Criterion A1: Analysing the Problem

Introduction

Emirates International School - Jumeirah (EIS-J) is the first school in the United Arab Emirates to offer the International Baccalaureate Diploma Programme (IBDP). The school, also, has an extensive After-School Activities Programme, which a large number of students participate in to benefit from the various skills to be acquired from each activity. The EIS-J Ecology Club has been in charge of environmental awareness in the school since the school's birth in 1991. Mrs. Mahrukh Cooper is the current club co-ordinator and I am the student co-ordinator, Harris Rasheed. The club is responsible for various events and activities throughout the year. This includes recycling used-material like paper and plastic, the annual Earth Week and the end-of-year party to celebrate the achievements and dedication of our volunteer recyclers to environmental causes. In addition, various competitions with eco-friendly themes are held regularly throughout the year for students to take part in. Competition entries are posted on the school's recycling board that many teachers and students admire. The club, also, has its own website that is updated regularly and has registered over 3000 hits. It can be visited on http://www.eisecoclub.piczo.com (see Appendix).

The Problem at Hand

A lot of paperwork is handled by the club and this is performed by Mrs. Cooper using a Microsoft Access 2007 database. She finds the job very time-consuming and tiring to carry out, ironically, despite the current use of information technology. After a lengthy discussion I had, to discourage her from resigning as the club co-ordinator, she agreed to resume her post if I was able to produce a more efficient system that would considerably reduce the workload and paperwork involved.

The Existing System (Pros & Cons)

After a chat with Mrs. Cooper about the problems she was encountering and a thorough look at the Access database used, I was able to deduce the benefits and problems of the system. A factor that greatly aided me in this case was that I had carried out a system analysis on the existing recycling monitoring system about a year ago so I knew how it worked inside-out. I had created the Access database system that she currently uses but I did not know its flaws because I could not find time in my previous system analysis to evaluate the success of my database.

Mrs. Cooper complimented some aspects of the database over the previous system.

She said that the data stored in the Access database was more secure than before because it was password protected and encrypted. This prevented unauthorised access to personal data and records. The previous system did not have any security.

The database also processed information faster than the systems used previously. This cut down a lot of time and frustration for many tasks that had to be carried out manually before.

Information in the database was stored altogether so it made it much easier to access than before where files were stored in different places of the computer's directory and had to be found first before they could be accessed. Another advantage of storing the data together was that searching was easier since queries could be passed using fields from different tables. Once a criterion was specified, the resultant data could be immediately viewed.

Data could be stored in the database in the form of look-up tables, which made it easier for other tables to use and process this data. Any amendments to data in the look-up tables would result in automatic updation in the tables linked to it.

The report format, produced by the database, was very beneficial. This allowed information to be output faster in a formal, easy-to-read layout.

On the other hand, the existing Microsoft Access database was unsatisfactory for a few reasons.

One of the reasons is that it cannot tally the number of times a recycler has participated from the attendance records stored in the database. This creates a problem at periodic intervals when Mrs. Cooper needs a report of volunteer recycler attendance in a tally format so that she can give out 'Recycler of the Month' certificates. Due to this problem, Mrs. Cooper has to count the number of times a recycler is mentioned in the report that is produced by the database (the semantics of this report makes it very unhelpful). A separate report then has to be created manually that contain the tallied records. This is a very time-consuming task, especially at the end of the year, when the entire academic year's worth of records has to be tallied. It is also a very frustrating task because if the count is lost anytime, it has to be repeated from the start. Mrs. Cooper also refuses to delegate this task to anyone because of its high-priority but the frustration of the task still exists.

In addition, Mrs. Cooper finds the database complicated to use because of the numerous buttons, prompts and technical procedures involved in using it. She finds it very difficult to manage if something goes wrong. In such cases, she waits for me to come to her classroom to address the issue. This stops work for days at times.

All database procedures have to be executed manually because they are in the form of queries. There are three methods used to process recycling activity data (add -> save -> delete). Once data is input, the procedures have to be found and then executed in the set order. If the procedures are not run in this order, the data might be deleted before it is processed with other tables.

Forms (input screens) in the database are not very user-friendly. This is because its layout and format does not allow quick input, which is an important factor of an information system. Therefore, the forms are unused despite their attractive designs. Data, simply, is input directly into the tables because it is faster. This destroys the purpose of having forms in the database and in turn is a waste of space.

The vast amount of tables makes it difficult to understand where various pieces of information are stored. Despite the lengthy titles given to each table, it takes time even for frequent users, to figure out the appropriate table to access. Tables also take time to load for common processes like opening, saving and deleting which slows down the entire process.

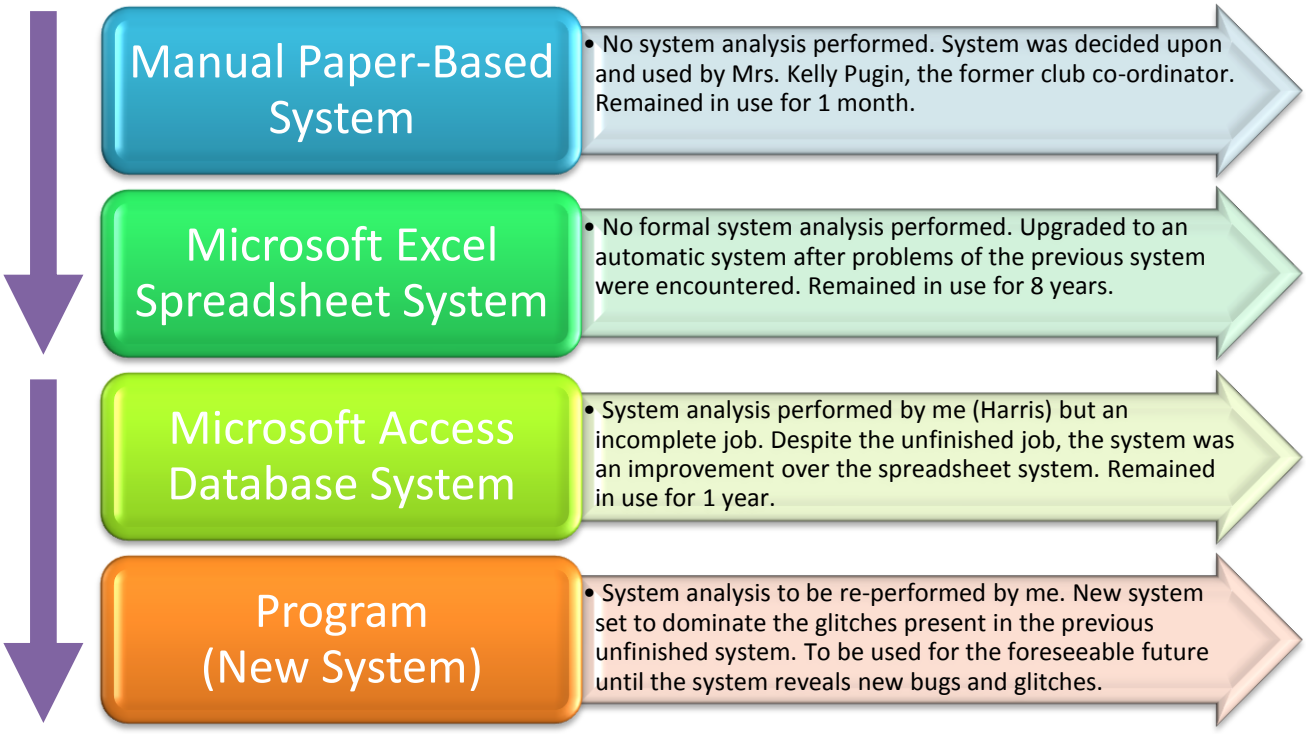
The size of the database is quite large (32.6MB). This results in long periods of time taken to transfer the database e.g. from a computer to an external data storage device, which is done on a very regular basis, to move it between home and school.

The database also crashes at times because it uses up large amounts of system memory to carry out its own simple procedures (open, save and delete).

After her evaluation of the current system, I realised that the new system would have to retain the benefits of the current system as well as eradicate the problems of the current system.

The System's Life Cycles

The first system used by the club was completely paper-based and it hindered efficiency because of the typical problems a manual system causes. This, soon after its implementation, was upgraded to a Microsoft Excel system which availed all the prospects that IT offers for these type of systems. The Excel system was very basic; it lacked the use of formulas and other advanced spreadsheet features, yet it was used for a long period of time because of the vast improvement over the first manual system. Once I joined the Eco Club and got more involved, I started to handle some of the club's paperwork as an extra-curricular activity. I decided to carry out a system analysis in 2008 on the Excel system to further reduce the work required and to increase efficiency. Despite the system analysis, some problems were unaddressed and remained present because of the lack of time I had to fix it as well as no clear, developable solution. After a year of the Access database's implementation, it was clear that the system would have to be re-designed to solve the problems or else, suitable alternatives would have to be found. Since I could not think of a way to alter the database so that it will monitor attendance and produce the desired output, I thought that it was time for a change.



Sample Data

There are two documents that recycling statistics are recorded on.

The first is the 'Thursday Recycling Rotas' slips. This is used by recyclers to keep a record of classrooms that they visit on their recycling collection rounds every Thursday during lunch break. The data from this document is input directly into the Access database table. After input, the data is processed and stored. Eight copies of this slip are issued every week; each copy is given to a team of recyclers who visit one of the school's eight building areas. An example of this form can be seen below.

The form above details the rooms visited in the recycling collection by Saif Qassim and Omar Rasheed on Thursday 19th March 2009. The fields on the form are self-explanatory. 'HS48' in the 'Rooms Recycling Collected' field is circled by the recycler because it had the most recycling material. This indicates that it should receive more points (part of the recycling point system).

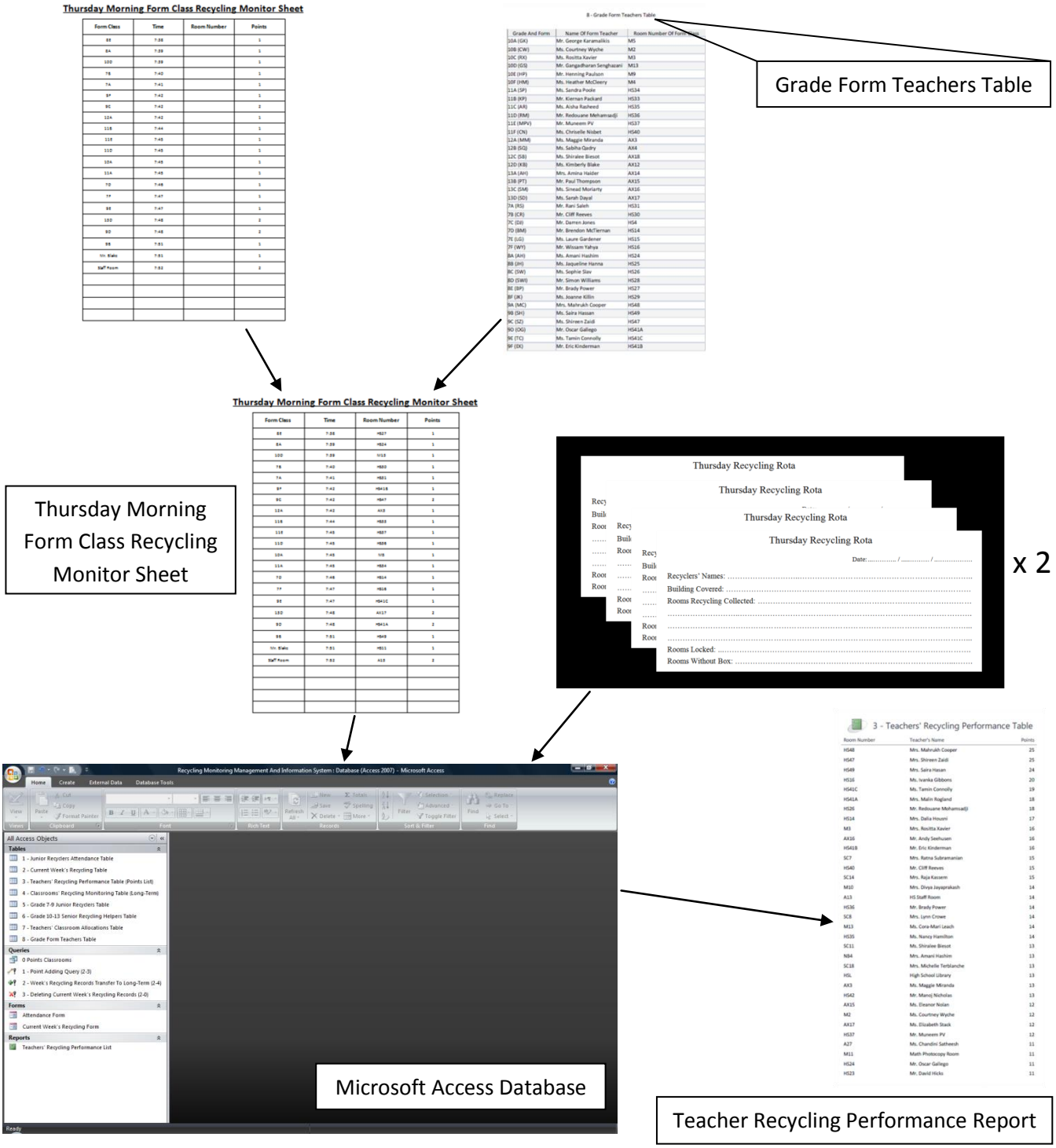
The second form is the 'Thursday Morning Form Class Recycling Monitor Sheet'. This form is used every Thursday morning by the recycling skip supervisor. Every form class sends a group of students to carry the class' recycling box to the drop-off point (otherwise known as the recycling skip). When the students get there, they tell the recycling skip supervisor the grade and division of the class they are representing (usually their own). The person on recycling duty notes this data on the form along with the time and number of points. The form is then cross-referenced with the 'Grade Form Teachers Table' before it is input and processed by the database. The form can be found on page 5.

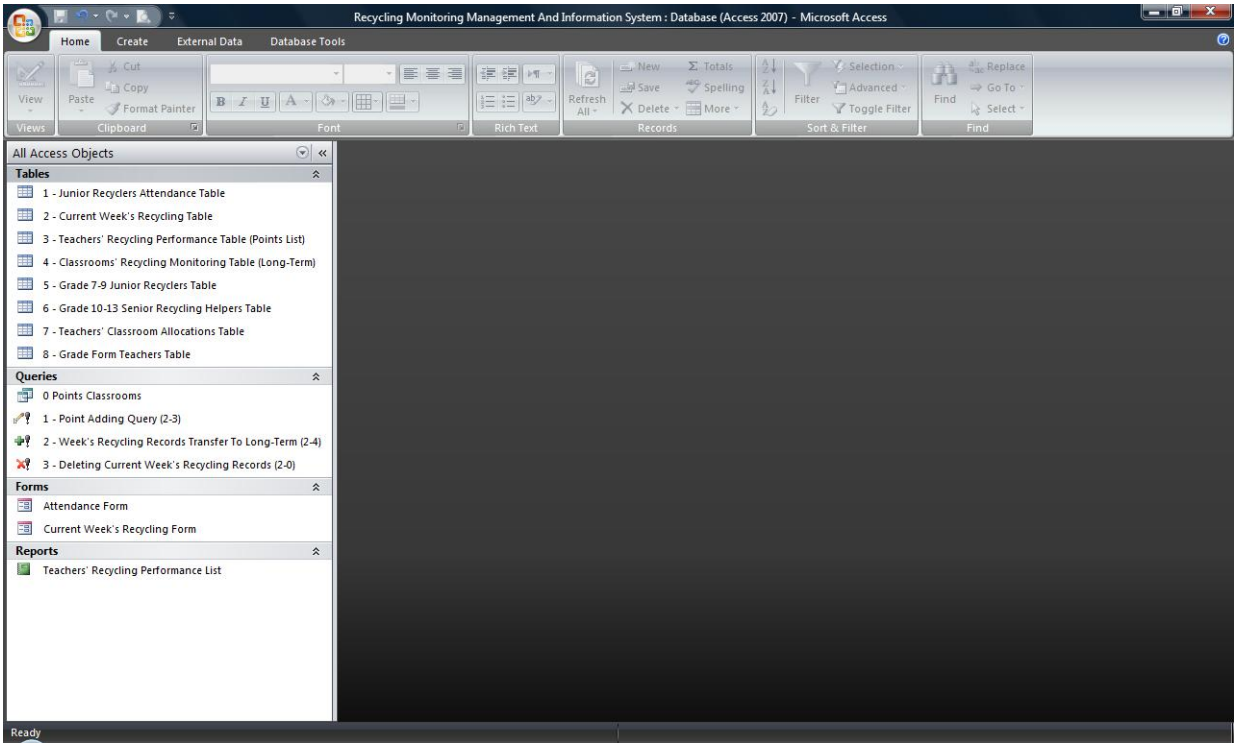
The 'Grade Form Teachers Table' is a data sheet and also a look-up table in the database. It can be found on page 6.

