Bowler
Opp one
Opp one
Opp two
Opp three
Opp four
Opp five
Opp six
Thirty Character Name For Test
New Opp right
Back

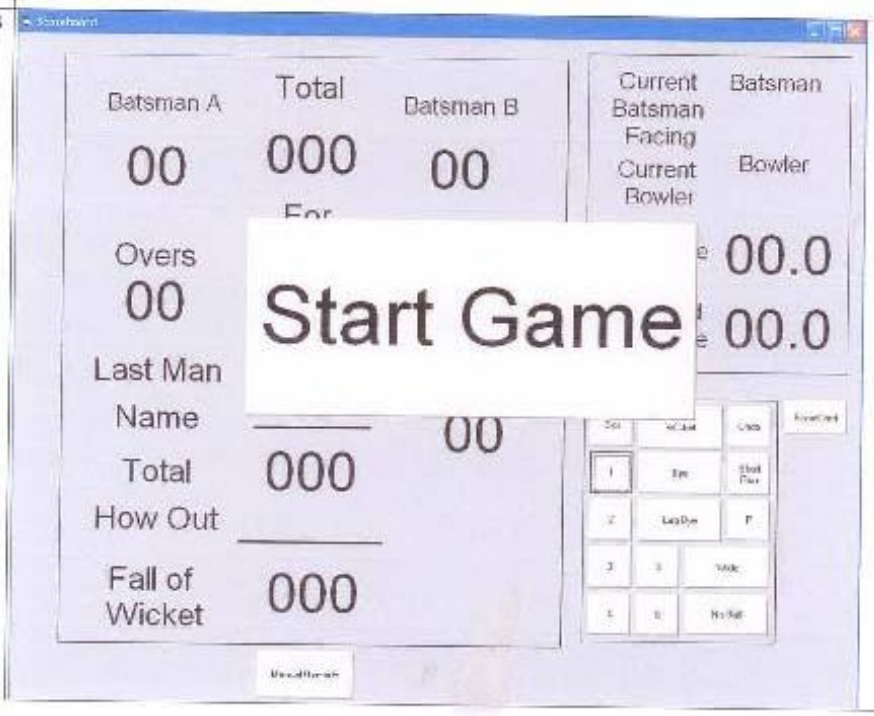
11


Opening Players
Select the following:
Facing Batsman
Ian Rolling
Other Batsman
Andy Barker
Opening Bowler
Opp one
Back
Done

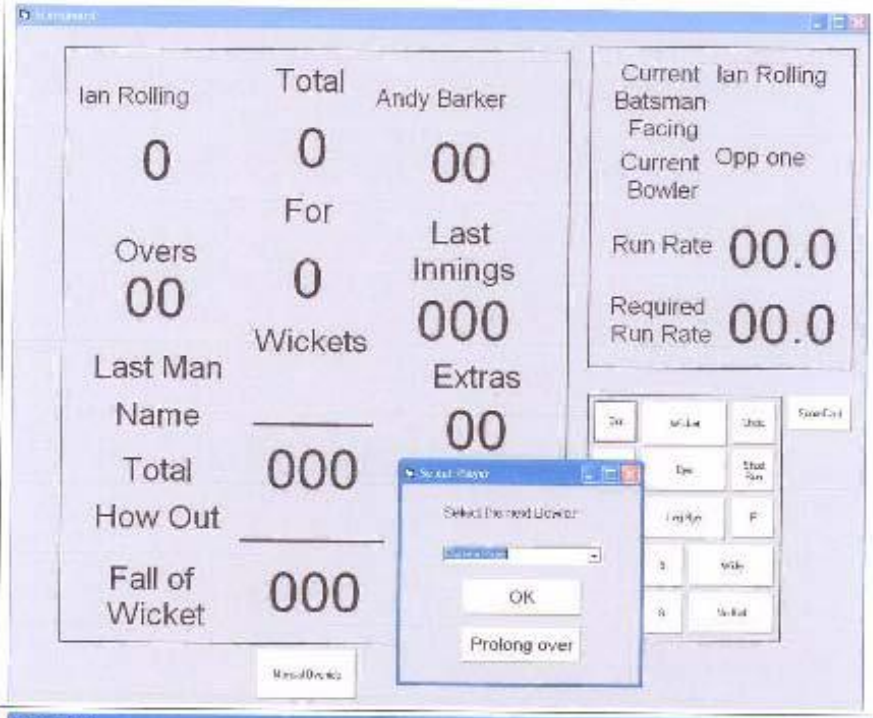
12


Player Search
Search
Opp one 188
Opp two 189
Opp three 190
Opp four 191
Opp five 192
Opp six 193
Player 18 194
Player 19 195
Player 20 196
Player 21 197
Player 22 198
Opp one 188
Opp two 189
Opp three 190
Opp four 191
Opp five 192
Opp six 193
Cancel
<--
-->
Done

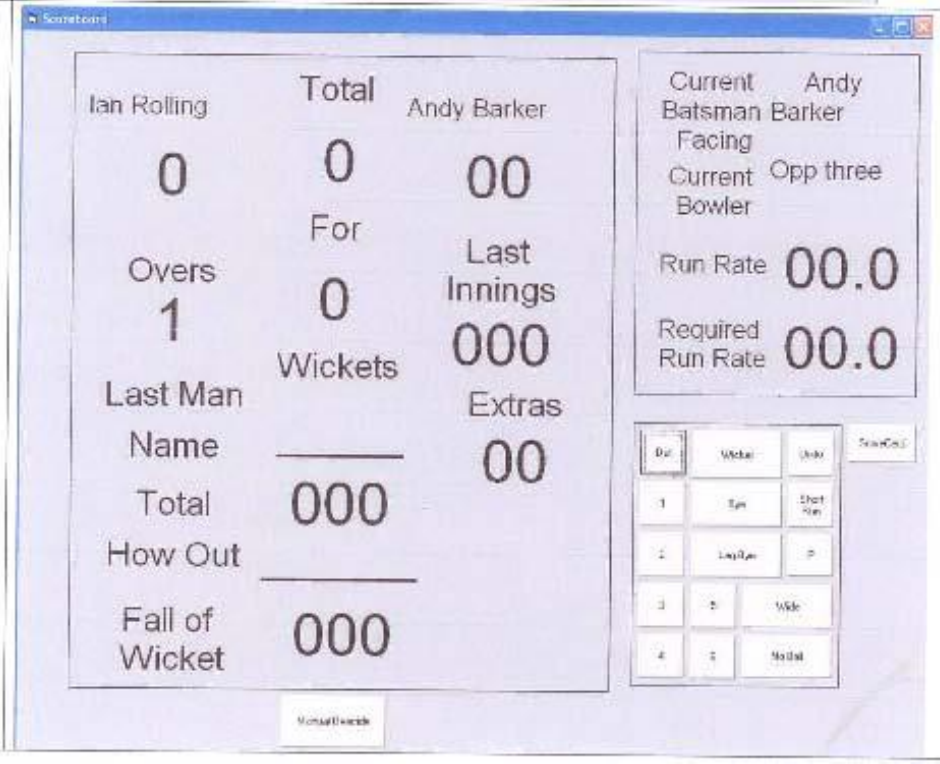
13


Batsman A
Total
Batsman B
00
000
00
Current Batsman Facing
Batsman
Current Bowler
Bowler
Overs
00
00.0
00.0
Start Game
Last Man
Name
Total
000
How Out
Fall of Wicket
000

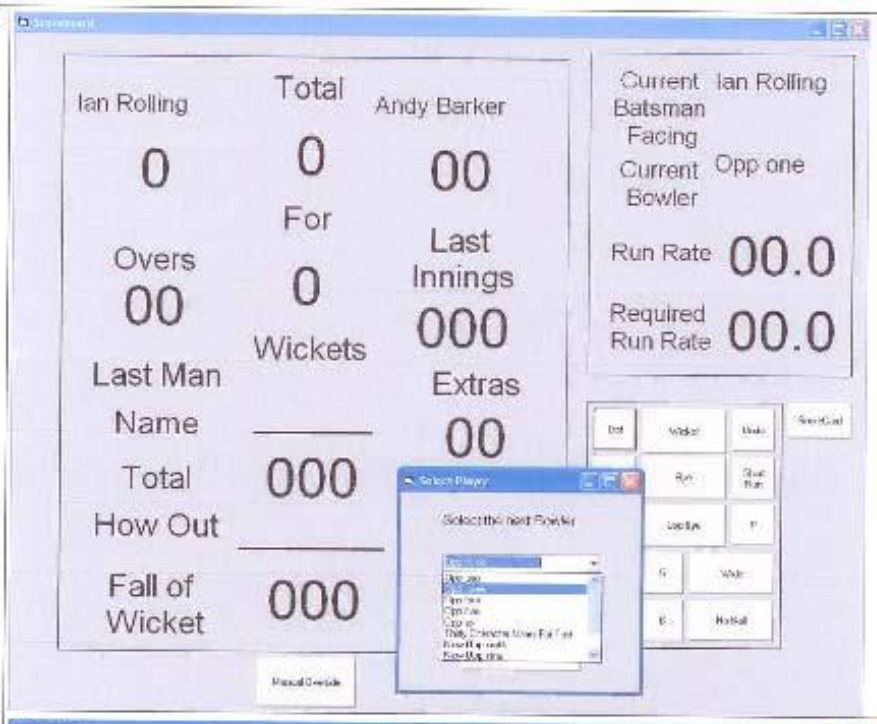
14


Ian Rolling
Total
Andy Barker
0
0
00
For
Overs
00
0
Last Innings
000
Wickets
Last Man
Name
Extras
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Ian Rolling
Current Bowler
Opp one
Run Rate
00.0
Required Run Rate
00.0
OK
Prolong over

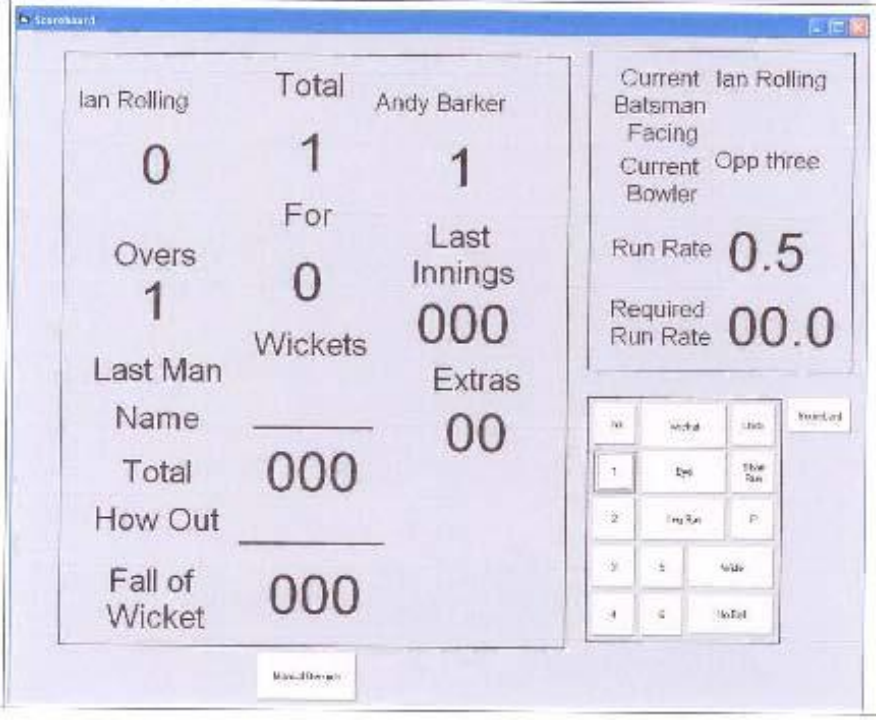
15


Ian Rolling
Total
Andy Barker
0
0
00
For
Overs
1
0
Last Innings
000
Wickets
Last Man
Name
Extras
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Andy Barker
Current Bowler
Opp three
Run Rate
00.0
Required Run Rate
00.0

16

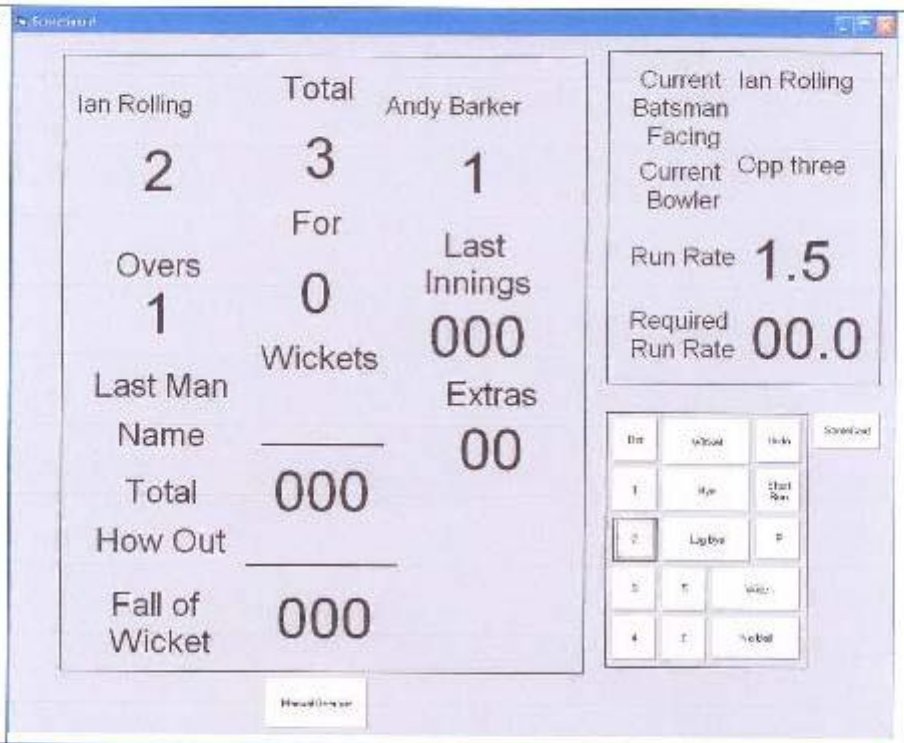

Ian Rolling
Total
Andy Barker
0
0
00
For
Overs
00
0
Last Innings
000
Wickets
Last Man
Extras
Name
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Ian Rolling
Current Bowler
Opp one
Run Rate
00.0
Required Run Rate
00.0
Select Player
Select the next Bowler