The form on page 5 is partially filled out <u>at the time of data collection during recycling duty</u>. The 'Form Class', 'Time' and 'Points' columns are completed but the 'Room Number' column is left blank. This field is required in the input process so the 'Grade Form Teachers Table' data sheet is used to cross reference each form class with the room number it is situated in every morning.

The System Architecture

The following diagram demonstrates the flow of data throughout the system from start to finish.





The screenshot above displays the elements/objects of the Access database. I will now explain the purpose and use of each element (tables, queries, forms, reports).

Tables

- *Junior Recyclers Attendance Table*
  This table stores the history of recycler attendance for every Thursday that recycling was collected. It stores the names of the volunteer recyclers as well as the blocks that they have collected recycling from. Simply, it is used to backtrack previous recycling collections if required. This sheet was never printed out by the user because the semantics of the table design prevented the desired output from being possible.
  In the new system, the table design and system processes will have to be altered so that the desired output can be obtained (tallied format). If this is achieved, the main disadvantage of the current Access system will be solved.

- *Current Week's Recycling Table*
  This table is used to temporarily store the recycling data of the current week's output. Data here is usually stored for a few minutes before it is processed. Processing is initiated manually by the user with the use of queries. Once the processes occur, the data in the table is deleted by one of the queries. Data from the two forms (Thursday Morning Form Class Recycling Monitoring Sheet & Thursday Recycling Rotas) is input into this table for processing.
  It is hoped that there will be no need for this table in the new system and that it will be replaced by a user-friendly input screen that will perform all the processes on completion.

- *Teachers' Recycling Performance Table (Points List)*
  This table is a point list of the recycling performance of each classroom in school. It details the recycling activity of each teacher throughout the year. Points from the two forms are processed with this table.
  In the new system, this is to be left unchanged because there are no issues with this area of the current system. It is still a necessary element of the new system.

- *Classrooms' Recycling Monitoring Table (Long-Term)*
  This table is simply a transaction log of all the recycling activity that has occurred. It is very similar to the "Junior Recyclers' Attendance Table" except it stores recycling statistics. The reason for this is that it is simply a reference sheet that can be used when there are any confusions or problems with the "Teachers' Recycling Performance Table". This table has never been printed.
  This is hoped to be refined in the new system so that it presents the data when it is needed. Printouts will not be required.

- *Grade 7-9 Junior Recyclers Table*
  This table stores the names and details of all the registered recycling volunteers. The names in this table are important because it is used as a look-up table by the "Junior Recyclers' Attendance Table" so that recyclers' names are available during input which makes the process faster. Recycling volunteers are registered through the Recycler Registration Form.
  This is to be kept the same in the new system but a new input interface is anticipated.

- *Grade 10-13 Senior Recycling Helpers Table*
  This table stores information on the senior Eco Club volunteers that help with the administration work of the club. It outlines what work they are willing to carry out in addition to their e-mail addresses for contact purposes.
  In the new system, this table is redundant because it is not a necessary element of the monitoring process. It can be created on a different application because it is not part of the recycling monitoring process but is simply there to store all the Eco Club's information together.

- *Teachers' Classroom Allocations Table*
  This table is a reference sheet where the room numbers of each classroom is detailed along with the teacher that uses it and whether it is an office or not (offices have higher priority for recycling collection than classrooms because there are no students to take the recycling). This information can be obtained from the School Classroom Project.
  In the new system, this will have to be stored as a look-up table (the same as before).

- *Grade Form Teachers Table*
  This table is a reference sheet for the "Thursday Morning Form Class Recycling Monitor Sheet". It is used to cross-reference details before input. It contains information on the location of each form class during morning registration and the name of the teacher who supervises it.

To reduce work in the new system, the cross-referencing stage will be automated. This means that this table will now become a look-up table in the system. It was initially stored simply to keep the Eco Club's information together.

Queries

- *0 Points Classrooms*
  This query is used to track classrooms that have no recycling activity. The reason for this is usually because they don't have a recycling box. When boxes are available, they are distributed to these rooms to increase the school's recycling activity.
  There is no need for this query in the new system as this was rarely used in the current system. The bottom part of the Teachers' Recycling Performance List can be viewed which are basically the records that would be part of the 0 Points Classrooms query result.

- *Points Adding Query*
  This query is the first of the three methods that are used in the database. It processes the information in the "Current Week's Recycling Table" with the "Teacher's Recycling Performance Table (Point List)" so that the points for the current week's output are added to the total point list.
  This will hopefully be made automatic in the new system to avoid the complication of running each query in the set order.

- *Week's Recycling Records Transfer To Long-Term*
  This query is the second of the three methods. It involves processing of "Current Week's Recycling Table" with "Classrooms' Recycling Monitoring Table (Long-Term)". In short, it is a transaction log of the "Point Adding Query". The process allows for later backtracking of recycling activity statistics.
  This is also hoped to be made automatic in the new system.

- *Deleting Current Week's Recycling Records*
  This query is the last of the three methods. The procedure simply clears all the data in the "Current Week's Recycling Table" so that new data can be entered into it for the next week. This is also to be made automatic in the new system if possible.

  The three queries above had to be created in this way because there was no alternative solution in Microsoft Access 2007 database. The new system will hopefully be able to replace this with the processor's memory making the process much faster.

Forms

- *Attendance Form*
  This form is the front end of the "Junior Recyclers Attendance Table". Despite its attractive design and many features, it is unused because input is quicker through the table's datasheet view.

In the new system, it will have to be user-friendly and allow quick user input.

- *Current Week's Recycling Form*
  This form is the main interface of the "Current Week's Recycling Table". This is similar to the "Attendance Form" because it is unused due to the user input limitations.
  The new system will also have to allow easier input for this form.

Reports

- *Teachers' Recycling Performance List*
  This report is printed periodically and posted in the staff room for the teachers to see. This encourages competition and increases recycling activity.
  This is an important report and it will have to be included in the new system. Changes are not necessary as the current report format is adequate.

System Input & Output

There are two types of data that is to be input into the system. They are:

- Continuous data
- Pre-set data

Continuous data flows through the system and is processed on a regular basis. Continuous data comes from the two forms (Thursday Morning Form Class Recycling Monitoring Sheet & Thursday Recycling Rotas). This data enters the system every week and is processed to display the final output.

Pre-set data, on the other hand, is set into the system and usually remains unchanged during the year. This data is used as a reference for the continuous data during processing. Pre-set data comes from the 'Form Teacher Allocation Sheet', which goes into the "Grade Form Teachers Table". Data on this sheet can possibly change from time to time throughout the school year but it seldom does. This is obtained from the Dean of Studies of each year group. The 'School Classroom Project' is another source of data that is input into the system at the beginning of every school year. This assignment is given to Senior Eco Club volunteers who collect data from around the school. The information is input into the "Teachers' Classroom Allocations Table" and rarely ever changes. The last and final document is the 'Recycler Registration Form'. This form is used by new recyclers to have their name registered into the Eco Club system. All new volunteer recyclers have to fill out this slip. This data is input into "Grade 7-9 Junior Recyclers Table". The bulk of this data is entered at the beginning of each academic year however it is common to find a registration at any time of the year.

Data from the system is usually output in the form of printed reports or displayed visually. The system produces three different output reports.

The first is the "Junior Recyclers' Attendance Table". Although the table is not in tally form as desired, it is still output. The second is the "Teacher's Recycling Performance List" report. The third and final is the "0 Points Classroom". This is simply noted down.

Criterion A2: Criteria for Success

A New System

With the analysis of the current system at hand, some of the objectives of the new system are clear so that the current system's problems can be avoided.

Goals & Objectives

The new system will have to be able to:

- produce a *tally report* of recycler attendance;
- produce a *points list* of teachers' recycling performance;
- hold data securely and *prevent unauthorised access*;
- *allow users to edit* the pre-set data *easily* which is used for look-up;
- produce a *list of candidates* for the Recycler of the Month award for any given month;
- use *simple user-friendly interfaces* that do not confuse the user and allows quick input;
- *process the input data automatically* without having to initiate each procedure manually in turn;
- take up *less space* on the computer hard disk;
- *keep a record* of all the recycling activity statistics that have been processed;
- *handle errors* and any sort of extreme/invalid data input.

If the new system is able to fulfil all of the above objectives, it can prove to reduce a large amount of the workload that was involved in using the previous systems.

I will now discuss how each of the objectives above can help solve the problems with the current system as well as maintain its advantages.

If the system:

- produces a tally report of recycler attendance, Mrs. Cooper doesn't have to count the number of times a recycler is mentioned in the "Junior Recyclers Attendance Table". This would reduce a great deal of frustration because the Recycler of the Month as well as Recycler of the Year can be chosen much more easily based on the greatest number of times that the recyclers have participated. Mrs. Cooper uses this figure along with the commitment and dedication they have shown during that time period.
- produces a points list of the teachers' recycling performance. This can be printed at intervals and posted in the staff room to ensure that teachers are kept up-to-date with the top recyclers. This objective would make it easier to take printouts of the point list and post it up.
- holds data securely and prevents unauthorised access, it will prevent outside unpermitted individuals from altering information or accessing data in the system.
- allows the user to edit pre-set data in the system easily, an alteration of data in the look-up tables will result in immediate updating.

- produces a list of candidates for Recycler of the Month award for any given month. This objective is an extension of the first objective. It will determine the candidates for the awards by analysing the statistics in the Recycler Attendance Report.
- include simple user-friendly interfaces that allow quick and easy input, Mrs. Cooper will be able to understand the system better. Technical errors can be avoided so there will be no delays in processing and inputting data. The interface will allow clear and direct actions to be taken.
- processes data automatically without having to initiate procedures manually, the time taken for the user to input data into the system and view the output will be reduced. This also makes the task simpler as the program handles all the processes in the appropriate order.
- takes up less space on the computer, the time taken to transfer the program from one place to another will be shorter and easier.
- keeps a record of processed recycling activity statistics, data can be backtracked in case there are any abnormal statistics in the point list perhaps because of unauthorised tampering.
- handles errors and extreme data, there will be fewer problems during the input stage for Mrs. Cooper and mistakes can be identified prior to the processing stage. She will be able to understand weird errors if errors are specifically output.

Limits of the Solution

The limits to the solution I have chosen are that:

- a printer will be needed to print out the reports;
- Java software will have to be installed on the systems it is used on for it to be able to run;

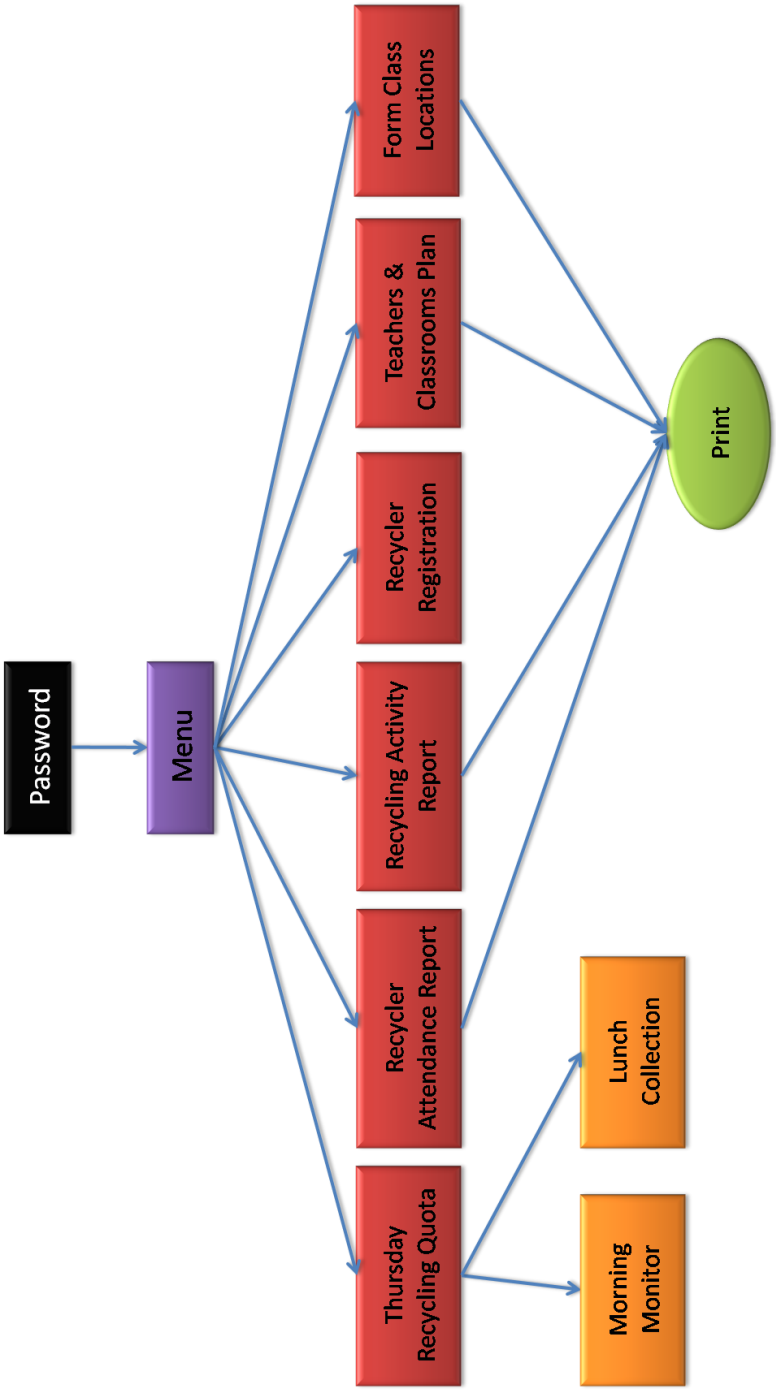
As can be seen, there are few limitations to the system because most of the requirements of the solution are already available from the previous system's requirements.

Criterion A3: Prototype Solution

Time to Design

With the goals and objectives of the new system and the analysis of the current system at hand, I set out to design the new system.

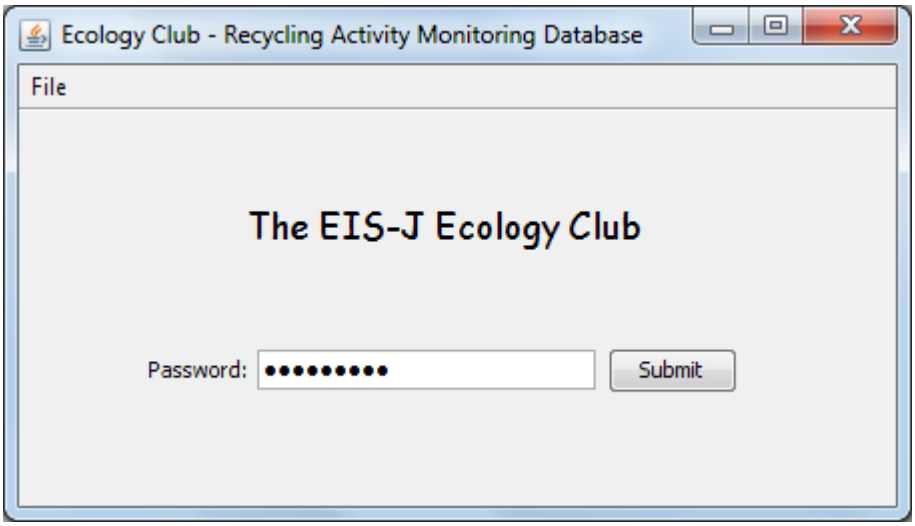
I planned the layout of the new system. The diagram below shows the user action flowchart that the new system will abide to.



Prototypes & Review

From the user-action flowchart on the previous page, I will create the prototype screens for my Recycling Activity Monitoring program. The following diagrams detail the prototypes I created for the new system and the discussion I had with the end-user, Mrs. Cooper, for each one.

Password



The 'File' menu will have only one menu item; Close.

Discussion

Mrs. Cooper: I like this screen. It's very simple yet elegant. Try adding a little colour of it if you have time?

Harris: Sure. I can add the club logo if you want and I'll also try adding a background colour.

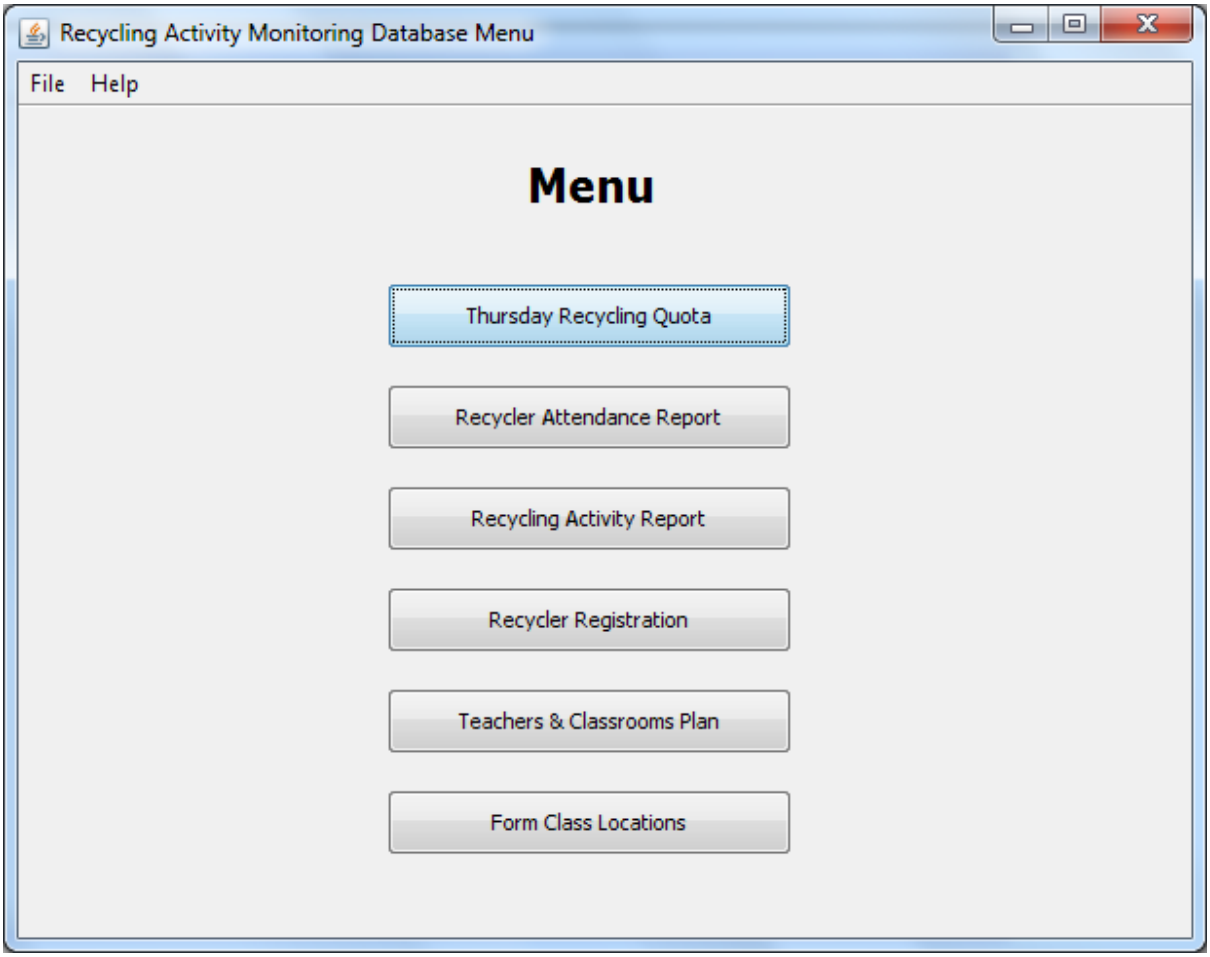
Mrs. Cooper: Sounds good. That reminds me... I tend to forget my passwords often. Is there anything you can do in case I don't remember the password? This has happened to me many times in the Access database. I forgot to mention this before during the problem analysis interview you conducted with me.

Harris: Hmm... Ok. How about I add a button saying 'Forgot Password' and allow you to answer a secret question of your choice whenever you want? I can also add a feature to let you change the password and secret question whenever you want.

Mrs. Cooper: Wow. Can you really do all that? I can't wait to see this new system. I hope this isn't going to take too much of your time to make it.

Harris: Don't worry. Since I've done a system analysis on this system before, I am very familiar with its processes which makes the job much easier.

Main Menu



The 'File' menu will have two menu items; Log out & Exit.
The 'Help' menu will have one menu item; About.

Discussion

Mrs. Cooper: Oh my goodness! This is such an improvement over the Access database. Everything is much clearer and simpler. Can you add a little colour to it though?

Harris: Ok. I will implement that with all the other screens as well. I also have to add another button to this screen to allow you to change the secret question and answer.

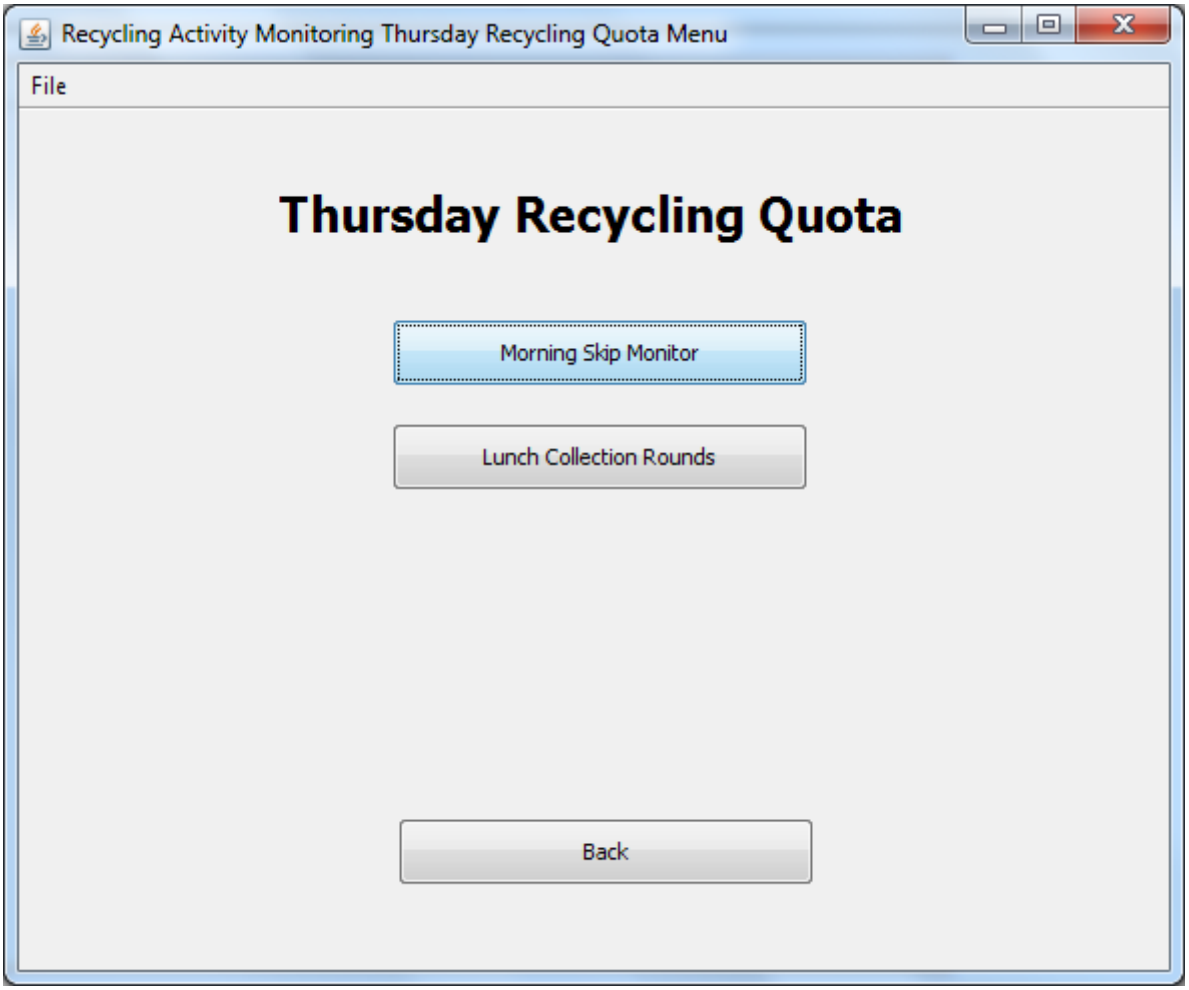
Mrs. Cooper: Yes. That would be very helpful. By the way... How would I get the list for the Recycler of the Month Candidates?

Harris: OH! I totally forgot about that for some reason. I'll add that button between the 'Recycler Registration' and 'Teachers & Classrooms Plan'.

Mrs. Cooper: Yes, please. Please show me the screen for that too when you have time to design it.

Harris: Ok.

Thursday Recycling Quota Menu



The 'File' menu will have two menu items; Log out & Exit.

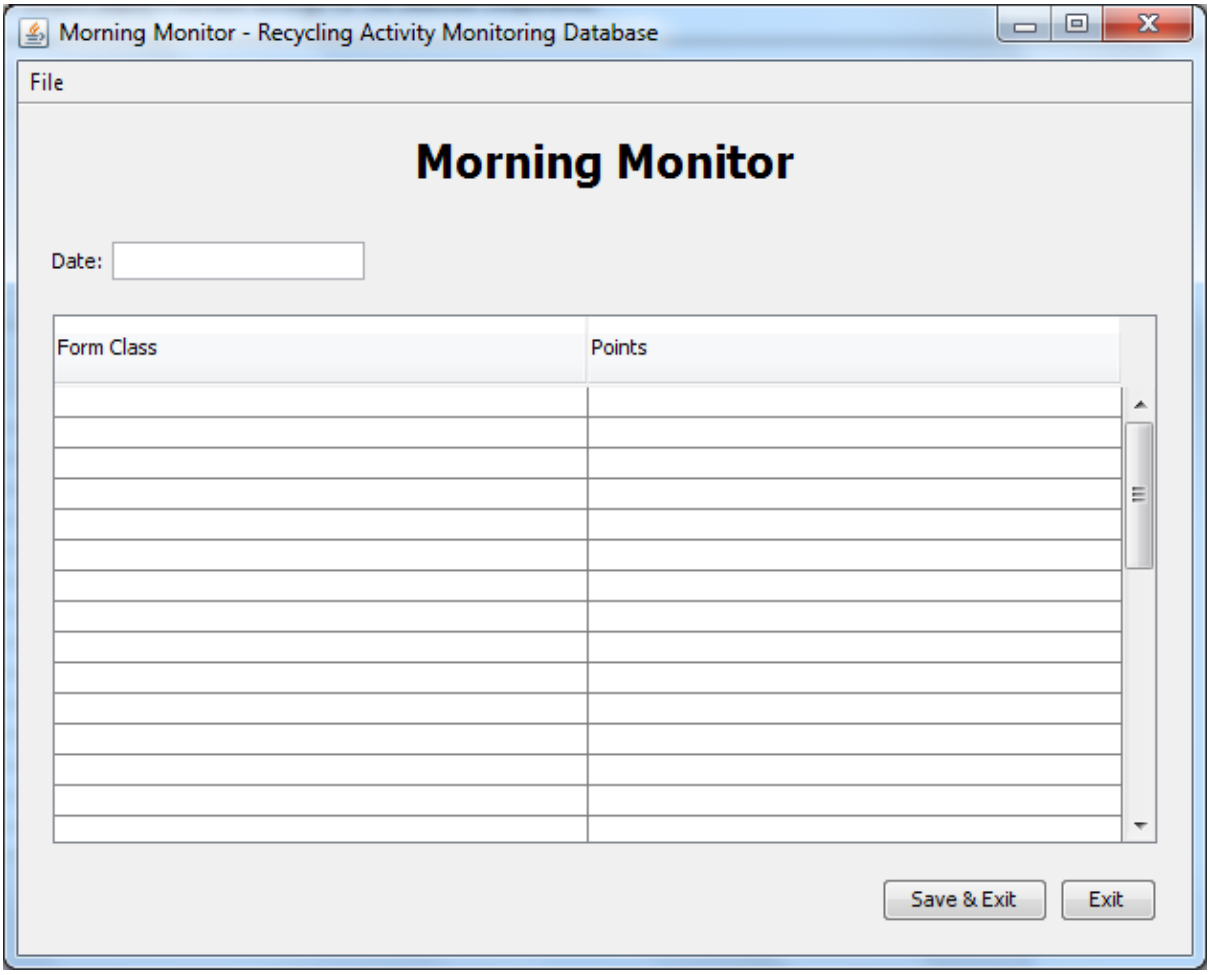
Discussion

Mrs. Cooper: This looks fine. It's pretty much self-explanatory so there's no problem understanding it. What is the difference between the 'Log out' and the 'Close' menu buttons though?

Harris: The 'Log out' button closes the current screen that the user is on and switches back to the password screen. This is useful if you need to step outside for some time and you don't want any unauthorised people to gain access to the system that has been left open. The 'Close' button just closes the entire application. Also, please note that the program will not save changes to the current screen that you are on if you don't press the 'Save' button if relevant to the screen.

Mrs. Cooper: I pretty much figured that out. Just wanted to confirm it.

Morning Skip Monitor



The 'File' menu will have two menu items; Log out & Exit.

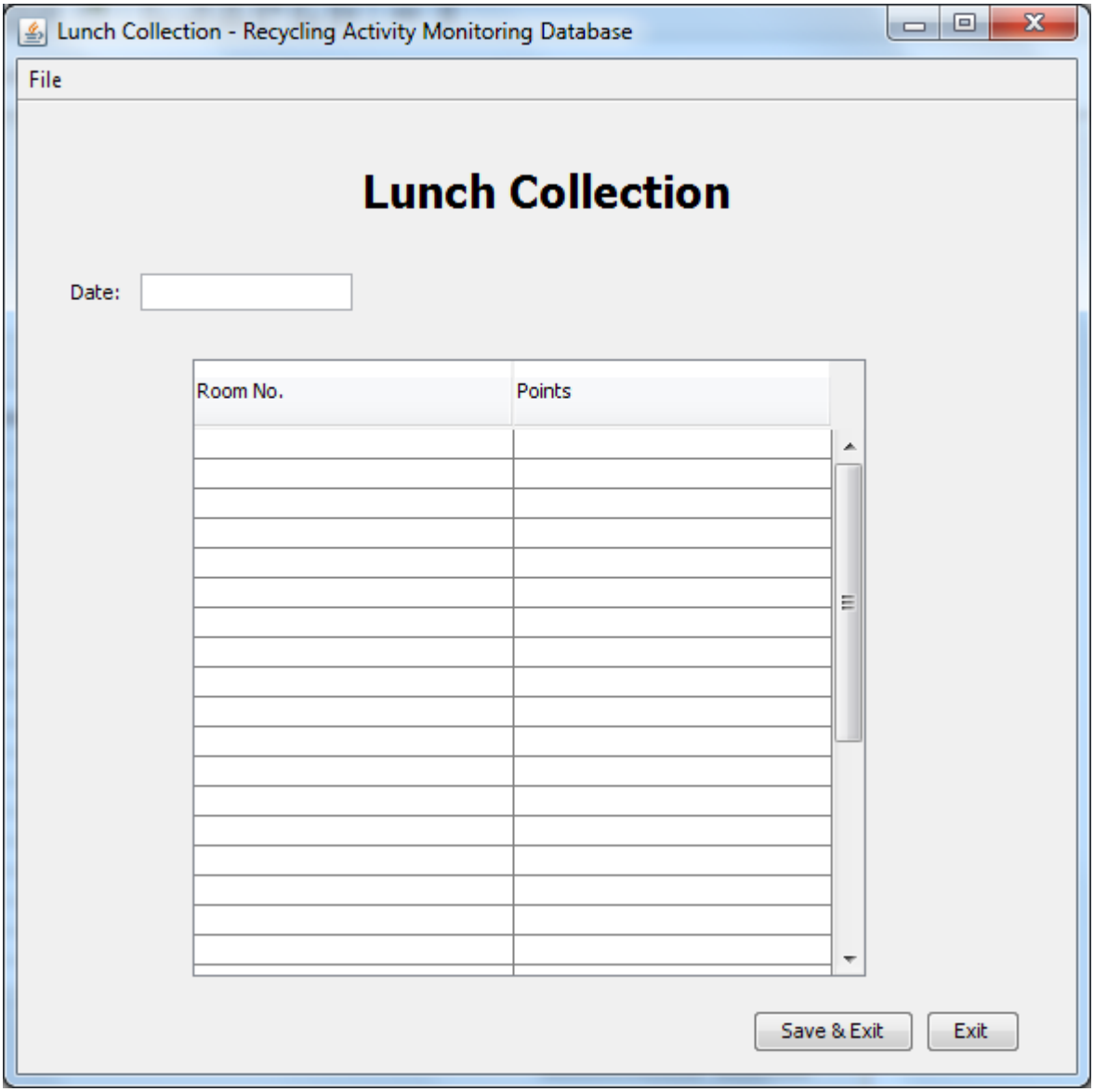
Discussion

Harris: The need to cross-reference 'Grade Form Teachers Table' manually will be unnecessary as this will be done automatically in the new system.