17

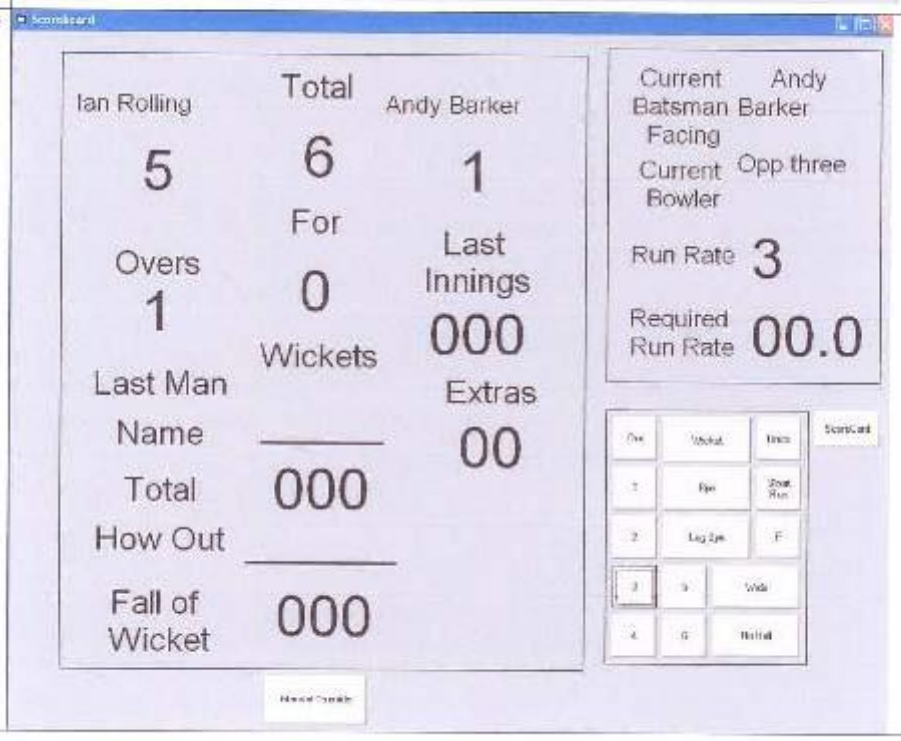

Ian Rolling
Total
Andy Barker
0
1
1
For
Overs
1
0
Last Innings
000
Wickets
Last Man
Extras
Name
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Ian Rolling
Current Bowler
Opp three
Run Rate
0.5
Required Run Rate
00.0

E

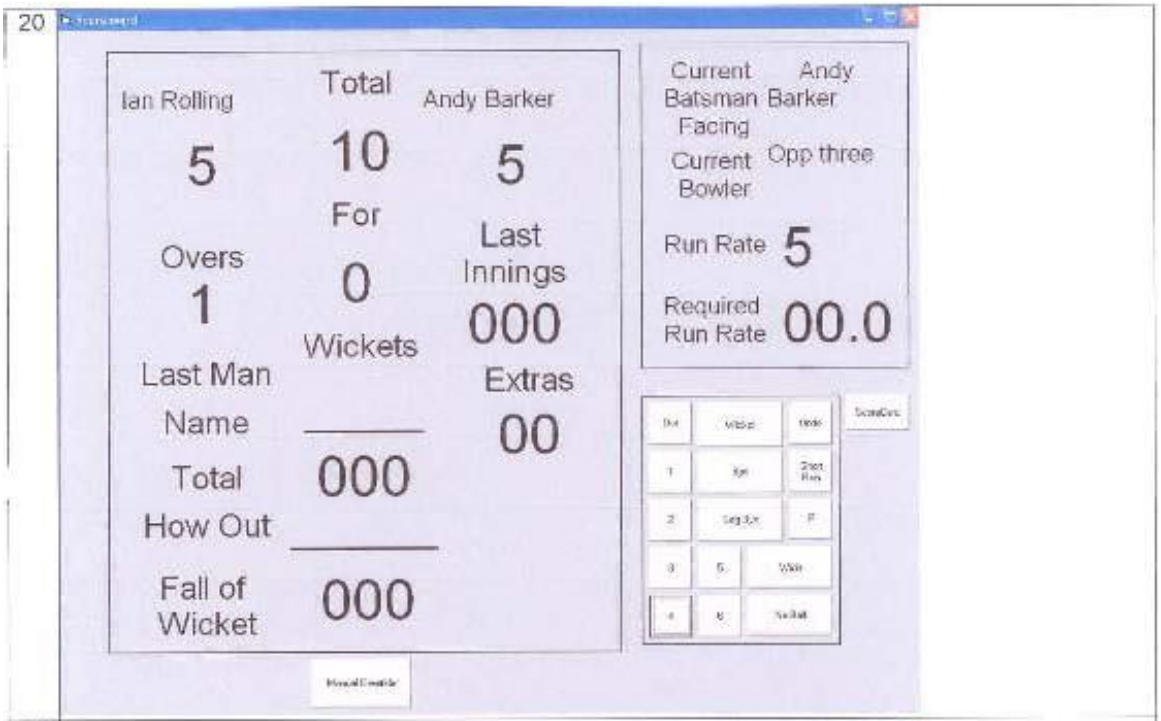
18


Ian Rolling
Total
Andy Barker
2
3
1
For
Overs
1
0
Last Innings
000
Wickets
Last Man
Extras
Name
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Ian Rolling
Current Bowler
Opp three
Run Rate
1.5
Required Run Rate
00.0

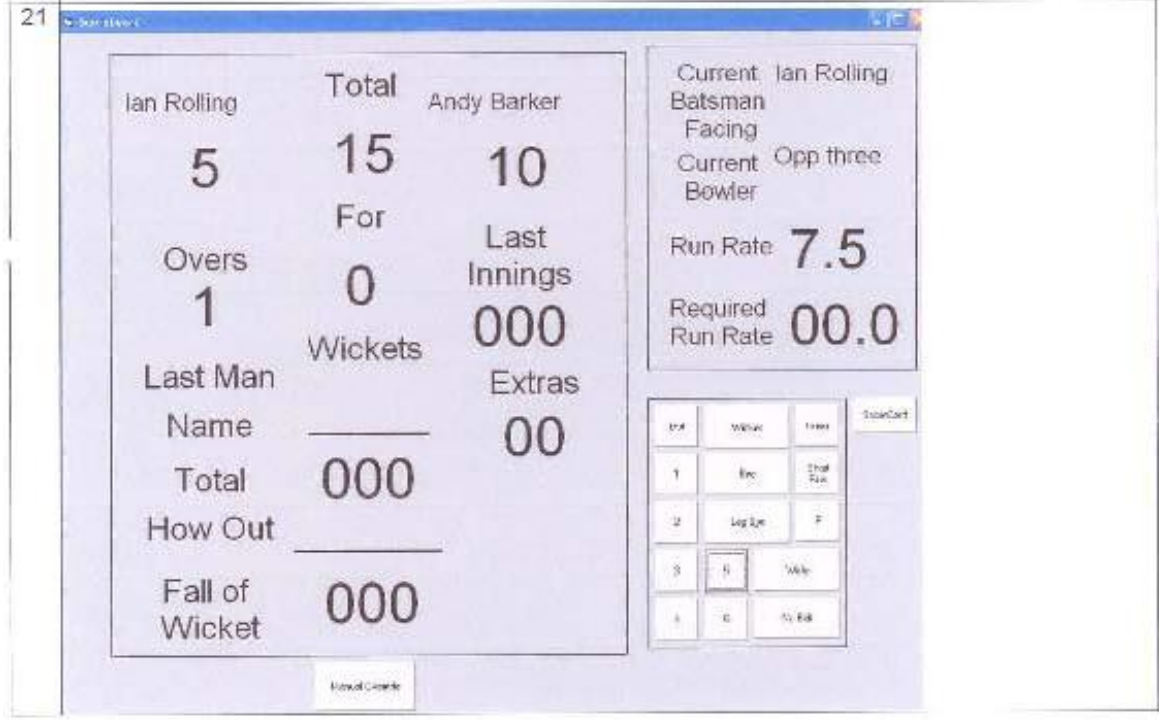
19


Ian Rolling
Total
Andy Barker
5
6
1
For
Overs
1
0
Last Innings
000
Wickets
Last Man
Extras
Name
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Andy Barker
Current Bowler
Opp three
Run Rate
3
Required Run Rate
00.0

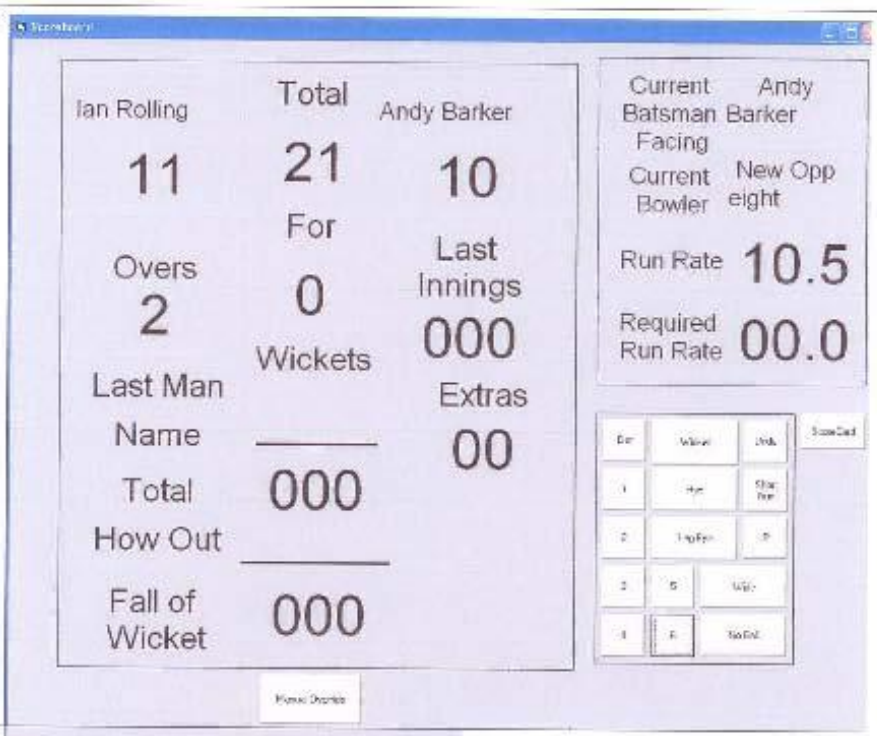
20


Ian Rolling
Total
Andy Barker
5
10
5
For
Overs
Last Innings
1
0
000
Wickets
Last Man
Extras
Name
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Andy Barker
Current Bowler
Opp three
Run Rate
5
Required Run Rate
00.0

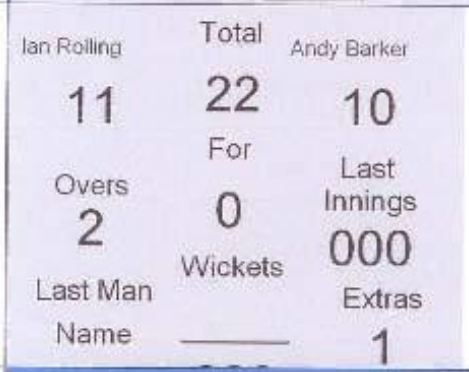
21


Ian Rolling
Total
Andy Barker
5
15
10
For
Overs
Last Innings
1
0
000
Wickets
Last Man
Extras
Name
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Ian Rolling
Current Bowler
Opp three
Run Rate
7.5
Required Run Rate
00.0

22


Ian Rolling
Total
Andy Barker
11
21
10
For
Overs
Last Innings
2
0
000
Wickets
Last Man
Extras
Name
00
Total
000
How Out
Fall of Wicket
000
Current Batsman Facing
Andy Barker
Current Bowler
New Opp eight
Run Rate
10.5
Required Run Rate
00.0

23


Ian Rolling
Total
Andy Barker
11
22
10
For
Overs
Last Innings
2
0
000
Wickets
Last Man
Extras
Name
1

24

Extra Scored
How many runs were scored from the bye?
4
2
5
3
6
Cancel
Ok

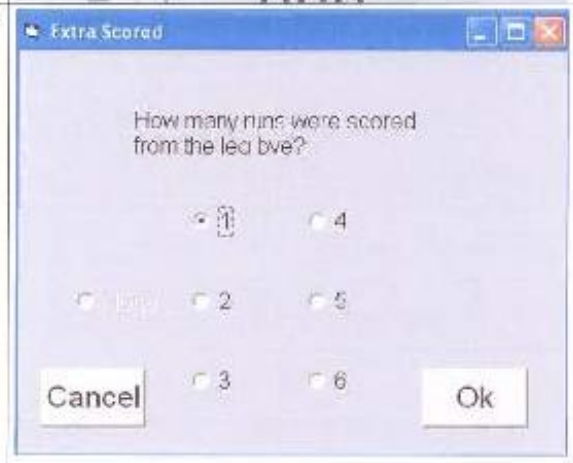
25

Extra Scored
How many runs were scored from the bye?
1
4
2
5
3
6
Cancel
Ok

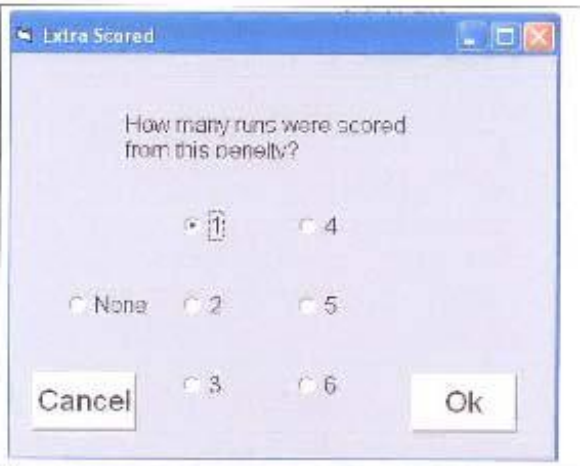
26

Ian Rolling
11
Overs
2
Last Man
Name
Total
28
For
0
Wickets
Andy Barker
10
Last
Innings
000
Extras
7

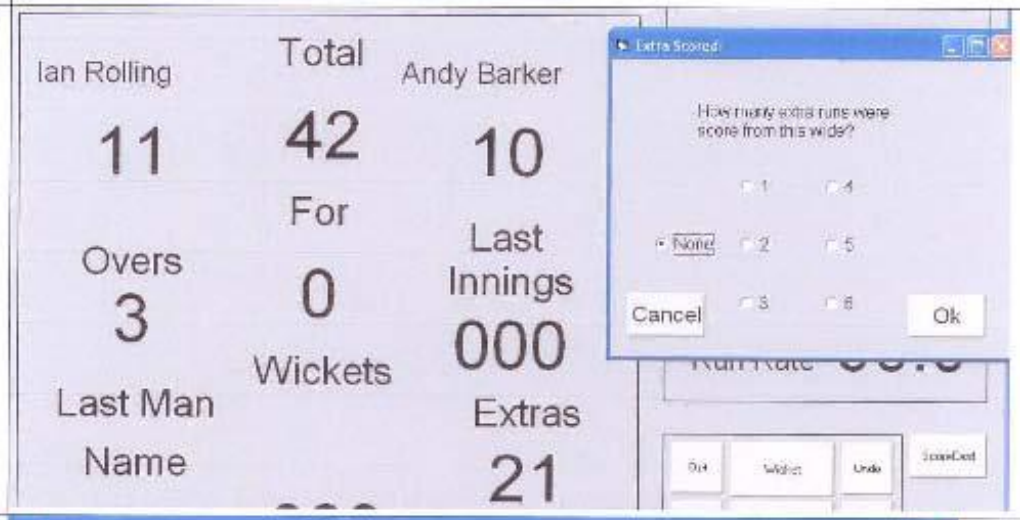
27


Extra Scored
How many runs were scored from the leg bye?
1
4
2
5
3
6
Cancel
Ok

28


Extra Scored
How many runs were scored from this penalty?
1
4
None
2
5
3
6
Cancel
Ok

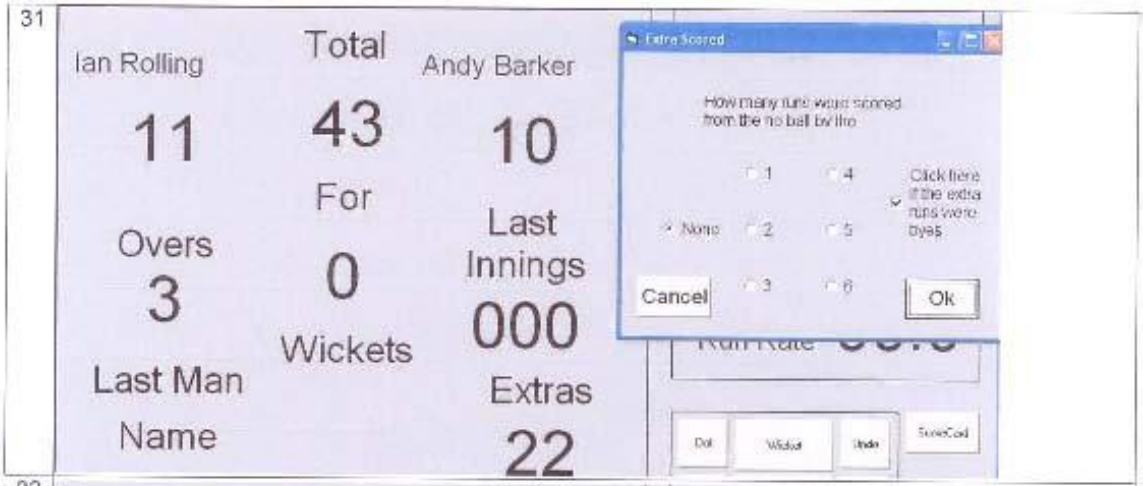
29


Ian Rolling
Total
Andy Barker
11
42
10
Overs
For
Last Innings
3
0
000
Wickets
Last Man Name
Extras
21
Extra Scored
How many extra runs were score from this wide?
None
Cancel
Ok

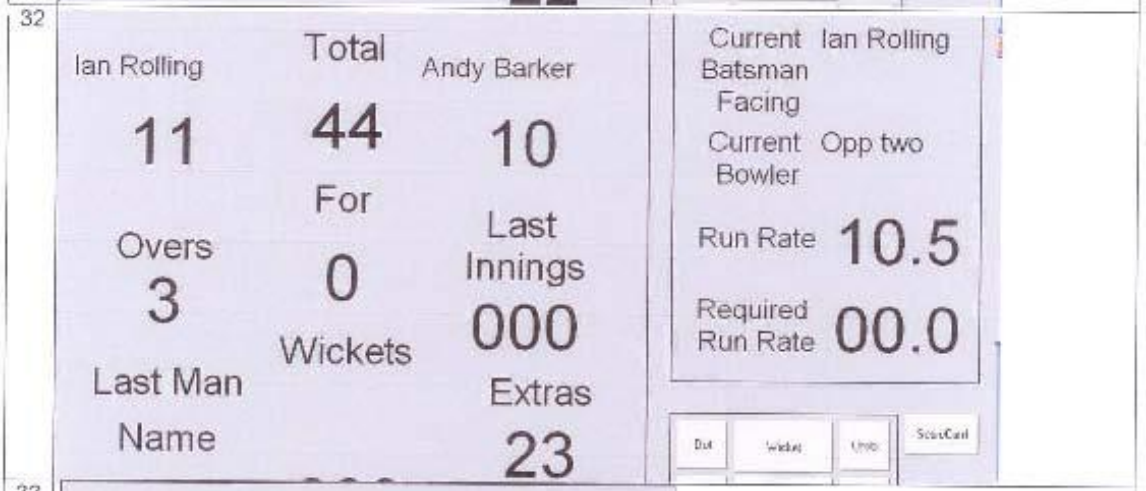
30

Ian Rolling
Total
Andy Barker
11
43
10
Overs
For
Last Innings
3
0
000
Wickets
Last Man Name
Extras
22
Current Batsman Facing
Ian Rolling
Current Bowler
Opp two
Run Rate
10.5
Required Run Rate
00.0

31


Ian Rolling
11
Overs
3
Last Man
Name
Total
43
For
0
Wickets
Andy Barker
10
Last
Innings
000
Extras
22
Extra Scored
How many runs were scored from the no ball by the
1
4
Click here if the extra runs were byes
None
2
5
Cancel
3
6
Ok
Bat
Wicket
Undo
ScoreCard

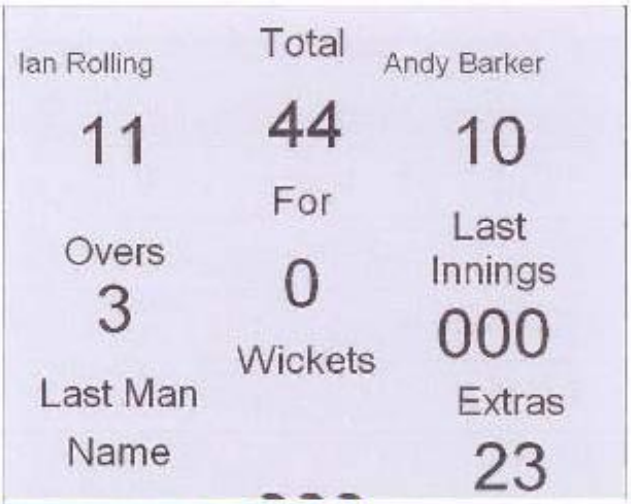
32


Ian Rolling
11
Overs
3
Last Man
Name
Total
44
For
0
Wickets
Andy Barker
10
Last
Innings
000
Extras
23
Current Batsman Facing
Ian Rolling
Current Bowler
Opp two
Run Rate
10.5
Required Run Rate
00.0
Bat
Wicket
Undo
ScoreCard

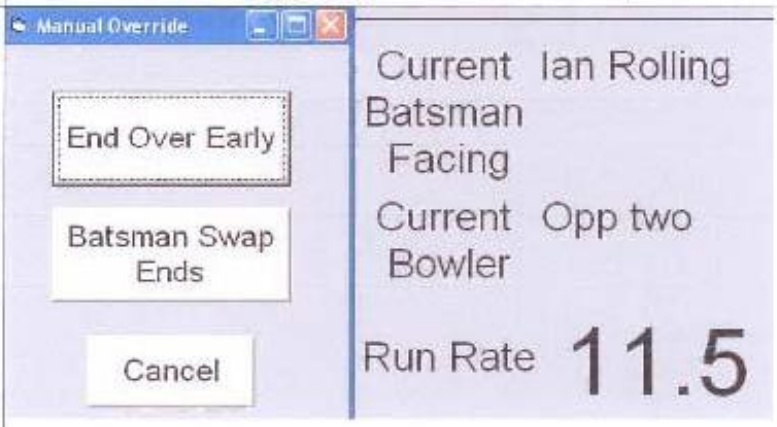
33

Ian Rolling
13
Overs
3
Last Man
Name
Total
46
For
0
Wickets
Andy Barker
10
Last
Innings
000
Extras
23

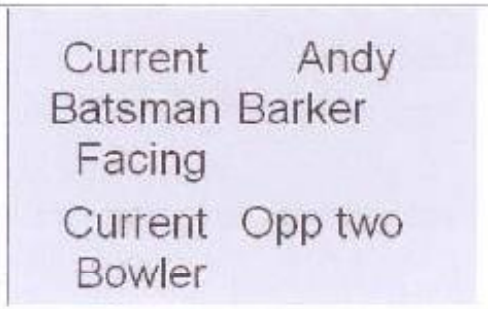
34

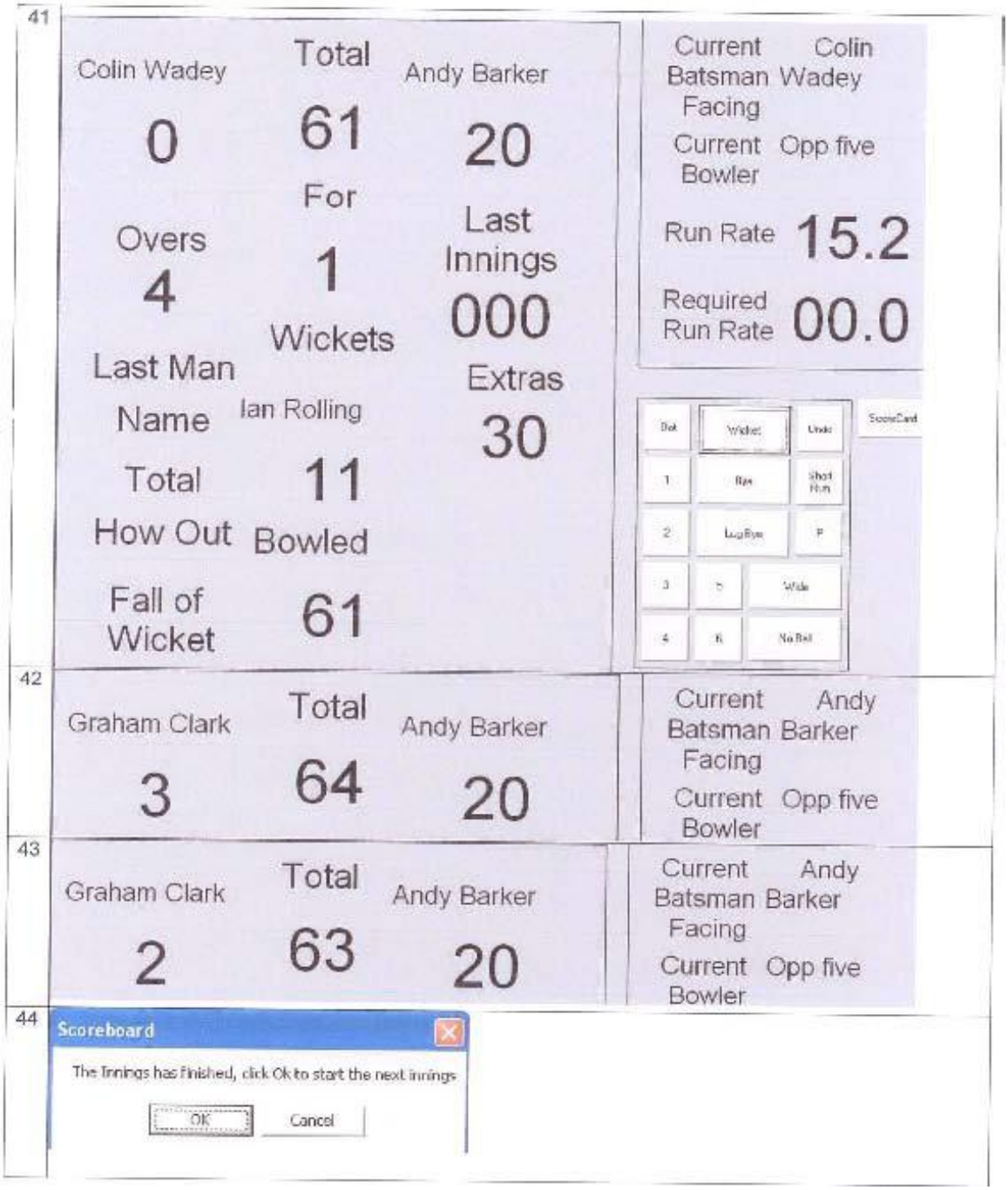

Ian Rolling
Total
Andy Barker
11
44
10
For
Overs
Last Innings
3
0
000
Wickets
Last Man Name
Extras
23

35

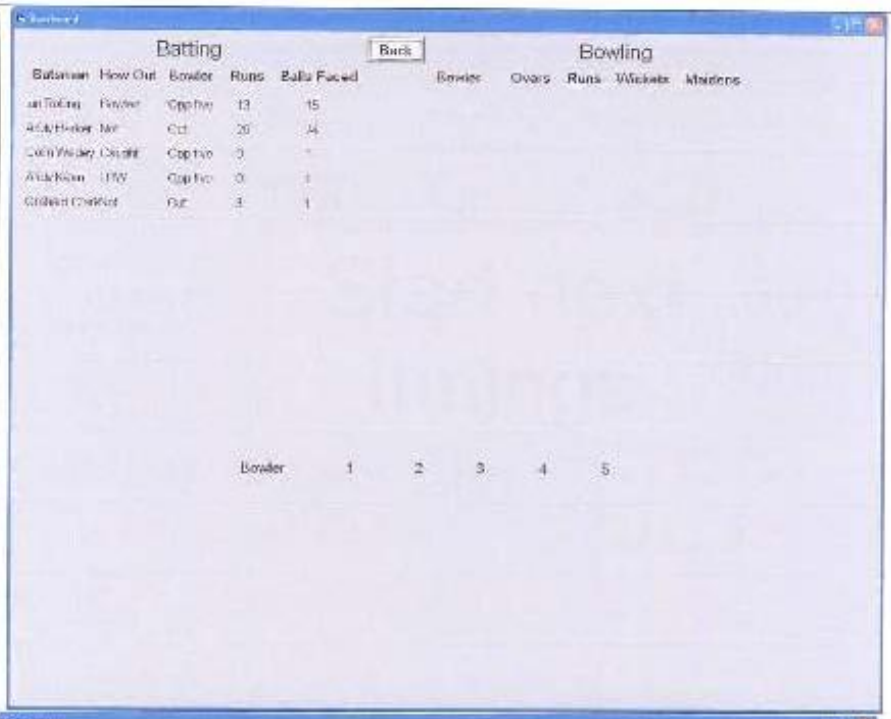

Manual Override
End Over Early
Batsman Swap Ends
Cancel
Current Batsman Facing
Ian Rolling
Current Bowler
Opp two
Run Rate
11.5