Mrs. Cooper: Great! That reduces the workload and time taken to input data into the system. I know you took the data home every weekend and got your brother, Omar to do it. It seems amazing that all that will be done automatically and instantly when the data is input. Also, I have a question. How many rows will there be in the table above?

Harris: I have decided to put thirty-eight rows in the table since only thirty-eight form classes can come to the recycling skip every Thursday morning. Even though at odd times Mr. Craig Blake (Dean of Students) or the staff room have their recycling sent on Thursdays, it's usually one or two people and the form classes that bring recycling is just below half so thirty-eight rows should be enough in any circumstances.

Lunch Collection Rounds



The 'File' menu will have two menu items; Log out & Exit.

Discussion

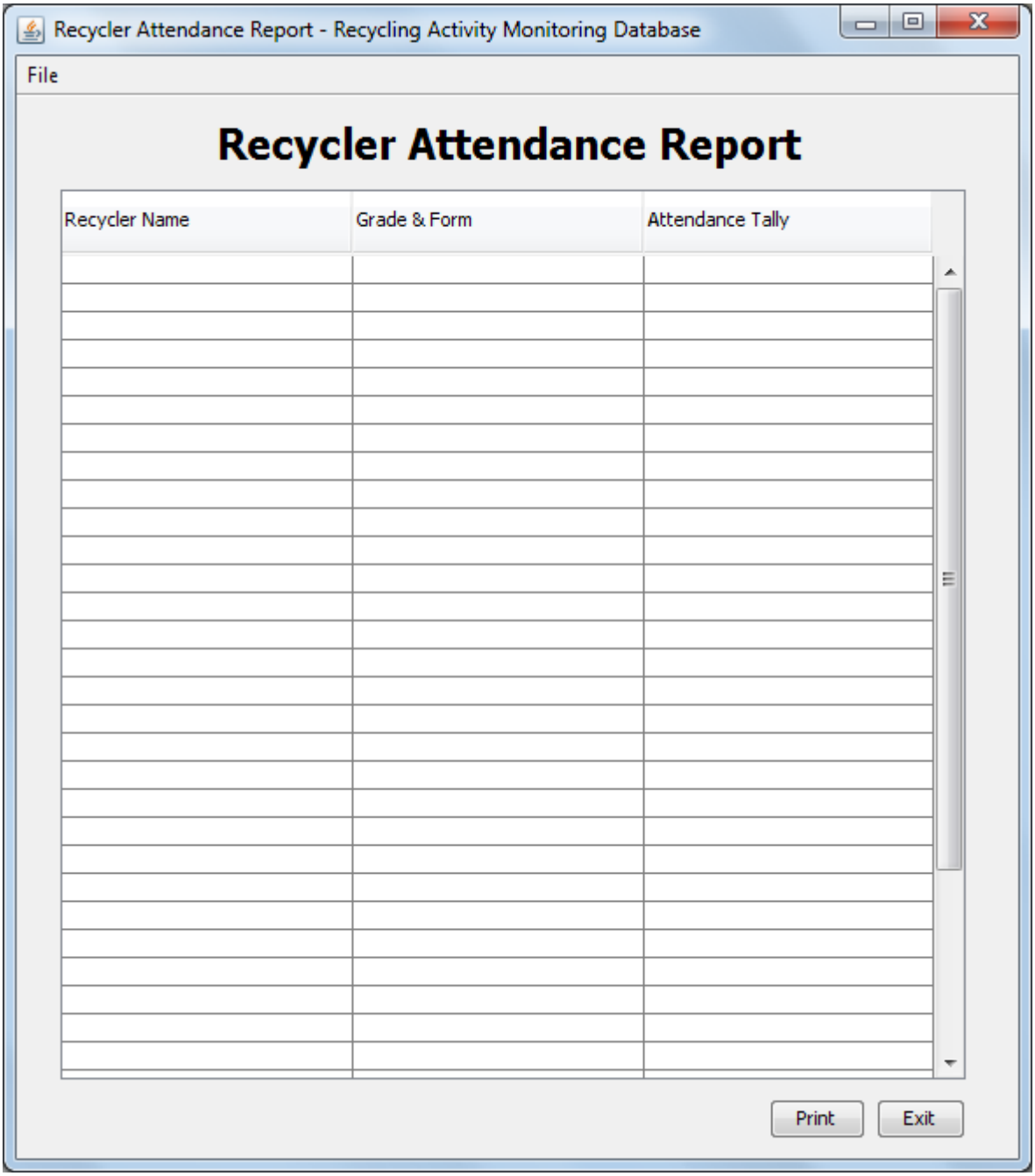
Harris: Please note that when the user presses Save & Exit, a message box will appear allowing the user to select from a list of recyclers that have been registered into the system. This process only occurs after the classrooms have been input.

Mrs. Cooper: What happens if the recycler's name is not registered?

Harris: I will have to add a button that will allow you to register a recycler at this stage. I forgot that a recycler might suddenly show up and do a recycling round without registering.

Mrs. Cooper: Ok. Please do so.

Recycler Attendance Report



The 'File' menu will have two menu items; Log out & Exit.

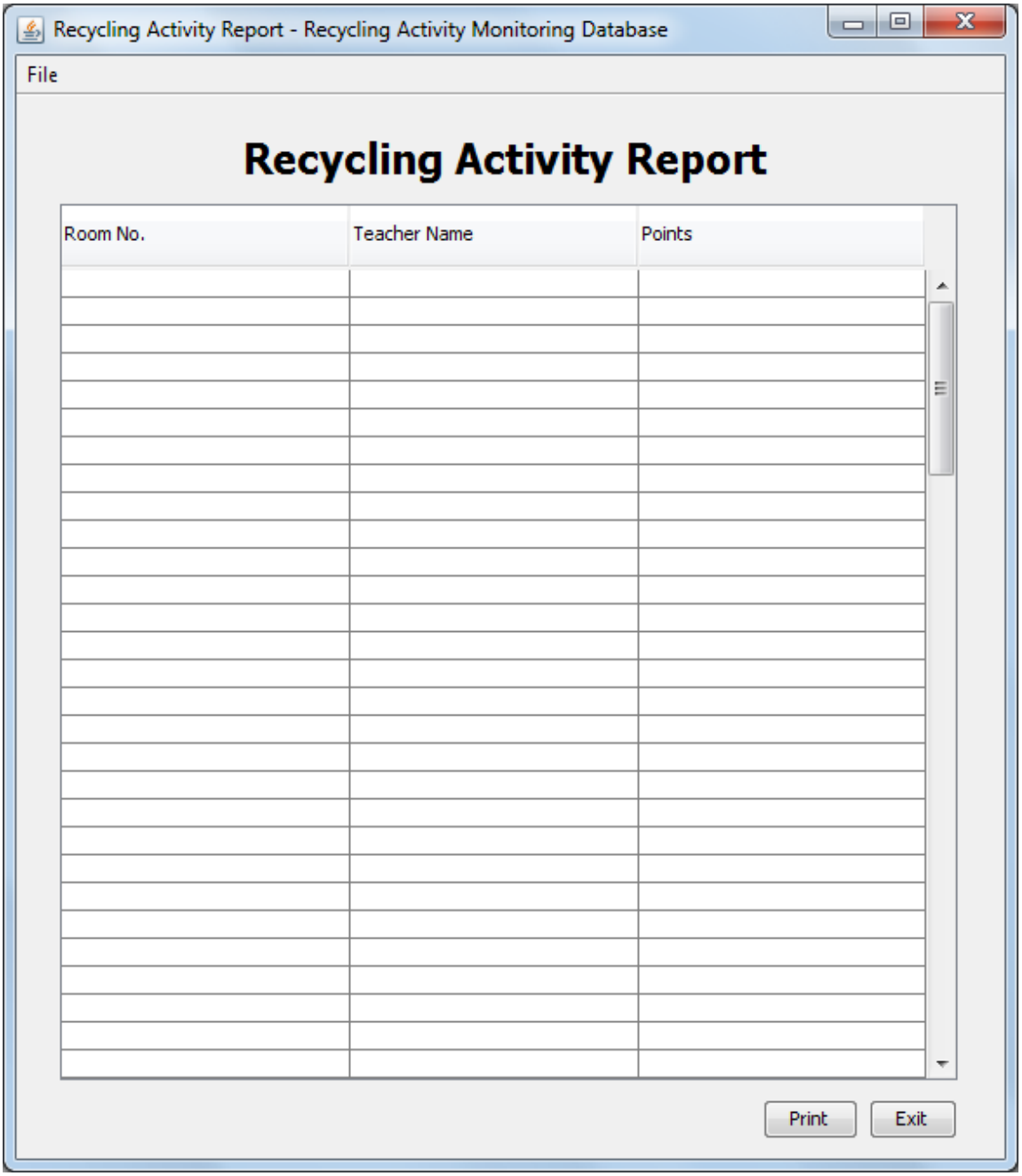
Discussion

Mrs. Cooper: It looks fine. I hope that the table will print out well.

Harris: I have never tried out the printing function but I will do my best to make sure that the report is output appropriately. FYI, this table will have one hundred and twenty-two rows for all the classrooms in the school.

Mrs. Cooper: Yes. That's good.

Recycling Activity Report



The 'File' menu will have two menu items; Log out & Exit.

Discussion

Mrs. Cooper: It looks ok. Hey... Is it possible if you can make this table sorted? According to points in descending order of course.

Harris: Yes. That can be easily done.

Mrs. Cooper: Could you also implement that with the previous report?

Harris: Yes.

Recycler Registration

The 'File' menu will have two menu items; Log out & Exit.

Discussion

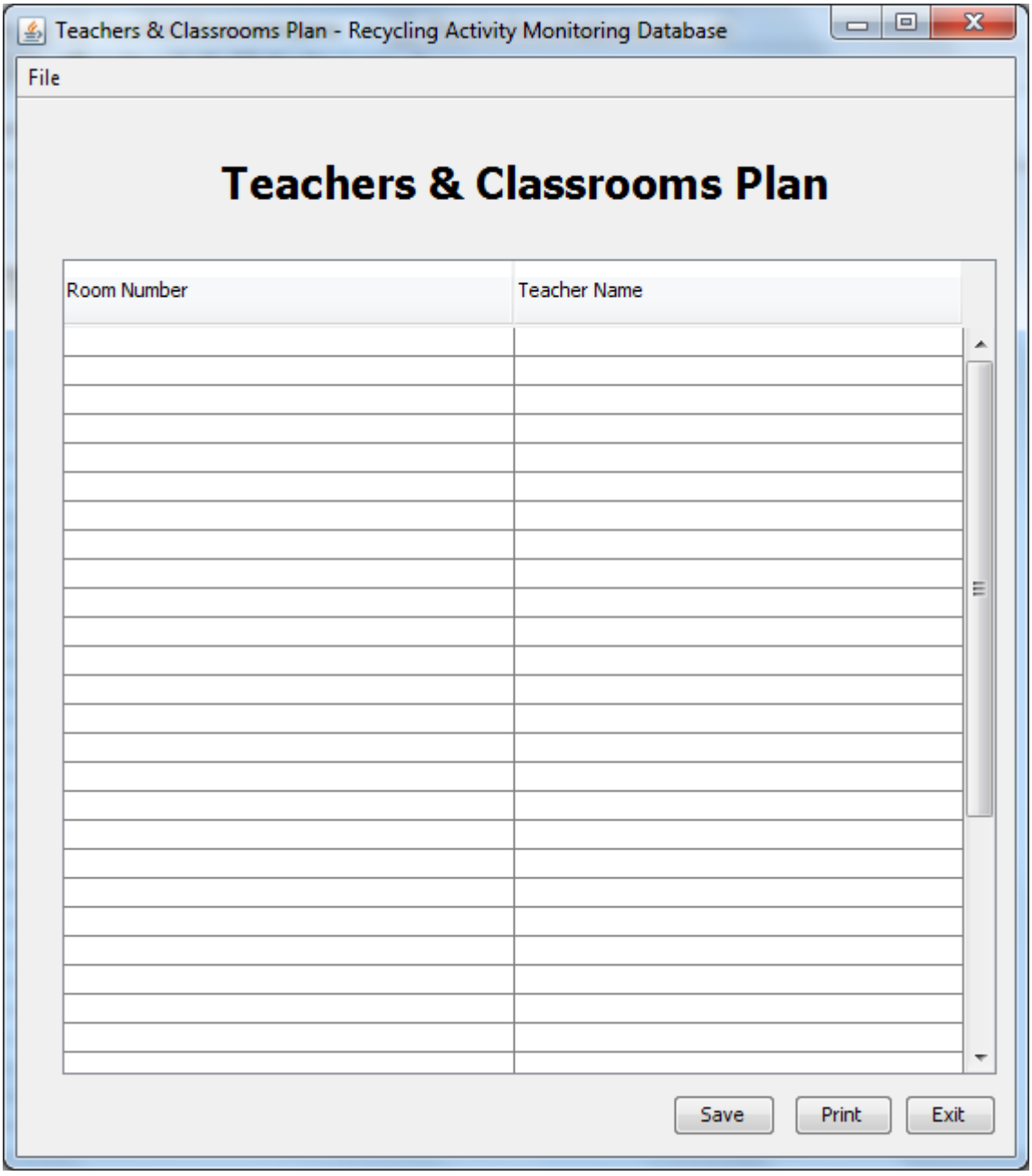
Mrs. Cooper: It looks ok but please keep the sizes of the buttons the same. It looks very weird this way.

Harris: Hahaha... Sure. I wasn't quite sure as to the layout I should have set for this screen so I played around with some styles. Do you prefer big or small buttons?

Mrs. Cooper: I think bigger buttons are easier to use. Bigger meaning taller than the buttons on this screen and not as long. I think you over-did it here.

Harris: Ok. Is that all?

Teachers & Classrooms Plan



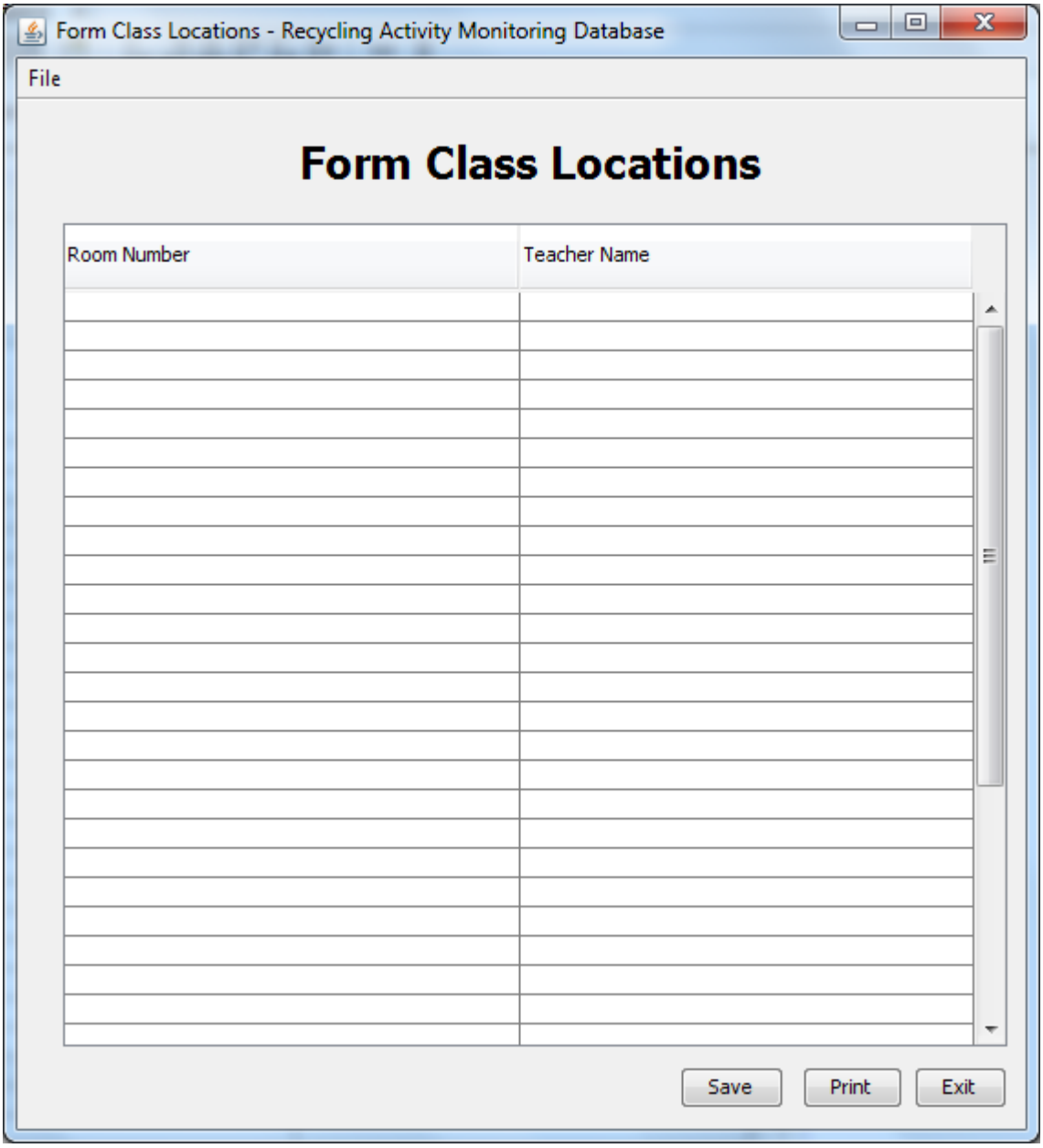
The 'File' menu will have two menu items; Log out & Exit.