36


Current Batsman Facing
Andy Barker
Current Bowler
Opp two


41
Colin Wadey
0
Overs
4
Last Man
Name
Total
How Out
Fall of
Wicket
Total
61
For
1
Wickets
Ian Rolling
11
Bowled
61
Andy Barker
20
Last
Innings
000
Extras
30
Current
Batsman
Facing
Colin
Wadey
Current
Bowler
Opp five
Run Rate
15.2
Required
Run Rate
00.0
Dot
Wicket
Undo
ScoreCard
1
Run
Shot
Run
2
Leg Bye
P
3
b
Wide
4
lb
No Ball
42
Graham Clark
3
Total
64
Andy Barker
20
Current
Batsman
Facing
Andy
Barker
Current
Bowler
Opp five
43
Graham Clark
2
Total
63
Andy Barker
20
Current
Batsman
Facing
Andy
Barker
Current
Bowler
Opp five
44
Scoreboard
The Innings has Finished, click Ok to start the next innings
OK
Cancel

45
A


Batting
Back
Bowling
Batsman How Out Bowler Runs Balls Faced
Bowler Overs Runs Wickets Maidens
Bowler 1 2 3 4 5

45

Batting
Back
Bowling
Batsman How Out Bowler Runs Balls Faced
Bowler Overs Runs Wickets Maidens
Opp one 1 8 3 1
Opp two 1 17 0 0
Opp three 1 21 2 0
Now Opp eight 1 0 0 1
Bowler 1 2 3 4 5
Opp one
Opp two
Opp three
Now Opp eight

48

49

Player ID	Name	Team	Runs Scored	Balls Faced	How Out	Bowler	Overs	Runs Against	Wickets	Maidens
1	Ian Rolling	BGS Staff	13	15	Bowled	Opp five	1	5	1	0
2	Colin Wat	BGS Staff	0	1	Caught	Opp five	0	0	0	0
3	Andy Koo	BGS Staff	0	1	LBW	Opp five	0	0	0	0
4	Roy Jones	BGS Staff	0	0			0	0	0	0
5	Andy Bark	BGS Staff	25	24			0	0	0	0
6	Graham C	BGS Staff	8	2			0	0	0	0
7	Ben Smit	BGS Staff	0	0			0	0	0	0
8	Kevin Bla	BGS Staff	0	0			0	9	1	0
9	Andrew P	BGS Staff	0	0			1	0	0	1
10	Kevin Sm	BGS Staff	0	0			0	0	0	0
11	Steve Ha	BGS Staff	0	0			0	0	0	0
12	Opp one	Test Opp	0	1	Caught	Ian Rolling	1	0	0	1
13	Opp two	Test Opp	5	3	Bowled	Andrew P	1	12	0	0
14	Opp three	Test Opp	0	1	Obstructe	Andrew P	1	21	0	0
15	Opp four	Test Opp	0	1	Timed Ou	Andrew P	0	0	0	0
16	Opp five	Test Opp	0	1	Run Out	Ian Rolling	0	13	2	0
17	Opp six	Test Opp	0	1	Hit Ball T	Andrew P	0	0	0	0
18	Thirty Cha	Test Opp	1	4			0	0	0	0
19	New Opp	Test Opp	0	1	Handling	Andrew P	1	0	0	1
20	New Opp	Test Opp	0	1	Retired	Andrew P	0	0	0	0
21	New Opp	Test Opp	7	2	LBW	Kevin Bla	0	0	0	0
22	New Opp	Test Opp	0	2	Bowled	Kevin Bla	0	0	0	0

50

Player ID	Name	Team	Inns	Ave	High Score	Not Outs	Overs	Runs	Wickets	Maidens	Econ. Rate	Strike Rate
1	Player 1	BGS Staff	3	0	62	3	3	65	0	0	78	0
2	Player 2	BGS Staff	7	0	18	2	0	18	0	0	0	0
3	Player 3	BGS Staff	0	0	0	0	0	0	0	0	0	0
4	Player 4	BGS Staff	0	0	0	0	0	0	0	0	0	0
5	Player 5	BGS Staff	0	0	0	0	0	0	0	0	0	0
6	Player 6	BGS Staff	0	0	0	0	0	0	0	0	0	0
7	Player 7	BGS Staff	0	0	0	0	0	0	0	0	0	0
8	Player 8	BGS Staff	0	0	0	0	0	0	0	0	0	0
9	Player 9	BGS Staff	0	0	0	0	0	0	0	0	0	0
10	Player 10	BGS Staff	0	0	0	0	0	0	0	0	0	0
11	Player 11	BGS Staff	0	0	0	0	0	0	0	0	0	0
12	Player 12	Opposition	2	0	48	2	7	48	0	1	13.5	0

User Testing

After testing the system myself, I asked my users to test the system. Firstly M[redacted] test the system. On the next page shows the questionnaire I asked my users to fill in upon completing the testing.

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

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

User Questionnaire

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

[redacted]

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:

lblRunRate = lblRunstotal / (lblOversBowled + 1) to

lblRunRate = lblRunstotal / (lblOversBowled + (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:

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:

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 lblBatsmanAText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
   lblBatsmanA.Caption = lblBatsmanA - 1
Else
   lblBatsmanB.Caption = lblBatsmanB - 1
End If
```

Was changed to

```
If lblBatsmanAText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
   lblBatsmanA.Caption = lblBatsmanA - 1
ElseIf lblBatsmanBText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
   lblBatsmanB.Caption = lblBatsmanB - 1
Else
   lblTotalExtras.Caption = lblTotalExtras - 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.Bowlingwkts = PlayerMatchData.Bowlingwkts + 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 these corrections

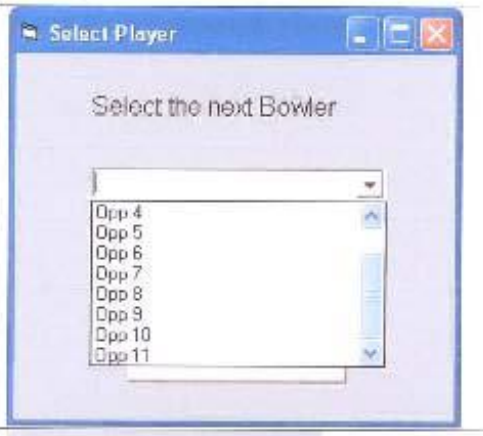
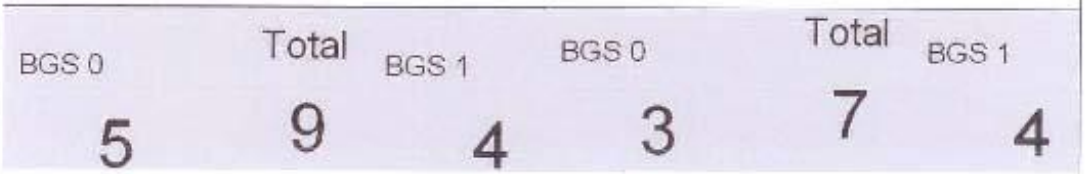
[Start Game]

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), Wide(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.

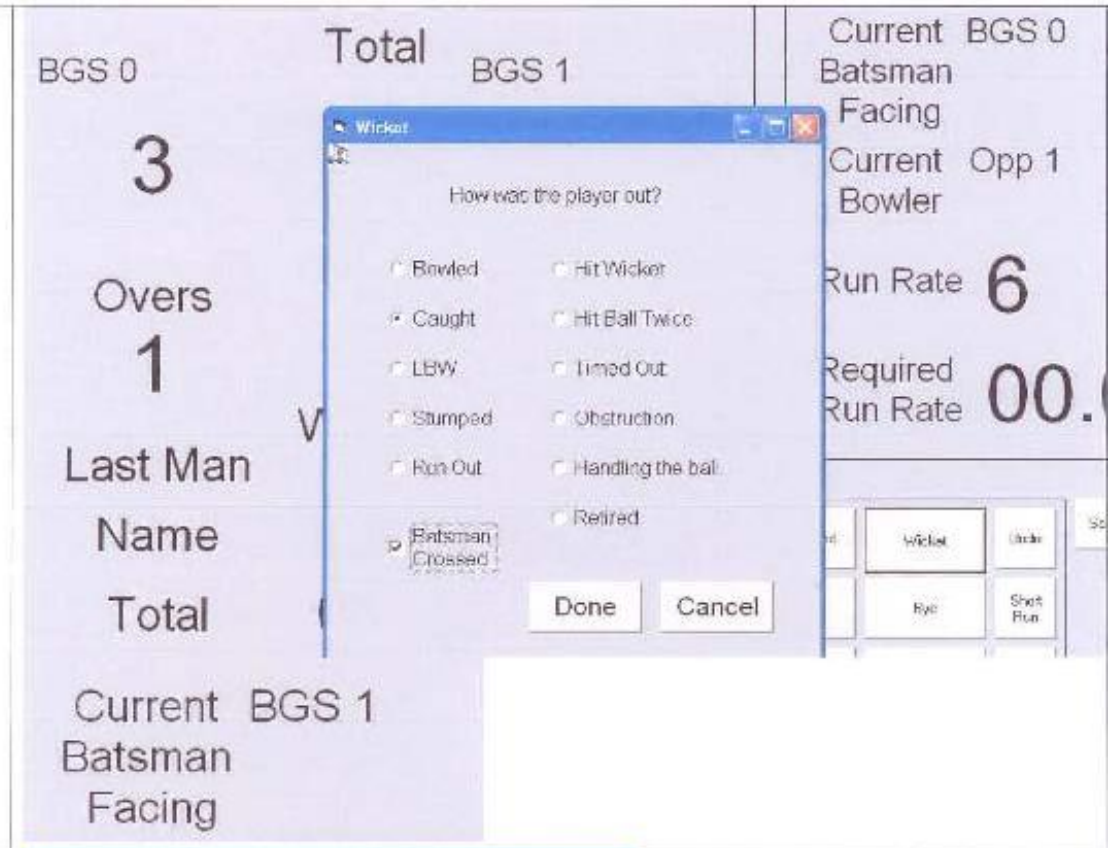
Test Detail

Test #	Test name	Input Data	Expected Result	Output (SS = Screenshot)	Actual Result
1	Check the penalty button	Starting the game	The penalty button has penalty on it	SS 1	As expected
2	Prolong over	Prolong over button pressed	When the select bowler window comes up again, the team is only listed once	SS 2	As expected
3	Checking run rate	Second over: three	The run rate should be 9 / 1.5 = 6	SS 3	As expected
4	Double clicking short run	Press short run twice	Two runs should be deducted from the total and from BGS 1	SS 4	As expected
5	Wicket caught where batsman cross	Press wicket and tick the batsman cross box	BGS 1 should be facing for the next ball	SS 5	As expected
6	Checking if the extra tick box appears	Press No ball, then wide, then bye	On the wide and bye windows, the option for the runs to be byes should not appear	SS 6	As expected
7	Short run after a bye	Press bye(2) then short run	The sort run should be deducted from extras not from the batsman	SS 7	As expected
8	Batsman swap ends	Bye(3) pressed	The facing batsman should change to the other	SS 8	As expected

	from running an extra		batsman		
9	Retirement shouldn't give a wicket	Wicket(Retired) selected	The total wickets shouldn't increase	SS 9	As expected
10.	End innings should remove back button	End the innings through the test 9 wickets button	When the user reaches the startplayers window, the back button should not appear	SS 10	As expected

1	Penalty
2	
3	
4	

5


Total
BGS 0
BGS 1
3
Overs
1
Last Man
Name
Total
Current Batsman Facing
BGS 1
Current Batsman Facing
BGS 0
Current Bowler
Opp 1
Run Rate
6
Required Run Rate
00.
Wicket
How was the player out?
Bowled
Caught
LBW
Stumped
Run Out
Batsman Crossed
Hit Wicket
Hit Ball Twice
Timed Out
Obstruction
Handling the ball
Retired
Done
Cancel

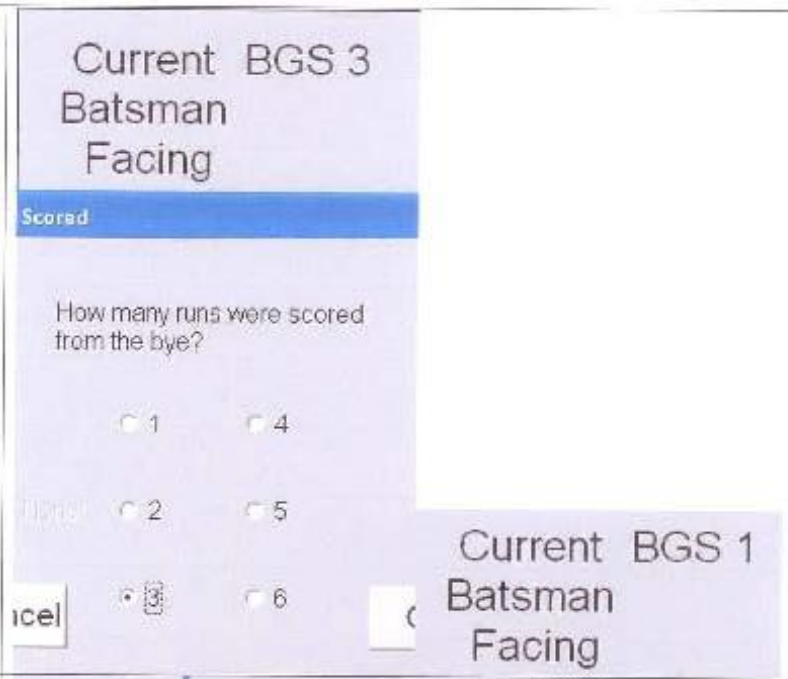
6

Extra Scored
How many runs were scored from the bye?
1
2
3
4
5
6
Cancel
Ok

7

Extras
4
Extras
3

8


Current Batsman Facing
BGS 3
Scored
How many runs were scored from the bye?
1
4
2
5
3
6
Current Batsman Facing
BGS 1
Retired

9

1
Batsman Crossed
Done
1
Wickets

10

Opening Players
Select the following:
Facing Batsman
Other Batsman
Opening Bowler
Back
Done

Second User Testing

After Mr ested the system, I then asked Mr o 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.

Headed notepaper used.

Dea

With reference to the computerised system you have developed, I would like to take this opportunity to thankyou for the time and effort you have put into the system and wish you well with you're 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 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 s from a list of batsman or entered by the user
- The system will have a separate window with a full scorecard of the whole matc
-

page 75 & 122

- 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 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

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

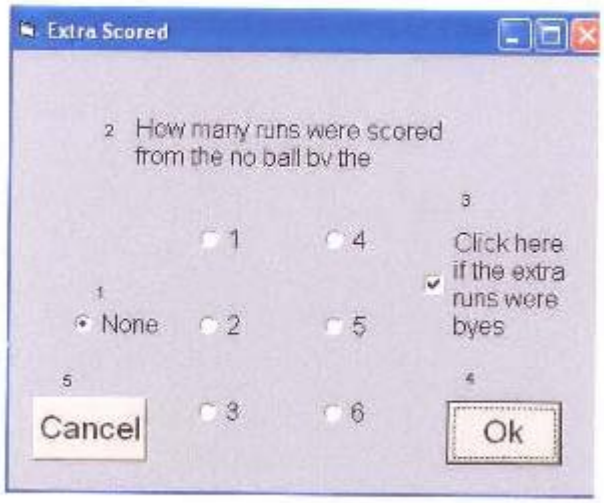
Yours sincerely,

[redacted] [redacted]

Scorer Statistician

Annotated Listings

This is the Extras form and contains:



1. Optextras(0 – 6) Array
2. lblHowmanyextras
3. chkextrabyes
4. cmdextrasok
5. cmdextrascancel

```
Private Sub cmdextrascancel_Click()
Me.Hide
End Sub
```
This hides the window

```
Private Sub cmdextrasok_Click()
For i = 0 To 6
   If optextras(i).Value = True Then
      Scoreboard.lblRunstotal.Caption = Int(Scoreboard.lblRunstotal) + i
      Scoreboard.lblTotalExtras.Caption = Int(Scoreboard.lblTotalExtras) + i
      l = i
   End If
Next i
If k = 1 Then
   Scoreboard.lblRunstotal.Caption = Int(Scoreboard.lblRunstotal.Caption) + Runsperwnb
   Scoreboard.lblTotalExtras = Int(Scoreboard.lblTotalExtras) + Runsperwnb
End If
If k = 1 And chkextrabyes.Value = 0 Then
   i = l
Else
   i = 0
End If
k = l
Call AfterBall
Me.Hide
End Sub
```
This loops through 0 to 6 finding the number of runs scored in the extra and sets that to l.

If the extra was a wide or no ball then the additional runs for a wide or no ball are added.

This is an if statement that makes i equal to the number of runs the batsman scored.

K is set equal to 1 then AfterBall is called

```
Private Sub AfterBall()
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
         PlayerMatchData.Battingruns = PlayerMatchData.Battingruns + i
```
Opens the Summary file then loops through each record. If the Current batsman is found, i.e. ~~72 of 110~~ 77 of 122 is added to their total.

```
        PlayerMatchData.BattingBallsFaced = PlayerMatchData.BattingBallsFaced + 1
        Put #1, j, PlayerMatchData
      End If
      If PlayerMatchData.MatchPlayerName = CurBowl And k = 1 Then
        PlayerMatchData.bowlingruns = PlayerMatchData.bowlingruns + i
        Put #1, j, PlayerMatchData
      End If
    Next j
Close #1
If CurBat = BatA Then
  Scoreboard.lblBatsmanA.Caption = Scoreboard.lblBatsmanA.Caption + i
Else
  Scoreboard.lblBatsmanB.Caption = Scoreboard.lblBatsmanB.Caption + i
End If
If k = 1 And Extraball = True Then
Else
  Bowled = Bowled + 1
End If
If k = 1 Or k = 3 Or k = 5 Then
  Tempbat = CurBat
  CurBat = Otherbat
  Otherbat = Tempbat
End If
Scoreboard.lblFacingBatsman.Caption = CurBat
If lblOversBowled <> 0 Then
  Scoreboard.lblRunRate = Scoreboard.lblRunstotal / Scoreboard.lblOversBowled
End If
Open StatisticPath For Random As #1 Len = Len(MatchData)
I = LOF(1) / Len(MatchData) + 1
  With MatchData
    .Ballnumber = Ballnum
    .Bowler = CurBowl
    .Batsman = CurBat
    If k > -1 And k < 7 Then
      .Runs = k
    End If
    .Extra = Extratype
    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
    Else
      .Wicket = ""
    End If
  End With
  Put #1, I, MatchData
  Open SummaryPath For Random As #2 Len = Len(PlayerMatchData)
  For j = 1 To 22
    Get #2, j, PlayerMatchData
      If PlayerMatchData.MatchPlayerName = CurBat Then
```

an additional ba is added to the balls faced

Adds the runs to the bowler if applicab

Adds the runs scored to the relevant batsman

If an extraball is given then bowled doesn't incre otherwise it does.

Swaps the batsman if they ran an odd number of runs.

Updates the scoreboard with the current batsman and the ru rate.

Updates the statistics file with all the data for this ball.

Updates the summary file.

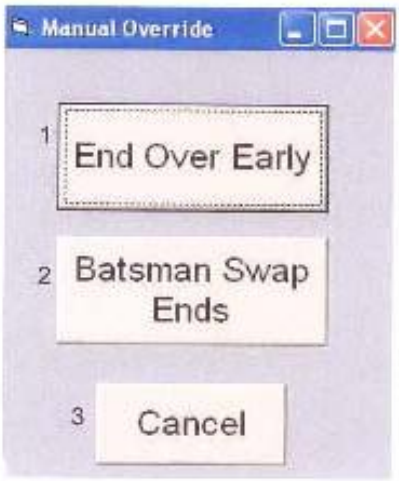
```
            PlayerMatchData.BattingBallsFaced = PlayerMatchData.BattingBallsFaced + 1
            Put #2, j, PlayerMatchData
        End If
    Next j
Close #1
Close #2
Ballnum = Ballnum + 1
Extratype = ""
WicketType = ""
If Bowled = 6 Then
    Call PreEndOver
End If
If chkextrabyes.Value = True Then
    chkextrabyes.Value = False
End If
End Sub
```

adds 1 to the ball num.

Resets Extra and Wicket variable.

If 6 balls have been bowled the over ends.

Resets the check box.



This is the ManualOverride form and contains:

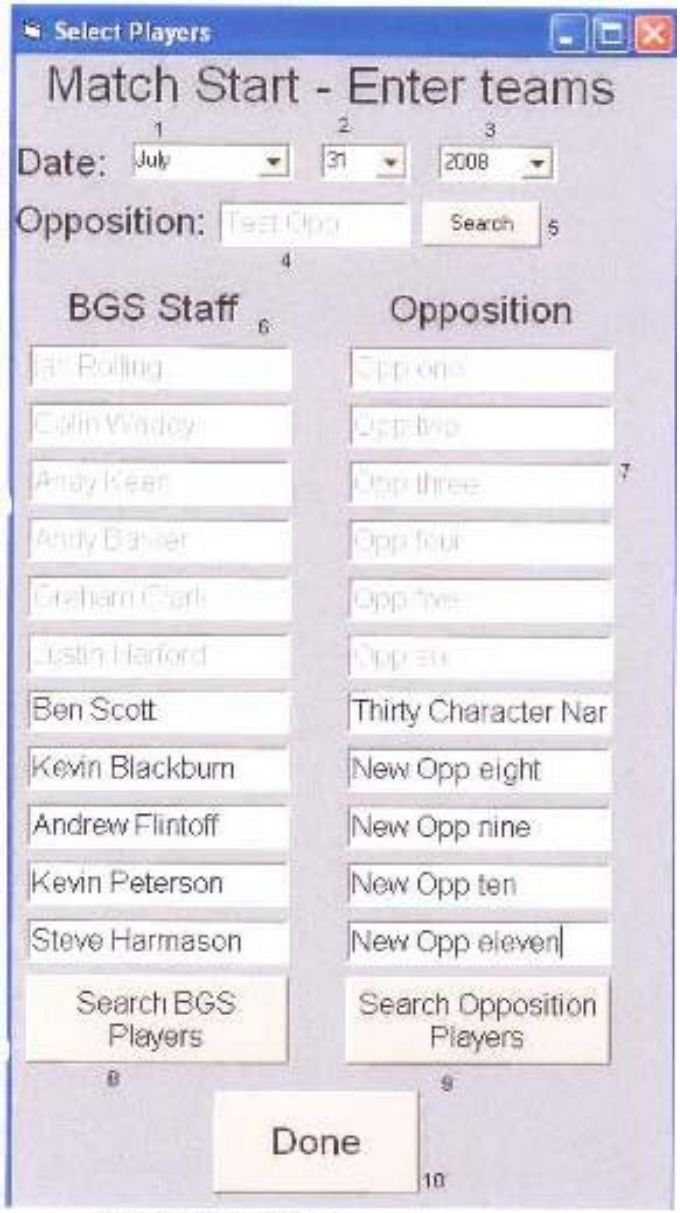
1. cmdEndOver
2. cmdBatSwap
3. cmdMOcancel

```
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
```

Swaps the batsman when pressed

Calls end over

Hides the window.




```
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
         .strikerate = 0
         .bowlingruns = 0
      End With
   Put #1, lastrecord, PlayerRecord
   End If
Next i
For i = 0 To 10
   If txtOppoplayer(i).Enabled = True Then
      With PlayerRecord
         lastrecord = LOF(1) / Len(PlayerRecord) + 1
         .PlayerID = lastrecord
         .PlayerName = txtOppoplayer(i)
         .PlayerTeam = txtOpposition
         .Innings = 0
         .AverageRuns = 0
         .Highscore = 0
         .NotOuts = 0
         .Overs = 0
         .Runs = 0
         .Wickets = 0
         .Maidens = 0
         .econrate = 0
         .strikerate = 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
```

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
            .Bowlingwkts = 0
            .Bowlingmaidens = 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
            .Bowlingwkts = 0
            .Bowlingmaidens = 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
```

Adds all 22 players in the team to the summary file.

Shows the settings window.

Opens the team search window.

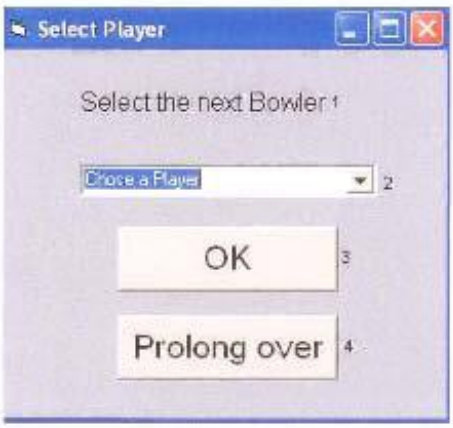
Sets the number of days in the day combo box, when the month is chosen.

```
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
```

Then sets the date for years from 2007 to the current year.

```
Private Sub Form_Load()
Unload Scoreboard
End Sub
```

Unloads the scoreboard



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
```

Calls a subroutine depending whether its a next bowler or batsman.

```
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.lblOversBowled.Caption = Scoreboard.lblOversBowled + 1
   Bowled = 0
End If
Call EndOver
```

Validates to check a player is selected.

Sets the Curbowl to the new bowler.

Clears the nextplayer list.

Adds 1 to the overs bowled.

```
Me.Hide
End Sub
```

Then hides the window

```
Private Sub NewBatsman()
If cmbNextPlayer.Text = "" Then
   MsgBox "Please Select a Player"
Else
   If BatOut = BatA Then
      BatA = cmbNextPlayer.Text
      Scoreboard.lblBatsmanAText.Caption = BatA
   Else
      BatB = cmbNextPlayer.Text
      Scoreboard.lblBatsmanBText.Caption = BatB
   End If
   If Wicket.chkBatcrossed.Value = True Then
      CurBat = Otherbat
      Otherbat = cmbNextPlayer.Text
   Else
      CurBat = cmbNextPlayer.Text
   End If
   Scoreboard.lblFacingBatsman = CurBat
   Chosen = False
   j = 2
   Do
      If Battingorder(j) = "" Then
         Battingorder(j) = cmbNextPlayer.Text
         Chosen = True
      End If
      j = j + 1
   Loop Until Chosen = True
   Me.Hide
   cmbNextPlayer.Clear
   Bowled = Bowled + 1
   If Bowled >= 6 Then
      Call PreEndOver
   End If
End If
End Sub
```

Replaces the batsman out with the new batsman.

If the batsman cross then the batsman swop.

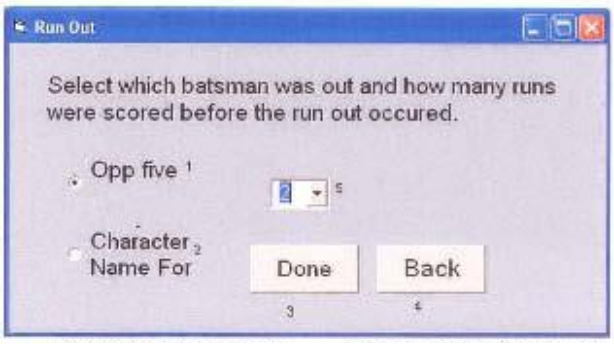
Updates the scoreboard.

Sets the next batsman into the batting order.

Calls the end of the over if 6 balls have been bowled.

```
Private Sub cmdProlong_Click()
cmdNextPlayer.Clear
Me.Hide
End Sub
```

Hides the window to allow another ball to be bowled



This is the Run Out form and contains:

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

```
Private Sub cmbRunOutBack_Click()
Me.Hide
End Sub
```

Hides the window.

```
Private Sub cmbRunOutdone_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"
End If
```

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

```
j = Int(Scoreboard.lblRunstotal)
Scoreboard.lblRunstotal = j + Int(cmbrunsscored.Text)
Call EndBall(Int(cmbrunsscored.Text))
```

Updates the total runs.

```
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.lblLastManFow = Scoreboard.lblRunstotal
```

Opens the summary file then updates the last man information.

```
For i = 1 To 22
   Get #1, i, PlayerMatchData
      If PlayerMatchData.MatchPlayerName = BatOut Then
         PlayerMatchData.BattingHowOut = Scoreboard.lblLastManHowOut.Caption
         PlayerMatchData.BattingWktBowler = CurBowl
         Put #1, i, PlayerMatchData
      End If
Next i
```

Loops through the summary file and updates the wicket information for that batsman.

```
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.cmbNextPlayer.AddItem PlayerMatchData.MatchPlayerName
      End If
Next i
```