Discussion

Mrs. Cooper: It looks good. This is the screen that displays the reference table, right?

Harris: Yes.

Form Class Locations



The ‘File’ menu will have two menu items; Log out & Exit.

Discussion

Harris: This is the other reference table which is more important because of the Morning Skip Monitor screen. It is essential in the cross-referencing process so the data should be up-to-date.

User Feedback Summary

Now that the initial prototypes were completed, I went on to create the prototypes for the additional screens that had to be added to the program.

After taking a look at the prototypes on the previous pages, Mrs. Cooper gave major feedback to each one. The following are the new screens being added to the system.

- Answer Secret Question
- Security Settings
- Assign Secret Question
- Change Password

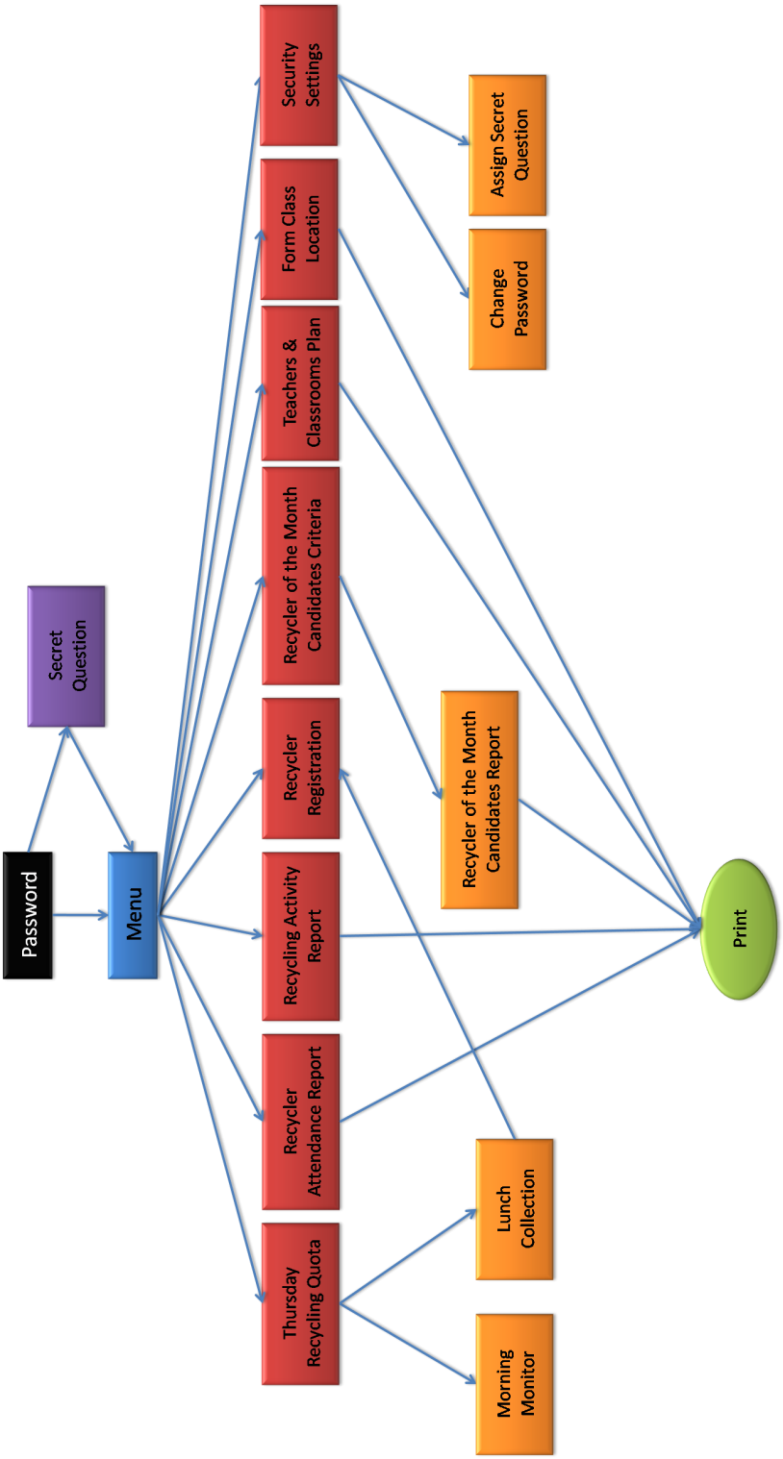
The following screens were also forgotten in the design process so their prototypes will have to be designed for evaluation before the program is created.

- Recycler of the Month Candidate Criterion
- Recycler of the Month Report

In addition, the other changes that were suggested include adding:

- more colour to some of the screens so that it looks more attractive. I thought of setting a background colour to the screens. Since green is the most user-friendly colour and the least strain on the eyes as well as being the Eco Club's theme, I will make use of it in the program.
- a button to allow the user to access the Recycler Registration screen during the input stage of Lunch Collection in Thursday Recycling Quota.
- maintain button sizes so that it looks neat.

On the following page is a re-designed diagram of the program's structure.


Password
Secret Question
Menu
Security Settings
Form Class Location
Teachers & Classrooms Plan
Recycler of the Month Candidates Criteria
Recycler Registration
Recycling Activity Report
Recycler Attendance Report
Thursday Recycling Quota
Assign Secret Question
Change Password
Recycler of the Month Candidates Report
Lunch Collection
Morning Monitor
Print

Secret Question

The 'File' menu will have two menu items; Log out & Exit.

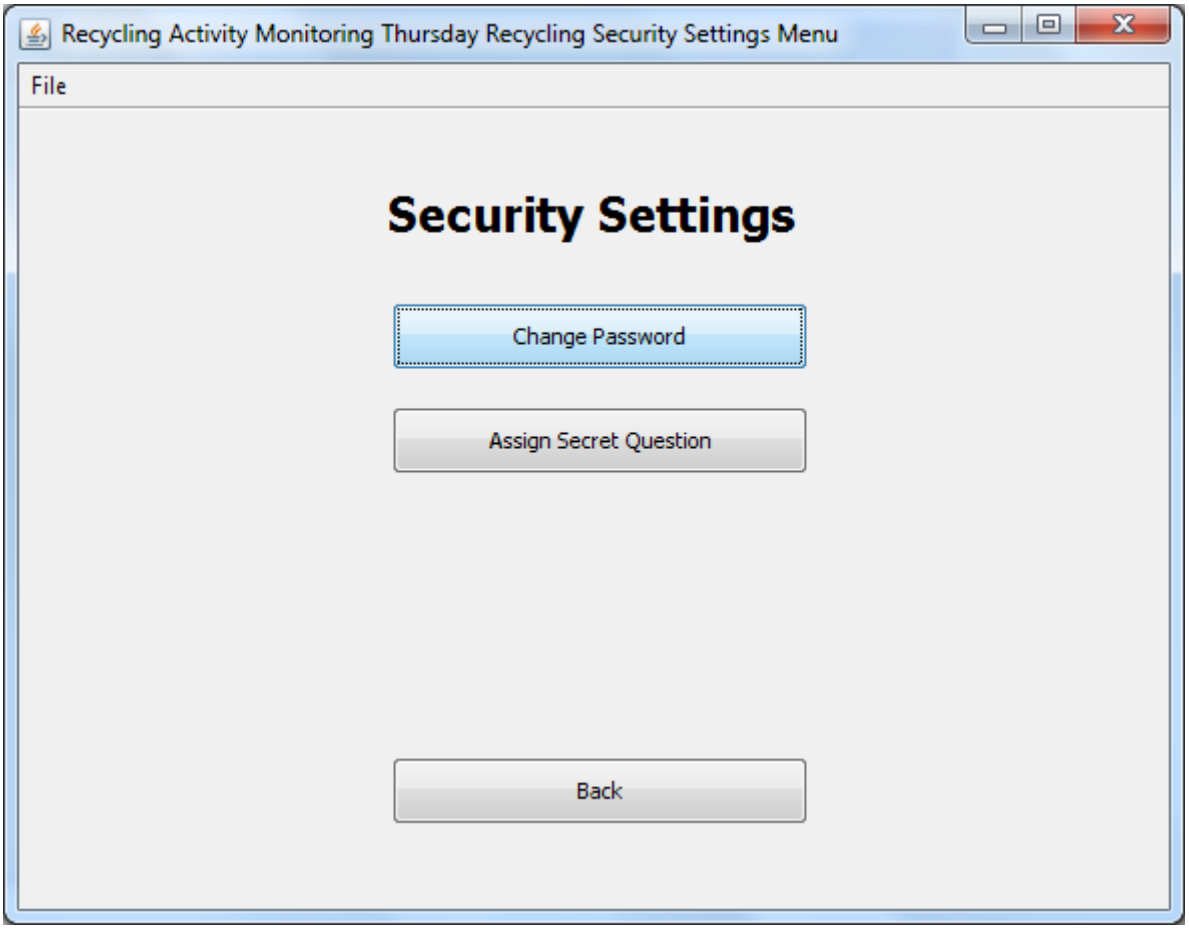
Discussion

Mrs. Cooper: Great! I will be able to change the question and the answer right?

Harris: Yes. I was thinking of just making this a pop-up box.

Mrs. Cooper: That would be fine too since the screen has so little information on it.

Security Settings



The ‘File’ menu will have two menu items; Log out & Exit.

Discussion

Mrs. Cooper: Looks good.

Change Password

The 'File' menu will have two menu items; Log out & Exit.

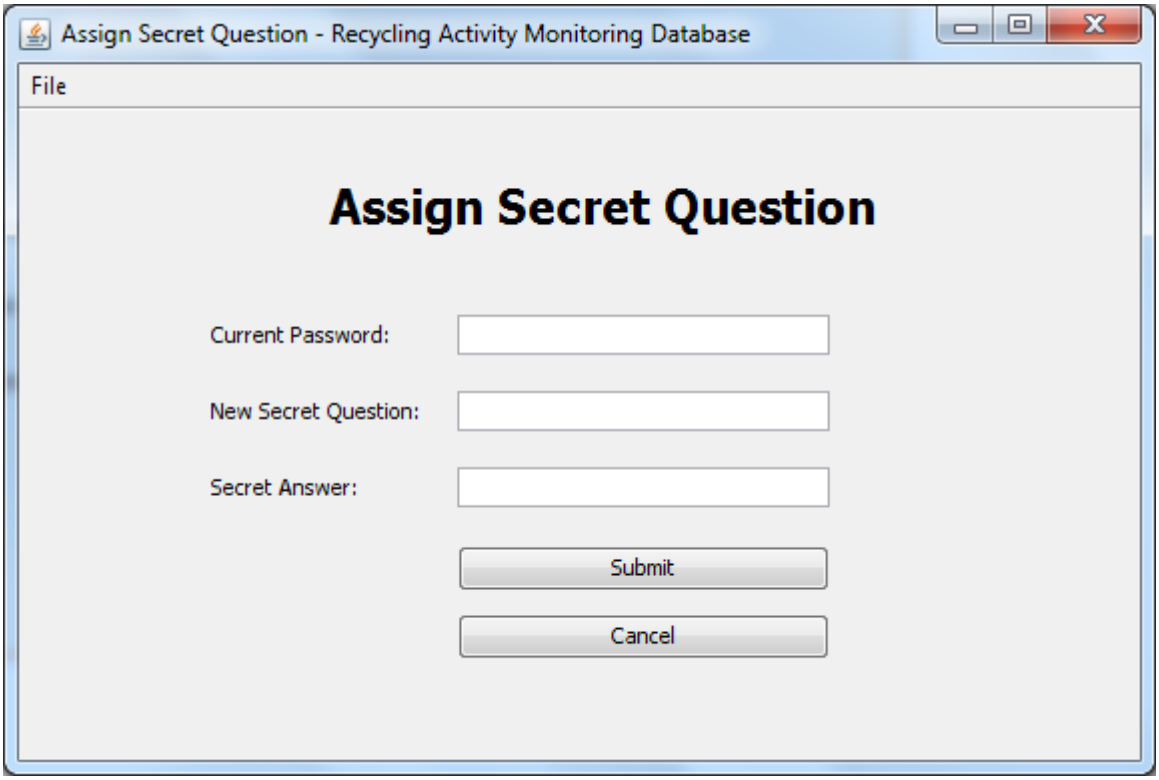
Discussion

Mrs. Cooper: It looks ok. Will someone be able to read what I am typing on this screen?

Harris: No. The text fields above are special because they hide the characters you write and display a symbol instead. It will appear like a password field you use to log on to your e-mail or like the one that is used on the password screen.

Mrs. Cooper: Great!

Assign Secret Question

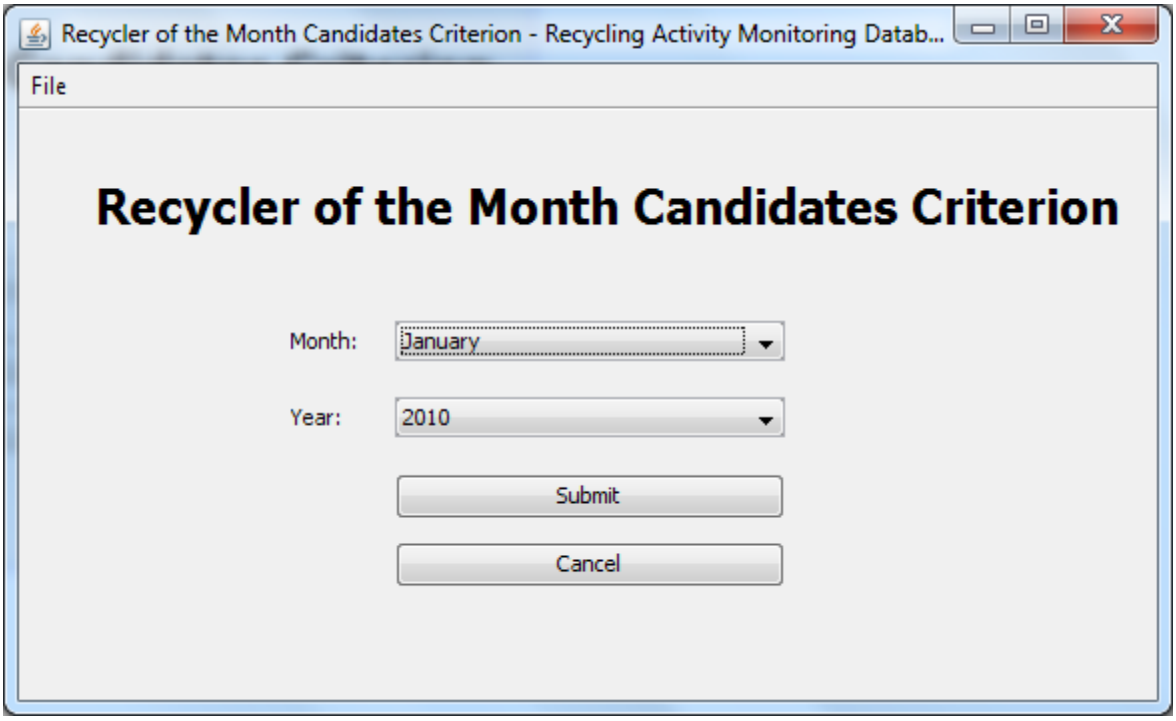


The 'File' menu will have two menu items; Log out & Exit.

Discussion

Harris: The same thing applies to this screen. The password cannot be read when you are typing it. However, the secret question and the secret answer will appear as text that can be read.