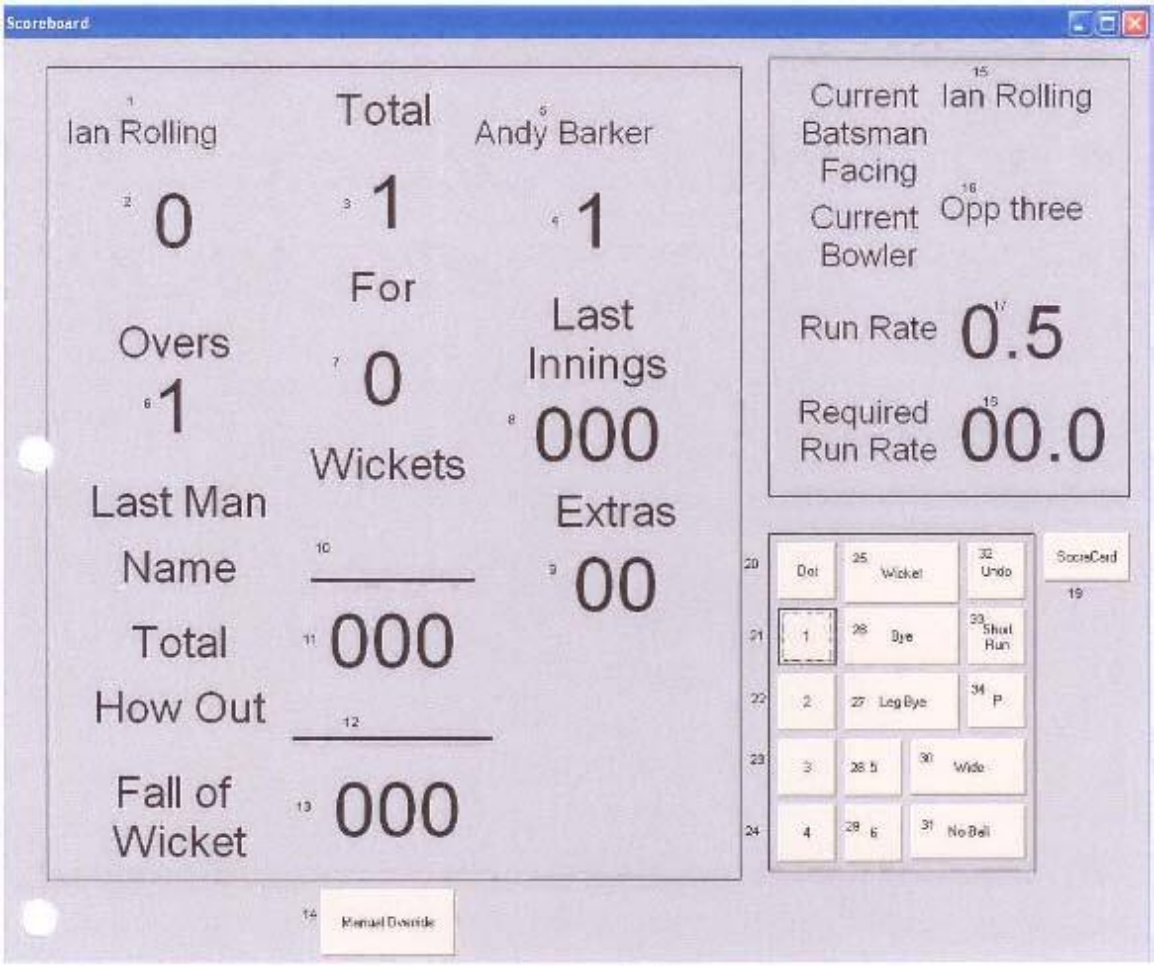
Then loops through creating a list for the next batsman in the nextplayer form

```
Me.Hide
NextPlayer.lblNextPlayertext.Caption = "Select the next batsman"
j = 2
```

Sets the nextplayer 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 buttons 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. lblrunstotal
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. lblRqRunRate
19. cmdScoreardShow
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. cmdPenelties
35. cmdStartGame

```
Dim Totalruns As Integer
Dim TotalWkts As Integer
Dim OversBowled As Integer
Dim Lastinns As Integer
Dim TotalExtras As Integer
Dim LmRuns As Integer
Dim LmNum As Integer

Private Sub cmdBye_Click()
Extras.Show
```

```
Extratype = "Bye"                                                    Sets
Extras.chkextrabyes.Visible = False
k = 0
Extras.lblHowmanyextras.Caption = "How many runs were scored fro
Extras.optextras(0).Enabled = False
End Sub

Private Sub cmdLegBye_Click()
Extras.Show                                                          Se
Extratype = "Leg Bye"
Extras.chkextrabyes.Visible = False                                  wind
k = 0
Extras.lblHowmanyextras.Caption = "How many runs were scored fro
Extras.optextras(0).Enabled = False
End Sub

Private Sub cmdManOverRide_Click()                                   Shows
ManualOverride.Show
End Sub

Private Sub cmdNoBall_Click()
Extras.Show                                                          Sets u
Extratype = "No Ball"
Extras.chkextrabyes.Visible = True                                   windo
k = 1
Extras.lblHowmanyextras.Caption = "How many runs were scored fro
Extras.optextras(0).Enabled = True
End Sub

Private Sub cmdPenelties_Click()
Extras.Show                                                          Sets u
Extratype = "Penelty"
Extras.chkextrabyes.Visible = False
k = 0                                                                for a
Extras.lblHowmanyextras.Caption = "How many runs were scored fro
Extras.optextras(0).Enabled = True
End Sub

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

Private Sub cmdrun_KeyPress(Index As Integer, KeyAscii As Integer
If KeyAscii > 47 And KeyAscii < 55 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
```

Validates that the scorecard can be shown, and shows it.

```
Private Sub cmdShortRun_Click()
Close #1
Open StatisticPath For Random As #1 Len = Len(MatchData)
i = LOF(1) / Len(MatchData)
Get #1, i, MatchData
    MatchData.Runs = MatchData.Runs - 1
    If lblBatsmanAText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
        lblBatsmanA.Caption = lblBatsmanA - 1
    ElseIf lblBatsmanBText.Caption = MatchData.Batsman And Trim(MatchData.Extra) = "" Then
        lblBatsmanB.Caption = lblBatsmanB - 1
    Else
        lblTotalExtras.Caption = lblTotalExtras - 1
    End If
    lblRunstotal.Caption = lblRunstotal - 1
Put #1, i, MatchData
End Sub
```

Takes away one run from the batsman and the total for a short run.

```
Private Sub cmdStartGame_Click()
cmdStartGame.Visible = False
lblFacingBatsman = OpeningBatsman
lblCurrentBowler = OpeningBowl
CurBowl = OpeningBowl
CurBat = OpeningBatsman
Otherbat = OpeningNonfaceBat
lblBatsmanAText.Caption = OpeningBatsman
lblBatsmanBText.Caption = OpeningNonfaceBat
lblBatsmanBText.Alignment = 2
lblBatsmanAText.Alignment = 2
BatA = OpeningBatsman
BatB = OpeningNonfaceBat
Battingorder(0) = OpeningBatsman
Battingorder(1) = OpeningNonfaceBat
cmdrun(0).SetFocus
Maiden = True
End Sub
```

This sets up the scoreboard at the start of a new game or innings, using variables set in the settings or startplayers form.

```
Private Sub cmdUndo_Click()
Close #1
Open StatisticPath For Random As #1 Len = Len(MatchData)
    i = LOF(1) / Len(MatchData)
    Get #1, i, MatchData
    lblRunstotal = lblRunstotal - 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
        Bowled = Bowled - 1
    End If
```

When undo is pressed the system looks at what happened the previous ball and resets the system to that ball

```
    If Bowled = -1 Then
        Get #1, (i - 1), MatchData
            Scoreboard.lblCurrentBowler.Caption = MatchData.Bowler
            Bowled = 6
            Tempbat = CurBat
            CurBat = Otherbat
            Otherbat = Tempbat
            Scoreboard.lblOversBowled.Caption = Scoreboard.lblOversBowled - 1
    End If
    If Trim(MatchData.Wicket) <> "" Then
        Scoreboard.lblWickets.Caption = Scoreboard.lblWickets - 1
        Scoreboard.lblLastManName.Caption = LastBatName
        Scoreboard.lblLastManTotal.Caption = LastBatTotal
        Scoreboard.lblLastManFow.Caption = LastBatFow
        Scoreboard.lblLastManHowOut.Caption = LastManHowOut
    End If
        Get #1, i, MatchData
    If lblBatsmanAText = CurBat Then
        lblBatsmanA.Caption = lblBatsmanA - MatchData.Runs
    Else
        lblBatsmanB.Caption = lblBatsmanB - MatchData.Runs
    End If
    lblFacingBatsman.Caption = CurBat
    With MatchData
        .Ballnumber = 0
        .Batsman = ""
        .BatsmanOut = 0
        .Bowler = ""
        .Extra = ""
        .Runs = 0
        .Wicket = ""
    Put #1, i, MatchData
    End With
Close #1
End Sub
```

If the undo is pressed at the beggining of an over it will revert to the previus over.

The system then clears what happened for the ball that was undone.

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

Opens the wicket form when a wicket occurs.

```
Private Sub cmdWide_Click()
LastAction = "Wide"
Extras.Show
Extratype = "Wide"
Extras.chkextrabyes.Visible = False
k = 1
Extras.lblHowmanyextras.Caption = "How many extra runs were score from this wide?"
Extras.optextras(0).Enabled = True
End Sub
```

Sets up and shows the extras window for a wide.

```
Private Sub Command1_Click()
Call UpdateFile
End Sub
```

This was a scenario test for updating the file and spreadsheet

```
Private Sub Form_Load()
```

```
Secondinnings = False
If j = 1 Then
Else
Me.Hide
MatchStart.Show
End If
End Sub

Private Sub AfterBall(i As Integer)
lblRunstotal = lblRunstotal + 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
         PlayerMatchData.Battingruns = PlayerMatchData.Battingruns + i
      End If
      If PlayerMatchData.MatchPlayerName = CurBowl Then
         PlayerMatchData.bowlingruns = PlayerMatchData.bowlingruns + i
      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
Bowled = Bowled + 1
If i = 1 Or i = 3 Or i = 5 Then
   Tempbat = CurBat
   CurBat = Otherbat
   Otherbat = Tempbat
End If
lblFacingBatsman.Caption = CurBat
If lblOversBowled <> 0 Then
   lblRunRate = lblRunstotal / (lblOversBowled + (Bowled / 6))
   If lblLastInnings <> 0 Then
      lblRqRunRate = (lblLastInnings - lblRunstotal) / (LimitedOvers - lblOversBowled)
      If lblRqRunRate < 0 Then
         If BattingTeam = "BGS Staff" Then
            k = 1
         Else
            k = 2
         End If
         Call EndMatch(k)
      End If
   End If
End If
If Bowled >= 6 Then
   Call PreEndOver
End If
End Sub
```

Sets Secondinnings to False when a game starts

This adds any run scored to the relevant batsman and bowler in the summary file.

Updates the scoreboard and swaps the batsman if required.

Updates the run rate and required run rate.

Calls the end match if the required run rate goes to or below 0.

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()
```

```
Me.Hide
Unload Scorecard
End Sub

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, j, PlayerMatchData
      End If
   Next j
For i = 0 To 10
   If i <> 0 Then
      Load lblBatsmanlist(i)
      Load lblHowOut(i)
      Load lblWicketBowler(i)
      Load lblBattingruns(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 = 1680
   lblHowOut(i).Visible = True
   lblHowOut(i).Caption = ""
   lblWicketBowler(i).Top = 1320 + 480 * i
   lblWicketBowler(i).Left = 3000
   lblWicketBowler(i).Visible = True
   lblWicketBowler(i).Caption = ""
   lblBattingruns(i).Top = 1320 + 480 * i
   lblBattingruns(i).Left = 4320
   lblBattingruns(i).Visible = True
   lblBattingruns(i).Caption = ""
   lblBallsFaced(i).Top = 1320 + 480 * i
   lblBallsFaced(i).Left = 5640
   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.BattingWktBowler
         If Trim(lblWicketBowler(i).Caption) = "" Then
            lblHowOut(i).Caption = "Not"
            lblWicketBowler(i).Caption = "Out"
```

Hides the score card to go back to the scoreboard.

This sets I to the number of bowlers that have bowled at least an over.

This creates a table of batsman with all the relevant columns

This fills in the table for each batsman.

```
            End If
            lblBattingruns(i).Caption = PlayerMatchData.Battingruns
            lblBallsFaced(i).Caption = PlayerMatchData.BattingBallsFaced
        End If
    Next j
Next i
For i = 0 To l - 1
    If i <> 0 Then
        Load lblBowlerlist(i)
        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 = 8160
    lblBowlerlist(i).Visible = True
    lblBowlerlist(i).Caption = ""
    lblBowlerlist(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 = 12000
    lblBowlerWicket(i).Visible = True
    lblBowlerWicket(i).Caption = ""
    lblMaidens(i).Top = 1320 + 480 * (i)
    lblMaidens(i).Left = 13320
    lblMaidens(i).Visible = True
    lblMaidens(i).Caption = ""
    j = 1
    Do
        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.Bowlingwkts
                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 l
        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.

```
        lblBowlerdatalist(i).Left = 4440
        lblBowlerdatalist(i).Visible = True
        lblBowlerdatalist(i).Caption = lblBowlerlist(i - 1).Caption
        lblBowlerdatalist(i).Width = 2000
    For k = 1 To 5
        Load lblOverData(k + (i - 1) * 5)
        lblOverData(k + (i - 1) * 5).Top = 8400 + i * 600
        lblOverData(k + (i - 1) * 5).Left = 5280 + k * 1200
        lblOverData(k + (i - 1) * 5).Visible = True
    Next k
Next i
k = 0
p = 1
Open StatisticPath For Random As #2 Len = Len(MatchData)
Do
i = 1
    SelectedBowler = lblBowlerlist(k)
    p = k * 5
    Do
        Get #2, i, MatchData
            If MatchData.Bowler = SelectedBowler Then
                If MatchData.Ballnumber Mod 20 = 0 Then
                    p = p + 1
                End If
                If Trim(MatchData.Wicket) <> "" Then
                    lblOverData(p).Caption = lblOverData(p).Caption & "W "
                ElseIf Trim(MatchData.Extra) = "Wide" Or MatchData.Extra = "No Ball" Then
                    lblOverData(p).Caption = lblOverData(p).Caption & "+"
                    If MatchData.Runs <> 0 Then
                        lblOverData(p).Caption = lblOverData(p).Caption & MatchData.Runs
                    Else
                        lblOverData(p).Caption = lblOverData(p).Caption & " "
                    End If
                Else
                    lblOverData(p).Caption = lblOverData(p).Caption & MatchData.Runs & " "
                End If
            End If
        i = i + 1
    Loop Until i >= LOF(2) / Len(MatchData) + 1
    k = k + 1
Loop Until k = l
Close #2
Close #1
End Sub
```

This lists all the bowlers in the bowler table, with 5 boxes to show the overs they have bowled.

This fills in these 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.ListIn
End Sub

Private Sub cmdMoveRight_Click()
SelectedPlayer = lstDatalist.Text
```

```
lstSelectedlist.AddItem SelectedPlayer
lstSelectedlist.ItemData(lstSelectedlist.NewIndex) = lstDatalist.ListIndex
lstDatalist.SetFocus
End Sub

Private Sub cmdsearchdone_Click()
i = 0
If Teamsearch 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.cmdOpposearch.Enabled = True
   End If
Else
   Startvalue = 0
   If Trim(searchteam) <> "BGS Staff" Then
      Do
         If MatchStart.txtOppoplayer(Startvalue).Enabled = False Then
            Startvalue = Startvalue + 1
         End If
      Loop Until MatchStart.txtOppoplayer(Startvalue).Enabled = True
   Else
      Do
         If MatchStart.txtBGSplayer(Startvalue).Enabled = False Then
            Startvalue = Startvalue + 1
         End If
      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
      Do
      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 & "\CricketMasterFile.txt" For Random As #1 Len = Len(PlayerRecord)
```

Adds a player to the selected list.

Validates the seach for a team.

Sets the opposing team to the chosen team.

Validates the number player selected then adds t to an array of hometeam or awayteam.

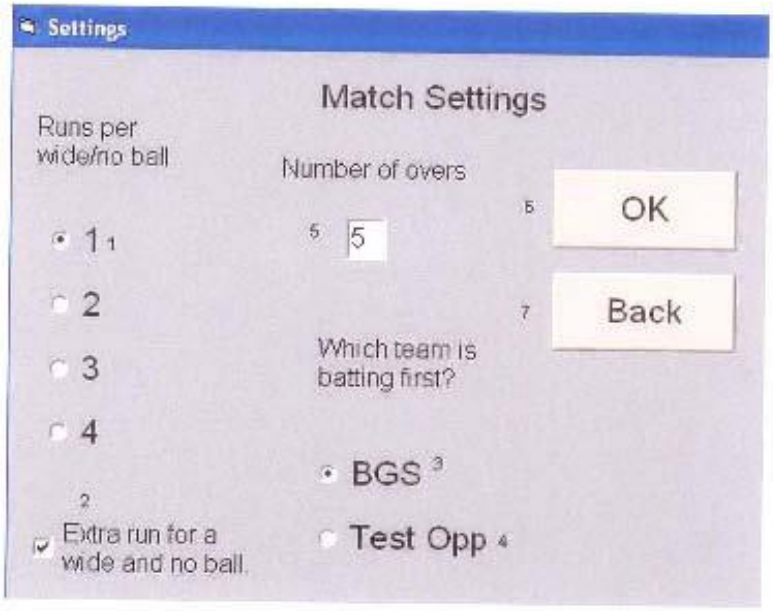
Hides the matchstat windo

```
If Teamsearch Then
Search.Caption = "Team Search"
For recordnum = 1 To LOF(1) / 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
            End If
         Next i
      Else
         Listed = True
      End If
      If Listed = False Then
         lstDatalist.AddItem PlayerRecord.PlayerTeam
      End If
   Next recordnum
Else
Search.Caption = "Player Search"
For recordnum = 1 To LOF(1) / Len(PlayerRecord)
   Get #1, recordnum, PlayerRecord
      If Trim(PlayerRecord.PlayerTeam) = Trim(searchteam) Then
         lstDatalist.AddItem PlayerRecord.PlayerName & PlayerRecord.PlayerID
      End If
   Next recordnum
End If
Close #1
End Sub
```

On loading 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 file.

```
Private Sub PlayerFill()
For i = Startvalue To (Startvalue + lstSelectedlist.ListCount - 1)
If Trim(searchteam) <> "BGS Staff" Then
   MatchStart.txtOppoplayer(i).Text = awayteam(i).PlayerName & awayteam(i).PlayerID
   MatchStart.txtOppoplayer(i).Enabled = False
Else
   MatchStart.txtBGSplayer(i).Text = hometeam(i).PlayerName & hometeam(i).PlayerID
   MatchStart.txtBGSplayer(i).Enabled = False
End If
Next i
SelectedPlayers = lstSelectedlist.ListCount - 1
End Sub
```

This puts players into the match start window.



This is the settings window, and contains the following:

1. Radrunsperwide(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
```

Hides this window

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

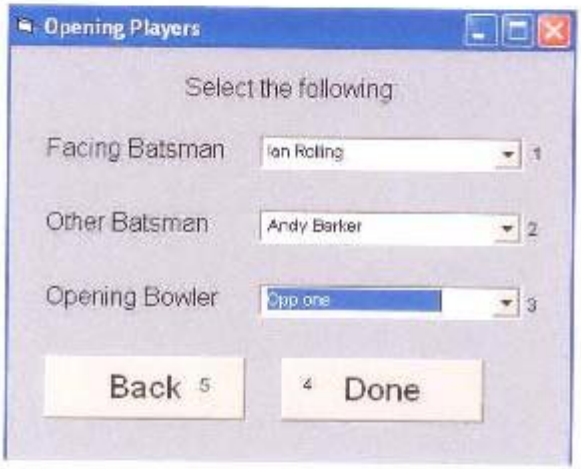
This sets all the variables to the options selected.

```
    OpeningBatteam = "BGS Staff"
ElseIf RadBatselect(1).Value = True Then
    OpeningBatteam = OpposingTeam
Else
    MsgBox "Select a team to opening the batting", vbOKOnly
    j = 1
End If
If j = 0 Then
Me.Hide
StartPlayers.Show
End If
End Sub
```

Validates a batting team has been selected.

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

When the form is loaded the opposing team is set as an option to open the batting.



This is the StartPlayers form and contains the following:

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

```
Dim deleteditem As String
Dim Itemadded As Boolean

Private Sub cmbFacingBat_Click()
Itemadded = False
For j = 0 To 10
    If deleteditem <> "" And Itemadded = False Then
        cmbNonfacingbat.AddItem deleteditem
        Itemadded = True
    End If
    If cmbNonfacingbat.List(j) = cmbFacingBat.Text Then
        cmbNonfacingbat.RemoveItem (j)
        deleteditem = cmbFacingBat.Text
    End If
Next j
End Sub
```

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

```
Private Sub cmdselectback_Click()
Me.Hide
```

Hides the window

```
Settings.Show
End Sub

Private Sub cmdSelectdone_Click()
If cmbFacingBat.Text <> "" And cmbNonfacingbat.Text <> "" And cmbOpeningbowl.Text <> "" Then
   OpeningBatsman = cmbFacingBat.Text
   OpeningNonfaceBat = 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
```

This sets the variables from the options selected, validating all options have been selected.

```
Private Sub Form_Load()
SummaryPath = App.Path & "\ScoreFiles\" & Trim(MatchStart.txtOpposition.Text) &
MatchStart.cmbDD.Text & MatchStart.cmbMM.Text & MatchStart.cmbYY.Text & ".txt"
StatisticPath = App.Path & "\ScoreFiles\" & Trim(MatchStart.txtOpposition.Text) & MatchStart.cmbDD.Text
& MatchStart.cmbMM.Text & MatchStart.cmbYY.Text & "Stats.txt"
Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
   If OpeningBatteam = "BGS Staff" Then
   BattingTeam = "BGS Staff"
   BowlingTeam = OpposingTeam
      For i = 1 To 11
         Get #1, i, PlayerMatchData
            cmbFacingBat.AddItem PlayerMatchData.MatchPlayerName
            cmbNonfacingbat.AddItem PlayerMatchData.MatchPlayerName
         Get #1, i + 11, PlayerMatchData
            cmbOpeningbowl.AddItem PlayerMatchData.MatchPlayerName
      Next i
   ElseIf OpeningBatteam = OpposingTeam Then
   BattingTeam = OpposingTeam
   BowlingTeam = "BGS Staff"
      For i = 1 To 11
         Get #1, i + 11, PlayerMatchData
            cmbFacingBat.AddItem PlayerMatchData.MatchPlayerName
            cmbNonfacingbat.AddItem PlayerMatchData.MatchPlayerName
         Get #1, i, PlayerMatchData
            cmbOpeningbowl.AddItem PlayerMatchData.MatchPlayerName
               Next i
      End If
   Close #1
   If i = 1 Then
      cmdselectback.Enabled = False
   End If
   End Sub
```

When the form loads a list is generated from the summary file to create the combo box lists.

This is the StartGame window and contains just one command button

cmdStartstart

```
Private Sub cmdStartstart_Click()
Me.Hide
MatchStart.Show
End Sub
```

Brings up the match start window to start a new game.

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
```

Hides the window

```
Private Sub cmdWicketdone_Click()
Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
   If optWicket(4).Value = True Then
      RunOut.Show
   Else
```

If run out is selected, shows the run out window

```
LastBatName = Scoreboard.lblLastManName
LastBatFow = Scoreboard.lblLastManFow
LastBatTotal = Scoreboard.lblLastManTotal
LastBatHowOut = Scoreboard.lblLastManHowOut
BatOut = CurBat
For i = 0 To 10
   If optWicket(i).Value = True Then
      Scoreboard.lblLastManHowOut = optWicket(i).Caption
      WicketType = optWicket(i).Caption
   End If
Next i
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
If WicketType <> "Retired" Then
   Scoreboard.lblWickets.Caption = Scoreboard.lblWickets + 1
End If
Scoreboard.lblLastManName = CurBat
Scoreboard.lblLastManFow = Scoreboard.lblRunstotal
If Wicket.chkBatcrossed.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(1) / Len(PlayerMatchData)
   Get #1, i, PlayerMatchData
      If PlayerMatchData.MatchPlayerName = BatOut Then
         PlayerMatchData.BattingHowOut = Scoreboard.lblLastManHowOut.Caption
         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.Bowlingwkts = PlayerMatchData.Bowlingwkts + 1
      End If
   Put #1, i, PlayerMatchData
Next i
For i = 1 To LOF(1) / Len(PlayerMatchData)
   Get #1, i, PlayerMatchData
      If PlayerMatchData.MatchPlayerName <> Otherbat And PlayerMatchData.MatchPlayerName <>
CurBat And Trim(PlayerMatchData.BattingHowOut) = "" And Trim(PlayerMatchData.MatchPlayerTeam) =
Trim(BattingTeam) Then
         NextPlayer.cmbNextPlayer.AddItem PlayerMatchData.MatchPlayerName
      End If
Next i
Close #1
Call EndBall(0)
If Scoreboard.lblWickets = 10 And Secondinnings = False Then
   Call EndInnings
ElseIf Scoreboard.lblWickets = 10 And Secondinnings = True Then
   If BowlingTeam = "BGS Staff" Then
```

Sets the last man information on the scoreboard.

Resets the scoreboard for the batsman who is out.

Validates the wicket was caught if the batsman cross.

Updates the summary file with the wicket information

Generates the next player list.

Calls 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
      End If
Me.Hide
End Sub
```

Sets up the nextplayer window.

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
   Bowlingwkts As Integer
   Bowlingmaidens As Integer
   Listed As Boolean
End Type

Public Type TeamSummary
```

Sets up the Master file variables

Sets up the summary file variables

```
    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 SearchDone As Boolean
Public Teamsearch As Boolean
Public OpposingTeam As String
Public MatchMonth As String
Public MatchDay As Integer
Public i As Integer
Public Runsperwnb 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
```

Sets up the team summary

Sets up the statistic file

```
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
   Put #1, i, PlayerMatchData
Next i
Close #1
Maiden = True
NextPlayer.Show
   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 = Extratype
      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
```

Adds one over to the bowler and the maiden

Swaps the batsmen.

Updates the statistics file for that ball

```
        Else
            .Wicket = ""
        End If
    End With
    Put #1, I, 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.Overs = Scoreboard.lblOversBowled.Caption
    Put #1, 50, TeamData
Unload Scoreboard
OpeningBatteam = 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.lblOversBowled) * 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)
```

Ends the innings

Resets the scoreboard to start a new innings

Then runs start players again

Ends the match

```
Open App.Path & "\Cricketmasterfile.txt" For Random As #3 Len = Len(PlayerRecord)
For i = 1 To 22
   Get #2, i, 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
               .bowlingruns = .bowlingruns + PlayerMatchData.bowlingruns
               If PlayerMatchData.Battingruns > .Highscore Then
                  .Highscore = .Runs
               End If
               If PlayerMatchData.BattingBallsFaced <> 0 Then
                  .Innings = .Innings + 1
               End If
               .Maidens = .Maidens + PlayerMatchData.Bowlingmaidens
               If PlayerMatchData.BattingBallsFaced <> 0 And Trim(PlayerMatchData.BattingHowOut) = ""
Then
                  .NotOuts = .NotOuts + 1
               End If
               .Overs = .Overs + PlayerMatchData.Bowlingovers
               .Wickets = .Wickets + PlayerMatchData.Bowlingwkts
               .AverageRuns = .Runs / (.Innings - .NotOuts)
               .econrate = .bowlingruns / .Overs
               .strikerate = (.Overs * 6) / .Wickets
            Put #3, j, PlayerRecord
            End With
         End If
      Next j
Next i
Close #2
Close #3
Scorecard.Show
Call UpdateFile
End Sub
```

Updates the master file with statistics for each player from the match

Shows the scorecard then calls the module to create the spreadsheets

```
Public Sub PreEndOver()
If Scoreboard.lblOversBowled + 1 >= LimitedOvers Then
   If Secondinnings = False Then
      Call EndInnings
   Else
      If BowlingTeam = "BGS Staff" Then
         k = 1
      Else
         k = 2
      End If
      Call EndMatch(k)
   End If
Else
NextPlayer.lblNextPlayertext.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 <> CurBowl And
Trim(PlayerMatchData.MatchPlayerTeam) = Trim(BowlingTeam) Then
               NextPlayer.cmbNextPlayer.AddItem PlayerMatchData.MatchPlayerName
```

Calls the end of the innings if the number of limited overs has been reached.

Sets up the next player window for bowlers for the next over.

```
            End If
        Next i
        NextPlayer.cmdProlong.Visible = True
        j = 0
        NextPlayer.Show
End If
End Sub

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

Public Sub UpdateFile2()
If Dir(App.Path & "\Spreadsheets\" & OpposingTeam & MatchStart.cmbYY.Text & ".xlsx") <> "" Then
    Kill App.Path & "\Spreadsheets\" & OpposingTeam & MatchStart.cmbYY.Text & ".xlsx"
End If
    Set objExcel2 = CreateObject("Excel.sheet")
Open SummaryPath For Random As #1 Len = Len(PlayerMatchData)
    For i = 1 To 22
```

Makes the prolong button visible then shows the window

This routine deletes the existing master file spreadsheet and creates a new master file in excel.