Recycler of the Month Candidates Criteria



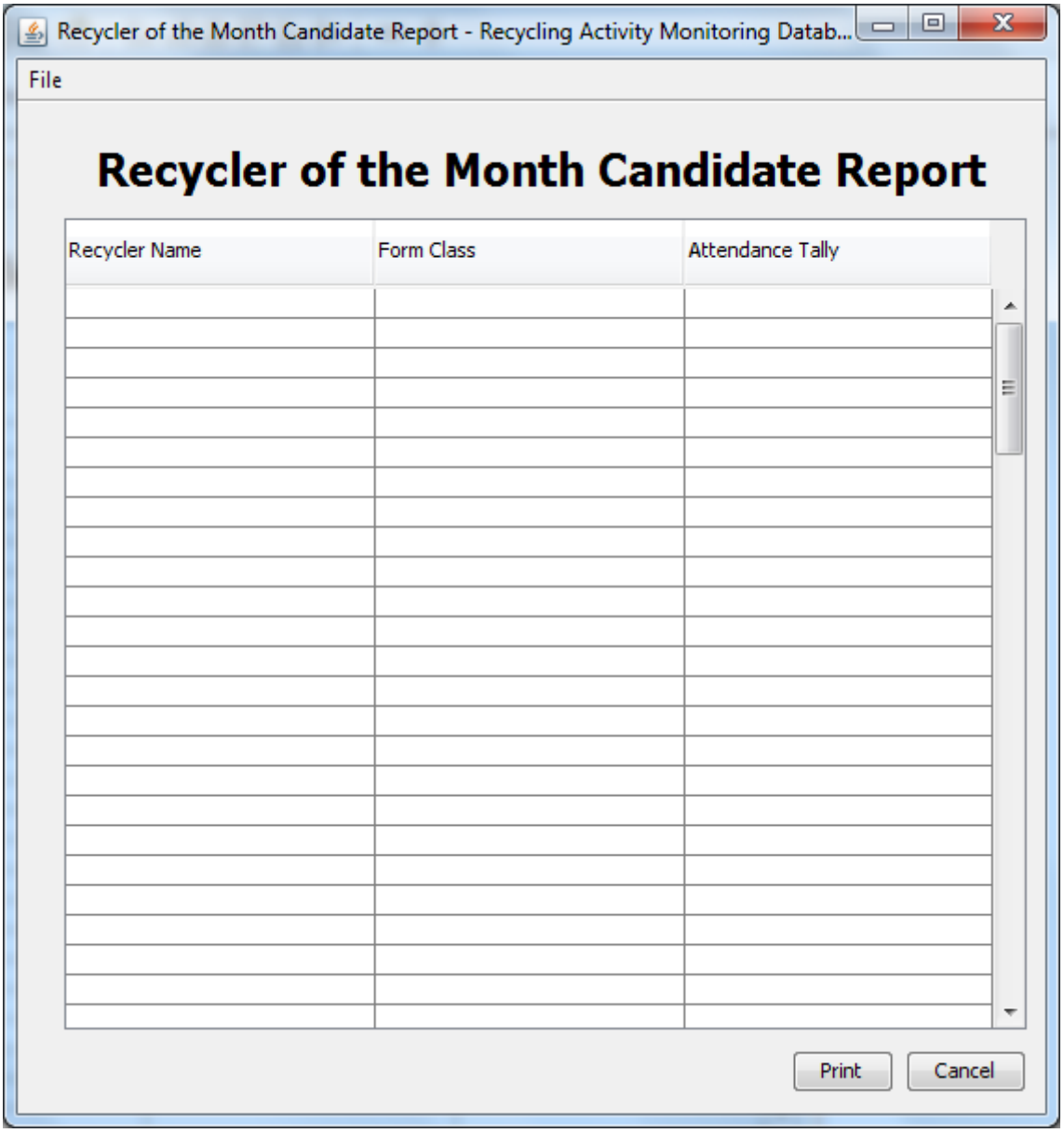
The 'File' menu will have two menu items; Log out & Exit.

Discussion

Mrs. Cooper: It looks fantastic! I can't wait to use this new system. So all I have to do is specify the month and year of the Recycler of the Month award I am giving and it'll give me a list of people with their attendance stats?

Harris: Yes!

Recycler of the Month Candidates Report



The 'File' menu will have two menu items; Log out & Exit.

Discussion

Mrs. Cooper: Perfect.

Harris: The candidate list in actual fact will probably be much shorter than it looks on this prototype.

Criterion B1: Data Structures

How to Organise Data

With my analysis and prototypes of the system completed, I thought about the data structures that I would use in the system. Lots of data is set to be held in the new Recycling Activity Monitoring system.

2D arrays will be mainly used to store data during processing because the vast majority of data is to be displayed in the form of tables. A random access file will also be used to store records of data and other information that needs to be retained when the program is exited.

The use of an abstract data structure would be redundant in the Recycling Activity Monitoring System I am developing. I could simply make use of 2D arrays throughout the program as only small amounts of data will be held in memory and this data structure is ideal for the manipulation and processing of JTable which is to be implemented throughout the program. The following table demonstrates the structure of a 2D array to be used in my program.

	Recycler Name	**Form Class**	**Attendance Tally**
Index Number	0	1	2
0	Omar Rasheed	8E(CB)	2
1	Harris Rasheed	13A(MM)	4
2	Norlisa Rosslee	12C(KB)	5

As more data is input into the system, the new information will be simply added to the attendance tally column. However if a new recycler is registered, a new row is created for him/her. This is likely to go on and the data structure will slowly increase over the long-term. At the beginning of every academic year, the size of this structure will dramatically increase when numerous recyclers register. This is one of the many 2D arrays I will implement in my program. The 2D arrays that store statistics will follow the above behaviour.

The other type of 2D array to be used in my program will be the transaction files of my program. An example is shown below.

	Recycler Name	**Date**
Index Number	0	1
0	Omar Rasheed	12/03/2010
1	Harris Rasheed	12/12/2009
2	Norlisa Rosslee	14/04/2009
3	Harris Rasheed	14/04/2009

As more data is input, these 2D arrays will grow large in size as every statistic occupies one row. Every week, a maximum of 160 records will be input as the worst case scenario. With a record size of 40 bytes, this means that the system could take up to 6,400 bytes per week which can build up over time and occupy lots of storage space. The table on the previous page demonstrates the duplication that has occurred with the records (Recycler Harris Rasheed) but this is necessary to keep track of dates. This array will be used in the Recycler of the Month Candidates Report.

A random access file will be made use of to store data in the computer's non-volatile memory so that it will be available when the program is opened again, after the computer is even shut down. There will be multiple instances and usages of random access files particularly when the processes are occurring with JTable. Since there are so many different types of records required in the system, six random access files will be used. The data processed with 2D arrays comes from random access files.

The following table demonstrates the various data types that will be held in the Random Access File named *FormClassroomLocation*.

Field Name	Form Class	Room Number
Random Access File Data Type	String	String
Java Program Data Type	String	String
Record Size (bytes)	8	5
Sample	12A (MM)	HS41A

This random access file is a reference table that will be used to match the form classes input to the room numbers so that statistics can be processed. The size of each record in this Random Access File is 13 bytes. Since the number of records in this random access file are of a set size because there are always 38 form classes, the total size of the file will be 494 bytes (13 bytes × 38 records).

The next table shows the fields to be used in the *RecyclingActivityStats* random access file.

Field Name	Room Number	Points
Random Access File Data Type	String	String
Java Program Data Type	String	Integer
Record Size (bytes)	5	3
Sample	M3	2

This random access file will store data for the Recycling Activity Report. It is one of the main objectives that the system has to fulfil. The data here would be read and processed with statistics that are input. The size of each record in this Random Access File is 8 bytes. The number of records

in this random access file would also be of a set size because there are only 122 classrooms in the school. Therefore, the total size of the file will be 976 bytes (8 bytes × 122 records).

The subsequent table shows the fields to be used in the *RecyclingActivityLog* random access file.

Field Name	Room Number	Points	Date
Random Access File Data Type	String	String	String
Java Program Data Type	String	Integer	String
Record Size (bytes)	5	2	10
Sample	AX16	1	27/04/2009

This random access file will keep a log of recycling statistics that are processed in the database. This data is simply kept as a backup. The size of each record in this Random Access File is 17 bytes. As more data is input, the size of this file will grow exponentially because it logs all recycling statistics transactions. This can reach a maximum of 160 records per week (38 form classes + 122 classrooms).

The following table shows the fields to be used in the *RecyclerAttendanceStats* random access file.

Field Name	Room Number	Points
Random Access File Data Type	String	String
Java Program Data Type	String	Integer
Record Size (bytes)	5	3
Sample	SC13	47

This random access file will store the recycler attendance statistics as a tally. This file is the key objective. The size of each record in this Random Access File is 8 bytes. A new record would be added to this file every time a new recycler is registered.

The next table shows the fields to be used in the *RecyclerAttendanceLog* random access file.

Field Name	Recycler Name	Date
Random Access File Data Type	String	String
Java Program Data Type	String	String
Record Size (bytes)	30	10
Sample	Jaime Tan	16/03/2010

This random access file will store data for the Recycler of the Month Candidate Report. The number of times a recycler's name is mentioned in this file will determine their placement on the Recycler of the Month Candidates Report. This data would come from the Lunch Collection input screen. The size of each record in this Random Access File is 40 bytes. When data is input, the size of this file will grow exponentially because it logs all recycler attendance transactions.

The subsequent table shows the fields to be used in the *TeacherClassroomPlan* random access file.

Field Name	Room Number	Teacher Name
Random Access File Data Type	String	String
Java Program Data Type	String	String
Record Size (bytes)	5	30
Sample	AX16	Mr. Stephen Munnery

This random access file is another reference table that will be used to match the room numbers in the Recycling Activity Report with the name of the teachers so that statistics can be well presented. The size of each record in this Random Access File is 35 bytes. Since the number of records in this random access file are of a set size because there are only 122 classrooms, the total size of the file will be 4,270 bytes (35 bytes × 122 records).

The following table shows the fields to be used in the *SystemSecurity* random access file.

Field Name	Password	Secret Question	Secret Answer
Random Access File Data Type	String	String	String
Java Program Data Type	String	String	String
Record Size (bytes)	20	60	20
Sample	mcooper	What is your favourite hobby?	Chess

This random access file stores the system security details that is essential for user authorisation. It stores the password, secret question and its answer. The total size of the random access file is 100 bytes.

As can be seen above throughout the data structures, only two data types will be required through the course of the program; String to store data and Integer to store numbers that are used for calculations. The String data type will be dominant throughout the program because this data is frequently processed with/in random access files and allows easier processing. Numerical data from random access files will be parsed before they are used for processes.

Criterion B2: Algorithms

Program Procedures

After having outlined the program data structures, I considered and started to design the algorithms to be used in my program.

The algorithms that will be used in my program will be:

- currentPassword
- currentSQtionAnswer
- storeRecyStats
- logRecyStats
- processRecyclerAttendance
- referenceFormLocation
- loadTable
- candidateSelection
- saveData
- registerRecycler
- deleteRecycler
- bubbleSort

Algorithm Description: *currentPassword*

Parameters Passed IN: String password

Return Type: boolean

Variables Used: int passCounter, String attemptPassword, RandomAccessFile RAF

Algorithm

1. User inputs password
2. Password is read from RAF
3. Check if the password is the same as the one in the RAF
    a. If equal, return true
    b. Otherwise, add one to passwordCounter
        i. Check if passCounter is equal to five
            1. If true, exit program
        ii. Output error message
        iii. Return false

Output Description: This algorithm will be used to verify if the password input on the login screen of the program is correct or not. It also deals with repeated failed attempts to gain access.

Algorithm Description: *currentSQtionAnswer*

Parameters Passed IN: None

Return Type: String[]

Variables Used: RandomAccessFile RAF, File SystemSecurity, String[] sQtion

Algorithm

1. Read RAF to get secret question
2. Remove additional spacing from question and assign it to 2D array
3. Read RAF to get secret question's answer
4. Remove additional spacing from question and assign it to 2D array
5. Return the secret question and answer in an array

Output Description: This algorithm will be used to read the secret question and the secret question's answer from the random access file and it returns this data.

Algorithm Description: *storeRecyStats*

Parameters Passed IN: String [][] tableData, int [] currentPoints

Return Type: None

Variables Used: String[][] array2, RandomAccessFile RAF, int recordSize, int records

Algorithm

1. Reads each record from random access file
    a. If data in the array and random access file matches, the points are processed.
    b. If not matched, output error

Output Description: This algorithm will be used to update the recycling activity points of classrooms.

Algorithm Description: *logRecyStats*

Parameters Passed IN: String[][] tableData, int rows, String date

Return Type: None

Variables Used: File file, RandomAccessFile RAF, int records, int recordSize, String roomNo, String points

Algorithm

1. Loops until all records have been written to the random access file
    a. Goes to the end of the random access file
    b. Takes recycler name and makes variable to the size of the field length
    c. Writes recycler name and date of attendance and adds the record to the file

Output Description: This algorithm will be used to keep a log of data that is input in the system.

Algorithm Description: *processRecyclerAttendance*

Parameters Passed IN: String[][] recyNames, int rows

Return Type: None

Variables Used: File file, RandomAccessFile RAF, int records, int recordSize, String currentLine

Algorithm

1. Loops until all recyclers have been processed
2. Reads every record of random access file
    a. If the record and recycler match, the attendance tally is increased by one

Output Description: This algorithm will be used to keep a tally of recycler attendance for the Recycler Attendance Report.

Algorithm Description: *referenceFormLocation*

Parameters Passed IN: String[][]tableData, int rows

Return Type: String[][]

Variables Used: int records, recordSize, String currentLine

Algorithm

1. Loop to read all records from random access file
    a. Loop to compare each record to JTable values in 2D array
        i. If current record's field and JTable record match, overwrite form class with room number.

Output Description: This algorithm will be used to match form classes to the their location/room number.

Algorithm Description: *loadTable*

Parameters Passed IN: File file, int recordSize, int firstField, int secondField

Return Type: String[][]

Variables Used: int records, String[][] tableData, String currentLine, String first, String second, String third

Algorithm

3. Loops to read all records in random access file
    a. Take a substring of each field, trim it and assign it to the 2D array row
    b. Repeat loop until all records are read

Output Description: This algorithm will be used to load JTable when a report is opened.

Algorithm Description: *candidateSelection*

Parameters Passed IN: String[][] logData, int records, String month, String year

Return Type: String [][]

Variables Used: int recordSize, int rows, rowCounter, String[][] currentMonthAttendance, boolean double

Algorithm

1. Loops through records in 2D array
    a. If current record month is equal to criterion month
        i. If record already exists, add one to currentMonthAttendance 2D array
        ii. Otherwise add record to currentMonthAttendance 2D array

Output Description: This algorithm filters and selects all recyclers who have attended recycling rounds for the month specified by the user.

Algorithm Description: *saveData*

Parameters Passed IN: String[][] tableData, int firstFieldSize, int secondFieldSize, File file

Return Type: None

Variables Used: int records, int recordSize, RandomAccessFile RAF

Algorithm

1. Loop to access beginning of each record in RAF
2. Overwrite record with each row from 2D array tableData

Output Description: This algorithm will be used to update the Teachers & Classrooms Plan and the Form Classrooms Locations with updated data from the user.

Algorithm Description: *registerRecycler*

Parameters Passed IN: String recylerName, String formClass

Return Type: None

Variables Used: int records, int recordSize, RandomAccessFile RAF, String currentLine

Algorithm

1. Loop to access beginning of each record in RAF
    a. If a record matches the name attempted to be registered
        i. If a record matches the form class attempted to be registered, Output Error
2. Add name and form class as a record to the end of the file

Output Description: This algorithm will be used register recyclers into the system so that their attendance can be processed.

Algorithm Description: *deleteRecycler*

Parameters Passed IN: String recylerName

Return Type: None

Variables Used: int records, int recordSize, RandomAccessFile RAF, File file

Algorithm

1. Loop to read each record in RAF
    a. If the name in the first field matches the recycler to be deleted, delete record

Output Description: This algorithm will be used to delete recyclers from the system when required.