```
        Get #1, i, 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) = .Bowlingwkts
            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.cmbYY.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.cmbYY.Text & ".xlsx"
Close #1
End Sub
```

This then creates a spreadsheet for the match that has been played.

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:

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

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.

Requirement Spec	Met?	Proof	Comments
The system will ask for the names of all the players at the start of the match for at least the BGS staff team.	Yes	SS 5 & SS 6	The system will also ask for the opponents names straight away.
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	No	None	I was not able to implement a feature to allow the user to change the name of players during 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.	Yes	SS 17 - 22	This requirement was fully met.
The system will add runs to the total whenever they are inputted	Yes	SS 17 - 22	This requirement was fully met.
The system will add runs to the correct section of the scoreboard (either batsman or the extras total)	Yes	SS 17 – 22, SS 29 & 30	This requirement was fully met.
When a wicket is taken the scoreboard will add that wicket on to the total number of wickets	Yes	SS 39 - 41	This requirement was fully met.
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	Yes	SS 40	The user has to choose from the list of players created at the start of the match.
The system will have a separate window with a full scorecard of the whole match	Yes	SS 45	This requirement was fully met.
The scorecard will display 2 tables per innings: one table containing the batting figures and the other the bowling figures.	Yes	SS 45	This requirement was fully met.
The batting table will show the runs scored by a batsman, and if that batsman is out; the method by which they were out.	Yes	SS 45	This requirement was fully met.
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.	Yes	SS 45	This requirement was fully met.

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.	Yes	SS 49 & SS50	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 confidant at programming using random access files so 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 straight forward 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 scenario's (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:

```
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 10
    With PlayerMatchData
      .MatchPlayerID = i + 1
      .MatchPlayerName = "BGS " & i
      .MatchPlayerTeam = "BGS Staff"
      .Battingruns = 0
      .BattingBallsFaced = 0
      .BattingHowOut = ""
      .BattingWktBowler = ""
      .Bowlingovers = 0
      .Bowlingruns = 0
      .Bowlingwkts = 0
      .Bowlingmaidens = 0
    End With
  Put #1, i + 1, PlayerMatchData
  Next i
  For i = 0 To 10
    With PlayerMatchData
      .MatchPlayerID = i + 12
      .MatchPlayerName = "Opp " & i + 1
      .MatchPlayerTeam = "Test Opp"
      .Battingruns = 0
      .BattingBallsFaced = 0
      .BattingHowOut = ""
      .BattingWktBowler = ""
      .Bowlingovers = 0
      .Bowlingruns = 0
      .Bowlingwkts = 0
      .Bowlingmaidens = 0
    End With
  Put #1, i + 12, PlayerMatchData
  Next i
Close #1
Extraball = True
Runsperwnb = 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 scorecard window.
- The scorecard 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 unneccessarily. 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 I, j, k, l and p. Using l 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 batsman. 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 notation 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:

<u>Editing a ball</u>

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 the 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.....
Runs.....
Extras .....
Wicket .....
End of Over .....
Short Run & Undo .....
End of Innings .....
End of the Match .....
Spreadsheets .....
Scorecard.....
3 Problem Solving .....
3.1 Error Message .....
3.2 Troubleshooting .....
3.3 Contact Details .....
4 Back-up Routines .....
5. Glossary.....
6. 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 new game.

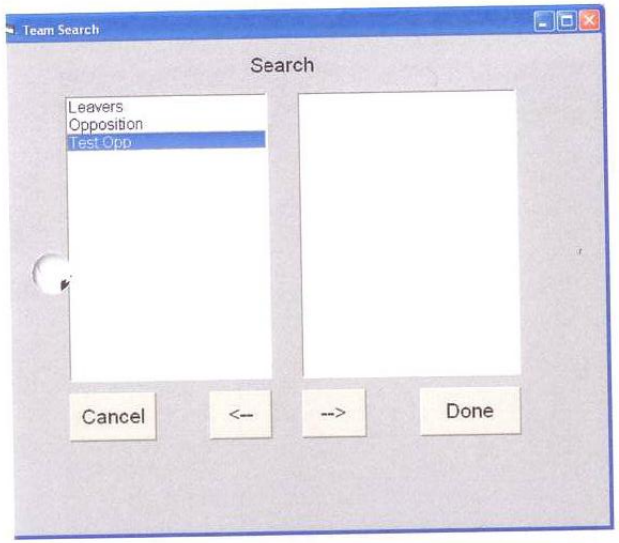
The following window will then come up:

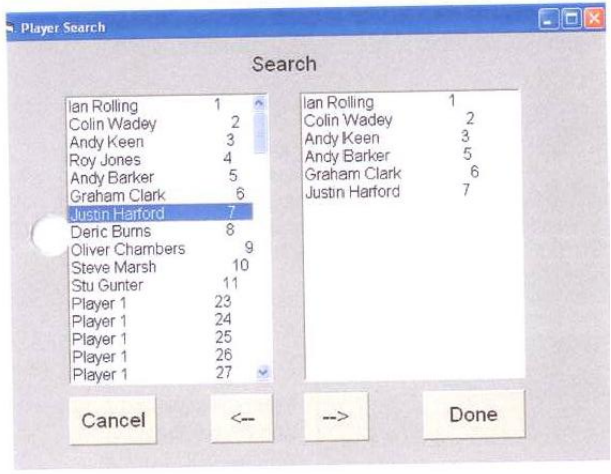
Firstly enter the date at the top, select the Month first to make a list of days appear.

Then you will need to select your opposition. If you have played the opposition before using this system 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.

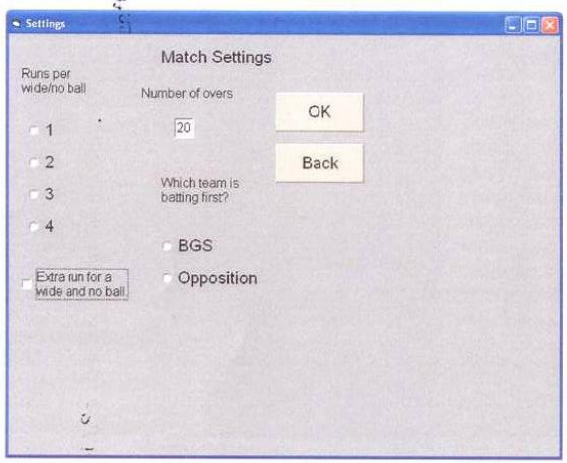
You will then need to select your players. For the BGS side press the Search BGS Players button, this will bring up the search window again listing players who have played for BGS through this system previously. Select these players, and type in the names of any remaining players in the remaining boxes.

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 int



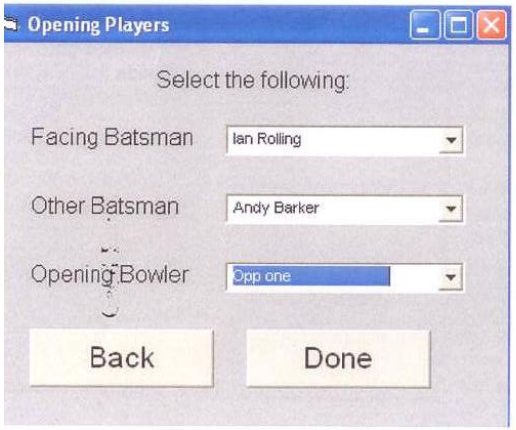


Once both teams have all players filled, press Done. The settings window will then come up:



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.

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

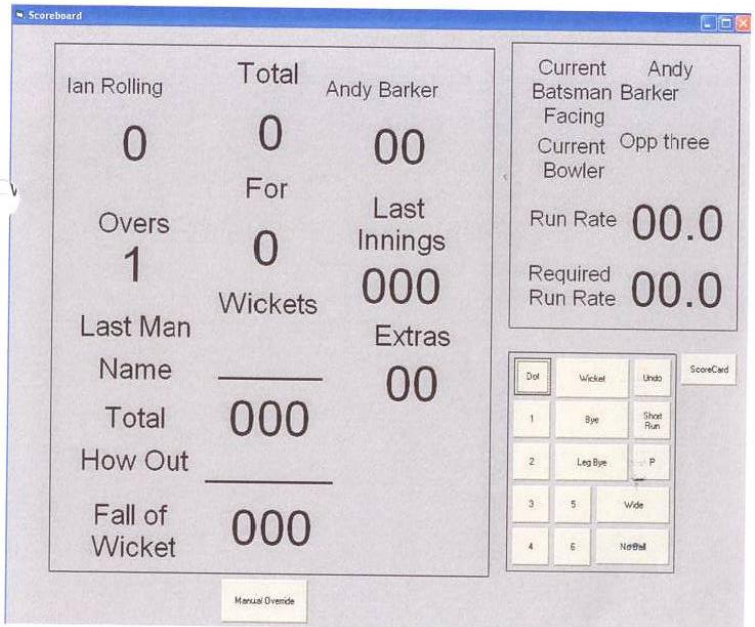


On this window simply select the facing batsman, non facing batsman and opening bowler from the lists. Then press Done.

(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.



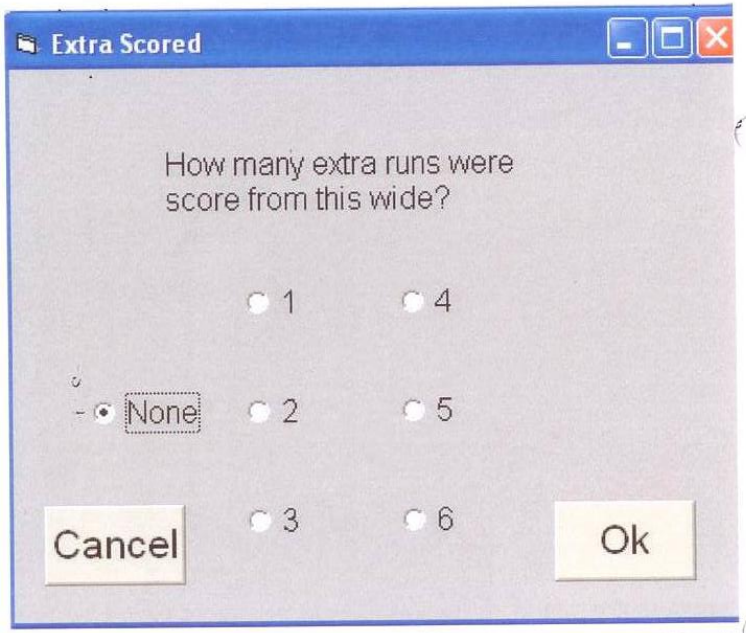
The Scoreboard more or less represents a real life scoreboard; it is real time meaning that it will be updated as soon as a ball has taken place. The buttons is the bottom right box give the options for what can happen every ball.

Runs

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

Extras

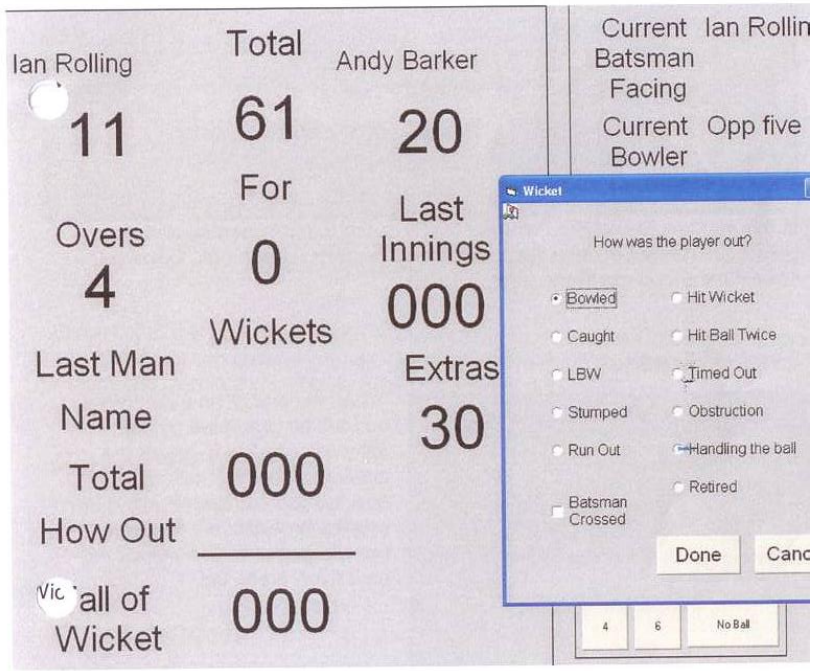
For any extra press the relevant extra button. Once you press any of these extras, the following window will appear:



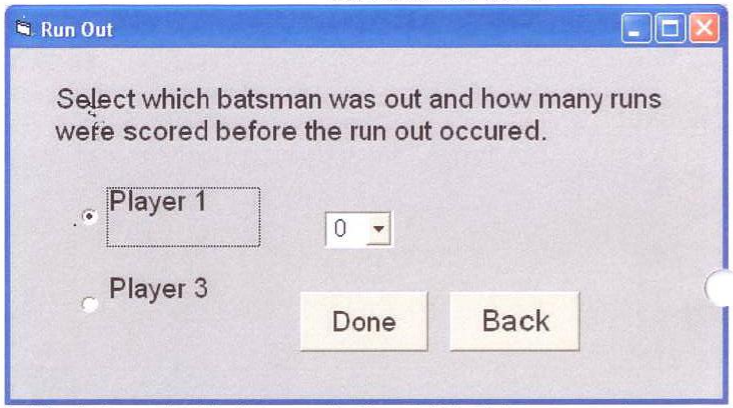
On this window simply select how many additional runs were scored from the extra.
(Note for a wide/no ball enter the additional runs scored by the batsman, not the initial runs scored from the wide/no ball itself).
Once the number of runs is selected, press Ok
(Note on a no ball you will be asked if 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:

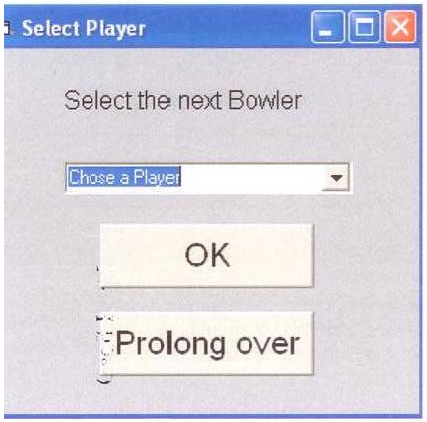


Select the method by which the batsman was out. If the batsman was caught and the batsman crossed in the process, check the batsman crossed box. Once you have done this, press Done. If a Run Out has occurred then an extra window will come up:



In this window, select the batsman that is out. If the batsmen scored runs before the run out occurred, enter this into the runs scored box. Once you have done this press Done.

End of Over



When the end of an over comes, you will be prompted by the following window to select the bowler to bowl the next over. If needed you can also Prolong the over by an extra ball. Once you have selected the bowler for the next over, press OK.

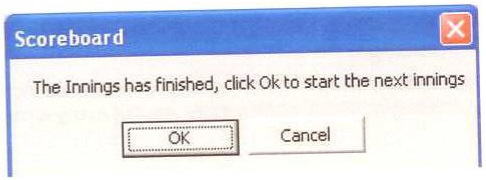
Short Run & Undo

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.



Press Ok to this pop up, and you will then need to select the Opening batsmar 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 oliver.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 Scorefiles 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
  Over 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

OCR
RECOGNISING ACHIEVEMENT

H447 Unit F454: Computing Project

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 top 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. [3/5]

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 [4/5]

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. [13/16]

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. [14/14]

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. [9/10]

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. [4/4]

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]