Algorithm Description: *bubbleSort*

Parameters Passed IN: String[][] tableData, int rows

Return Type: String[][]

Variables Used: int upper, int lower, int top, int bottom, String[][] tableData

Algorithm

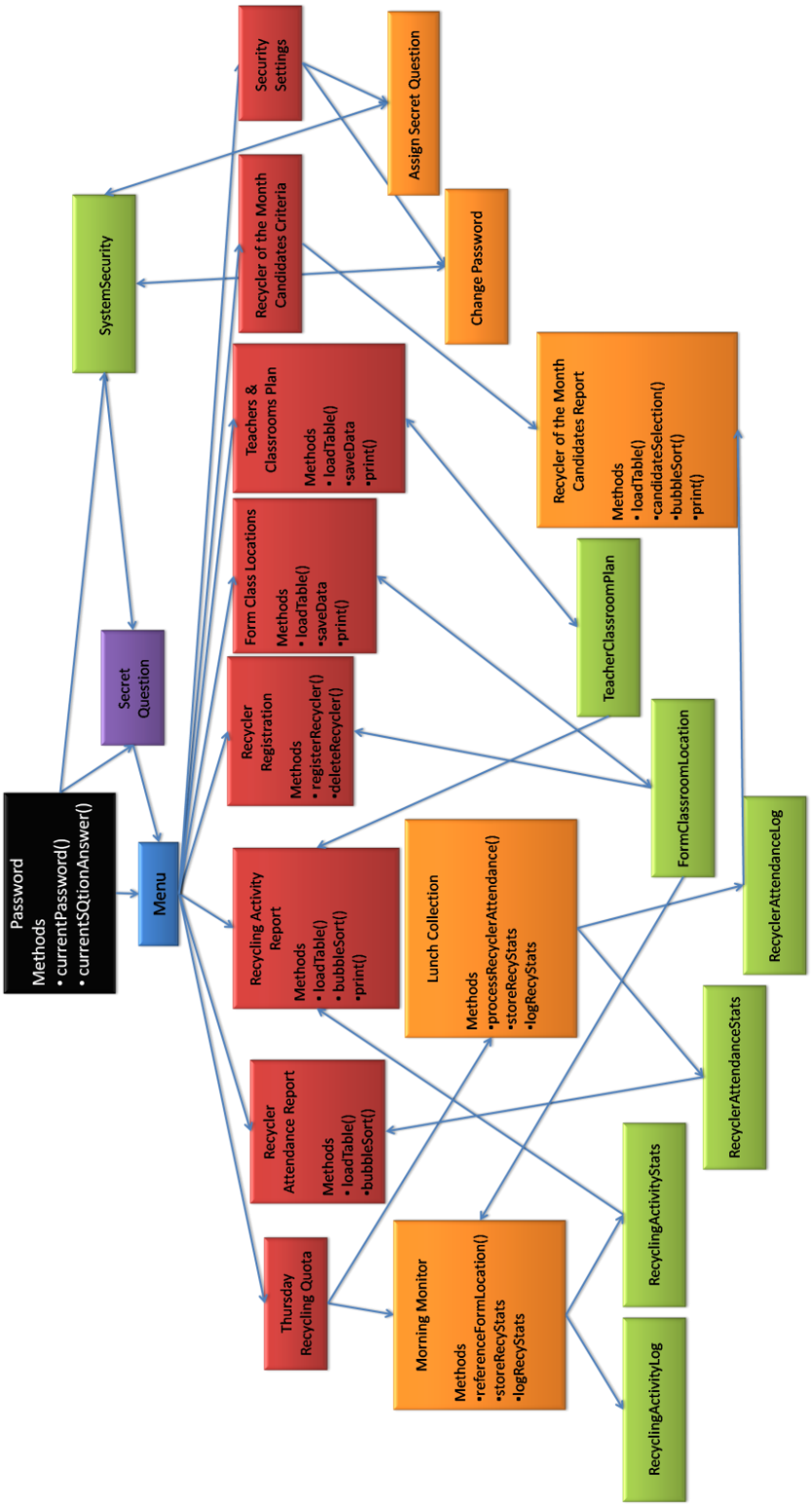
1. This algorithm has a standard procedure.

Output Description: This algorithm will sort the 2D array for reports by points/attendance tally.

Criterion B3: Modular Organisation

Program Modules

With the completion of the program's algorithm design, I set out to organise my program into different modules. The green boxes represent the random access files whereas the other colours simply denote the number of screens necessary to go through in order to get there from the password screen.



Criterion C1: Using Good Programming Style

Coding the Program

As a result of data structure, algorithm and modular design, I programmed the code required for the program to function properly. The following pages outline the program's code listing.

Criterion C2: Handling Errors

Errors & Extreme Data Prevention

There are numerous instances of error handling throughout the program. This has been done to prevent unexpected errors, extreme and invalid data from occurring or being input and the system crashing. In addition, user-friendly message boxes have been designed to inform the user of the error that occurred.

The major 'Exception' errors were dealt with using try catch blocks. For the catches, catches were placed in order for the message box communicating the error to be specific in the glitch that occurred.

Some Exceptions that were caught include FileNotFoundException, which occurs when a random access file that the program is attempting to access for reading or writing does not exist in the current directory. The rest of the errors were caught under a general exception and displayed in a message box as an error code. Some of the errors that fall under here include the IOException. This error is caught when errors occur with the input and out stream of the program. EOFException is when a file pointer attempts to read the contents of a file but reaches the end of the file before completing the process. This error is only likely to occur during the programming process or if the random access file data is tampered with and the size of any record(s) is/are changed.

The negligence of using try catch can result in runtime errors occurring without the knowledge of the user. Examples of a try catches used in my program are as follows:

```
catch(FileNotFoundException e)
{
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this, "The FormClassroomLocation.txt notepad file is
missing from the current directory. This process cannot function without this file.\nError Code: " +
e,"File is Missing!", JOptionPane.ERROR_MESSAGE);
}

catch(Exception e)
{
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this, "An error has occured. Please contact Harris
Rasheed to deal with this issue\nError Code: " + e,"Error Message",
JOptionPane.ERROR_MESSAGE);
}

catch(java.awt.print.PrinterException e)                    //Catches printer exception
{
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(this, "Unable to print due to " + e, "Print Job Error",
JOptionPane.ERROR_MESSAGE);
}
```

If any of the above exceptions occur, the appropriate message is displayed and the user prompted to take a course of action.

There are also many other errors that could occur as a result of user input. In order to prevent problems from occurring with this, I wrote several pieces of code for input data validation.

The following code shows the verification placed to prompt the user that the user's print job that has been cancelled because the Cancel button was pressed either intentionally or deliberately.

```
if (!table.print())
{
	JOptionPane.showMessageDialog(this, "The user has cancelled the print job.", "Cancelled Print Job", JOptionPane.INFORMATION_MESSAGE);
}
```

Another form of validation used is for the Lunch Collection Rounds screen. When the user is prompted to enter the number of recyclers who participated, the following code will prevent the system from crashing when a string is input.

```
int totalRecyclerNames;
try
{
	totalRecyclerNames = Integer.parseInt((String)JOptionPane.showInputDialog(null, "How many recyclers participated this week?","Number of Recyclers", 3));
}
catch(Exception e)
{
	JOptionPane.showMessageDialog(this, "Error! Please input a number.","Error", JOptionPane.ERROR_MESSAGE);
	return;
}
```

This is very important because without this code, the exception will not be caught and the problem will pass unnoticed. The processes after this code will not execute and the user will be provided with incorrect data with no known reason for it cause.

The code below serves as validation for the Recycler Registration screen. If the user attempts to register a recycler that already exists, the if structure below will hold true and an error message will be displayed to inform the user of the error.

```
if(line.substring(0,30).trim().equals(rName))
{
	if(line.substring(30,38).trim().equals(formClass))
	{
		JOptionPane.showMessageDialog(this, "This recycler already exists.", "Error", JOptionPane.ERROR_MESSAGE);
		return;
	}
}
```

The code below prevents the user from assigning a blank password because it removes the system security in place. On the other hand, it also reminds the user that a mandatory field has been left blank.

```
if((newPass.equals(""))||(newPassConf.equals(""))||(currentPass.equals("")))
		{
			JOptionPane.showMessageDialog(this, "Error. You have left a mandatory field blank.","Error", JOptionPane.ERROR_MESSAGE);
			return;
		}
```

Another code connected to this part of the program is as follows.

```
if(pass.equals(currentPass))
{
	if(newPass.equals(newPassConf))
	{
		RAF.seek(0);
		for(int i = newPass.length(); i < 20; i++)
		{
			newPass += " ";
		}
		RAF.writeBytes(newPass);
		JOptionPane.showMessageDialog(this, "The password was successfully changed.","Change Successful!", JOptionPane.PLAIN_MESSAGE);
	}
	else
	{
		JOptionPane.showMessageDialog(this, "The new password that you entered in both fields do not match. Please try again.","Error", JOptionPane.ERROR_MESSAGE);
	}
}
else
{
	JOptionPane.showMessageDialog(this, "The current password that you entered is incorrect.\nPlease note that this field is case-sensitive.","Error", JOptionPane.ERROR_MESSAGE);
}
```

This validates the data input. If the new password input does not match the one that is asked for in the confirmation box, an error is output to inform the user of the problem. This is a form of double entry verification. Without this, the user could change the password with a typo and this could pose to be a big problem because they will not be able to login again. The code also checks if the user has entered the correct authorisation password because if the database is left open, anyone could easily change the password.

The code below prevents the user from inputting a blank answer for the secret question prompt at the time of login.

```
if(gAnswer.equals(null)||gAnswer.equals(""))
{
	JOptionPane.showMessageDialog(this, "Error. Please input an answer!","Error Message",
JOptionPane.ERROR_MESSAGE);
	return;
}
```

The next code segment checks if the first column of JTable is blank when the Save button is pressed. Without this presence check validation, the system will attempt to process blank records and malfunction.

```
int count = 0;
for(int i = 0; i < rows; i++)
{
	if(tableData[i][0].equals(""))
	{
		count++;
	}
}
if(count == rows)
{
	JOptionPane.showMessageDialog(this, "The table is blank. Please input values to be
processed.","Error Message", JOptionPane.ERROR_MESSAGE);
	return;
}
```

Criterion C3: Success of the Program

Excellence of Execution

The following table details the objectives stated in Criterion A2 and evidence for having met each and every one of them in the design of the program.

Objective	Evidence
produce a *tally report* of recycler attendance;	The Recycler Attendance Report addresses this issue. This is very useful for Mrs. Cooper because it can also be printed. This will help a great deal in selecting the Recycler of the Year since this report shows the attendance tally for the entire year. Refer to Criterion D1 - Recycler Attendance Report on Page 137.
produce a *points list* of teachers' recycling performance;	The Recycling Activity Report produces this point list and even allows the user to print the report. This is useful to display on the school's bulletin board. Refer to Criterion D1 Recycling Activity Report on Pages 132-135.
hold data securely and *prevent unauthorised access*;	The implementation of the password screen and the secret question prompt, which can be used when the user does not remember the password, meets this objective. Refer to Stage D1, Page 116 & 117.
***allow users to edit* the pre-set data *easily* which is used for look-up;**	The Teachers' Classroom Allocations Table and Grade Form Teachers Table are the two main reference tables in the previous database that have been implemented into the Recycling Activity Monitoring System. Refer to Criterion D1 for the Teachers & Classrooms Plan and Form Class Locations on Pages 142-145 & 147.
produce a *list of candidates* for the Recycler of the Month award for any given month;	The Recycler of the Month Candidate Criterion screen and Report executes this process perfectly. This eliminates the major problem of the Access system. Refer to Criterion D1 Recycler of the Month Candidate Report on Page 140.
use *simple user-friendly interfaces* that do not confuse the user and allows quick input;	The fact that the program has solely been designed for the Recycling Monitoring Activity System satisfies this aim. The reason for setting this objective was because the Microsoft Access database was a general purpose software that included many irrelevant features that were confusing. The array of buttons and technical jargon used throughout the database was very confusing for her but the Java program directly caters to the Eco Club's needs.
***process the input data automatically* without having to initiate each procedure manually in turn;**	When the 'Save' button is clicked, all processes are executed in the appropriate order without the need of manual initiation. Refer to Page Criterion C1 to view the code that initiates the process of referencing form classes, updating the recycling

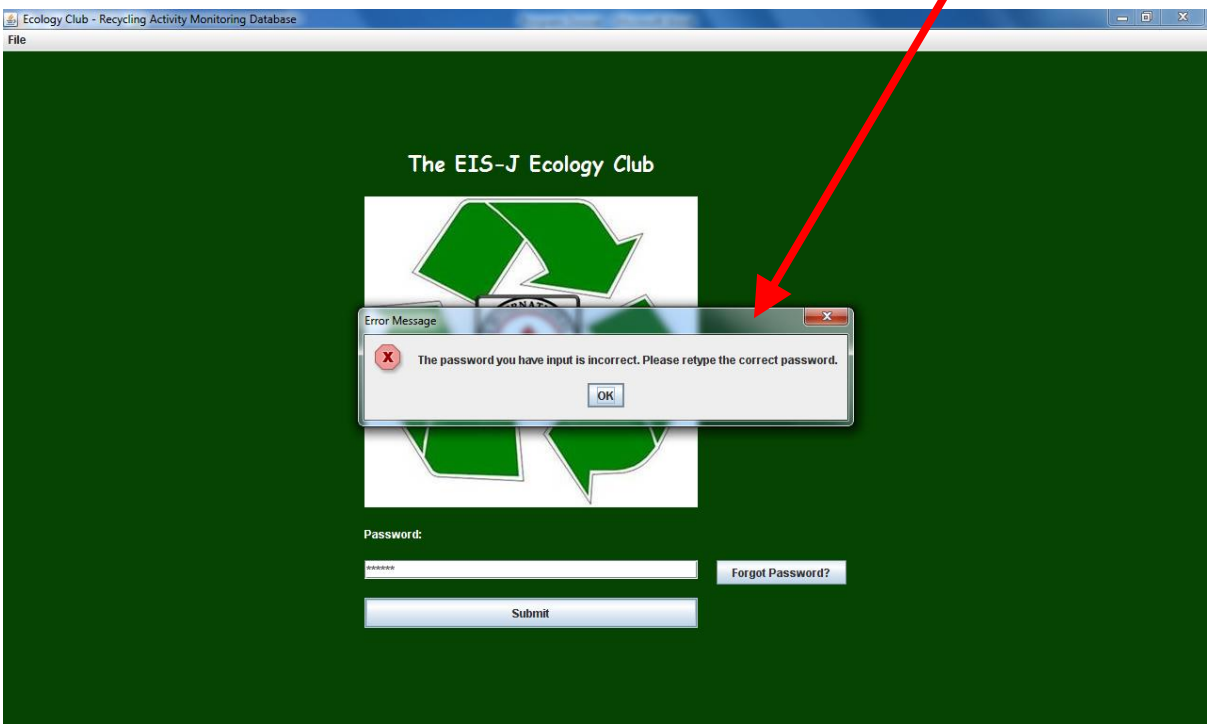
	statistics and keeping a log of the data that has been input.
take up *less space* on the computer hard disk;	Since the program is coded solely for this purpose with no extraordinary features that affect the computer system, very little space is used up. A total of 200KB is occupied by the program and its random access files. However, the size of the random access files will vary depending on the amount of data stored in them. For evidence of the size, please see Appendix (Page 161).
***keep a record* of all the recycling activity statistics that have been processed;**	The random access file "RecyclingActivityLog.txt" stores the details of all recycling statistics that are being processed. Refer to the logRecyStats method on line 1031 of Criterion C1. In addition, the RecyclingActivityLog text file can be found on Page 162.
***handle errors* and any sort of extreme/invalid data input.**	This objective is extensively addressed in the previous section; Criterion C2 – Handling Errors.

Criterion D1: Including an Annotated Hard Copy of the Test Output

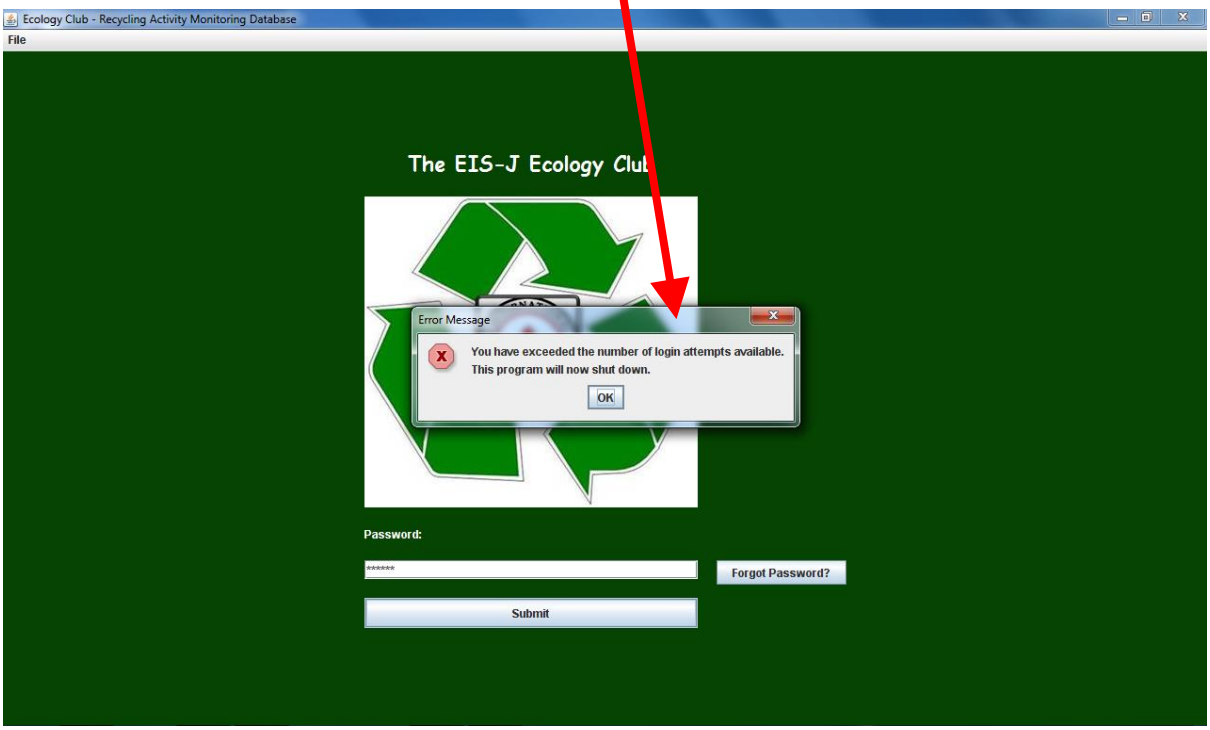
Program Output

This section will outline the output of the program. The following screen displays the password screen which is initiated on start-up of the program.

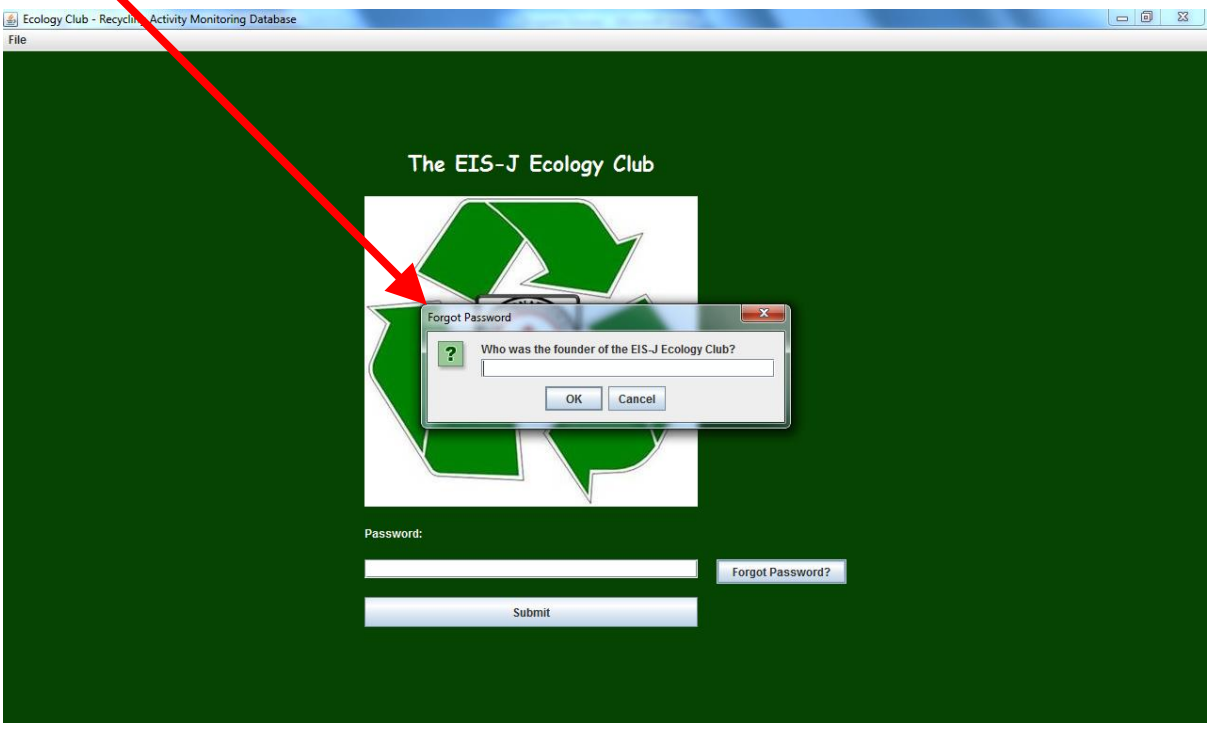
The set password is “waterfall”. If anything else is input such as “cooper”, the following error message appears. The password field is also case sensitive.



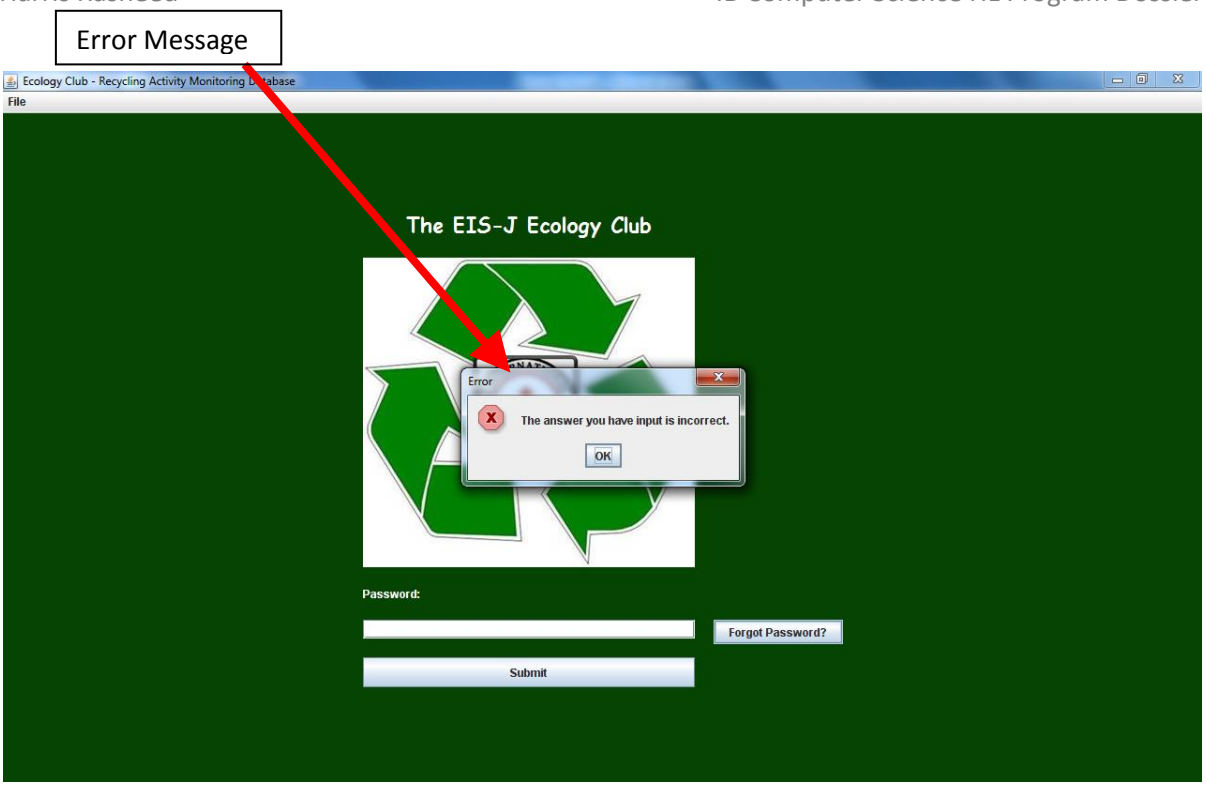
If the password is input wrong five times, the following prompt appears and the program terminates because of excess login attempts.



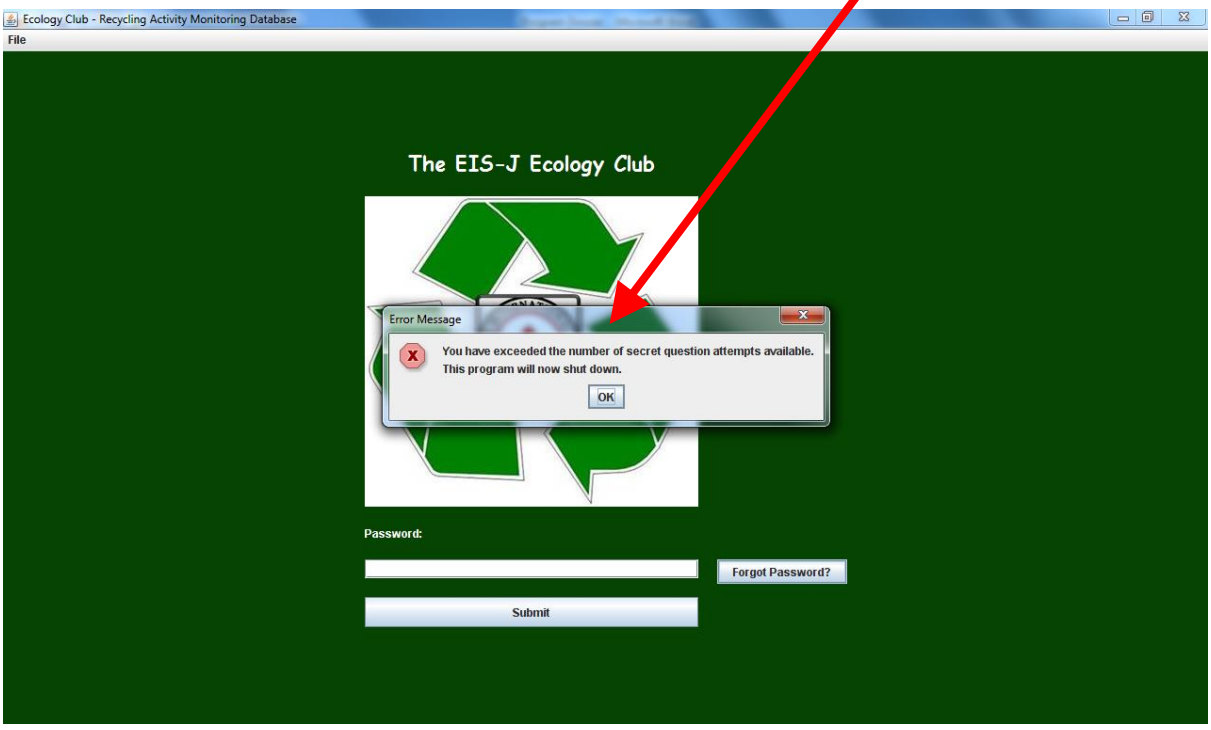
In addition, the same process occurs for the Forgot Password option. When the button is pressed, the following input dialog box appears.



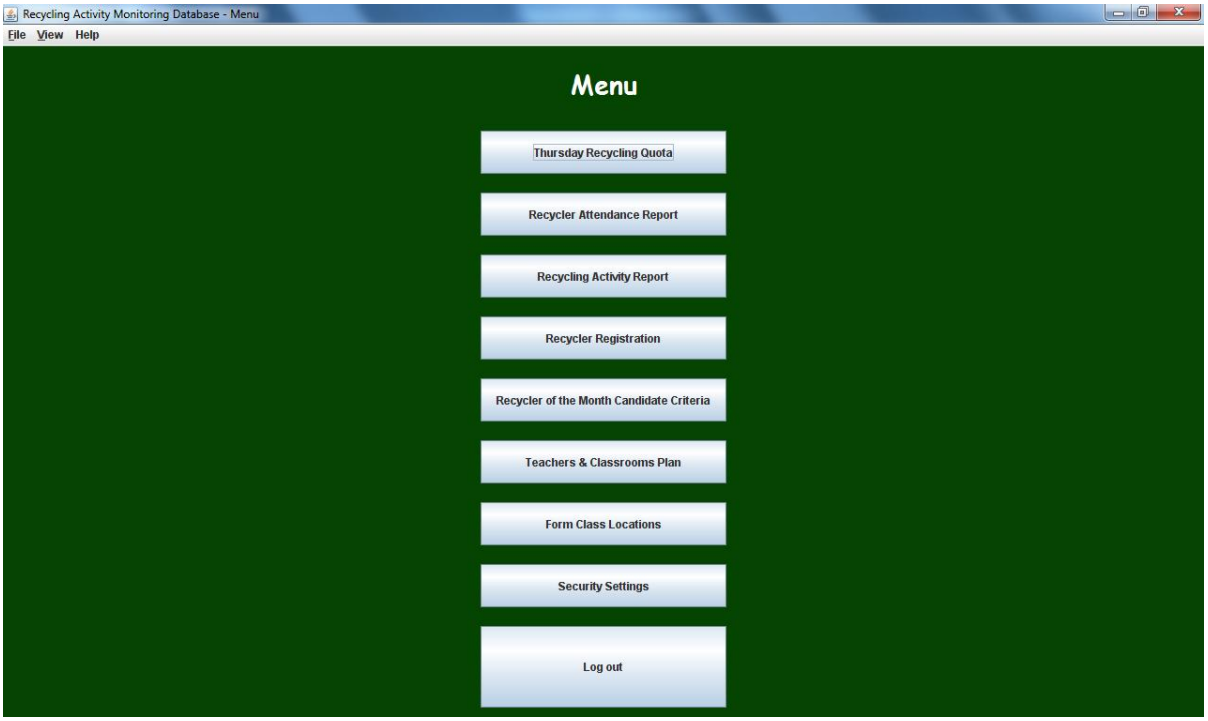
The answer is “Mrs Kelly Pugin”. If anything else is input, an error message will occur as shown on the next page.



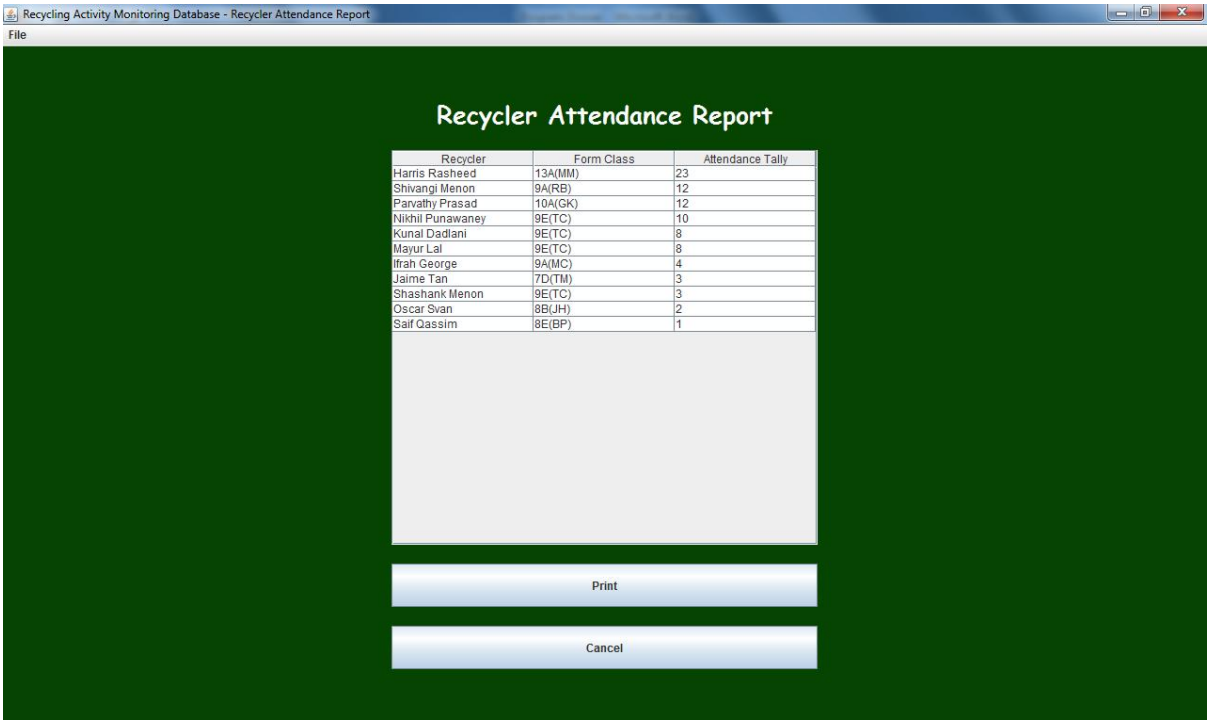
When three failed login attempts via the secret question have occurred, a prompt appears and the program shuts down once it is closed.



If the password or the secret question is correctly answered, the user will be logged onto the system and the menu screen will then appear. This screen is shown on the next page.

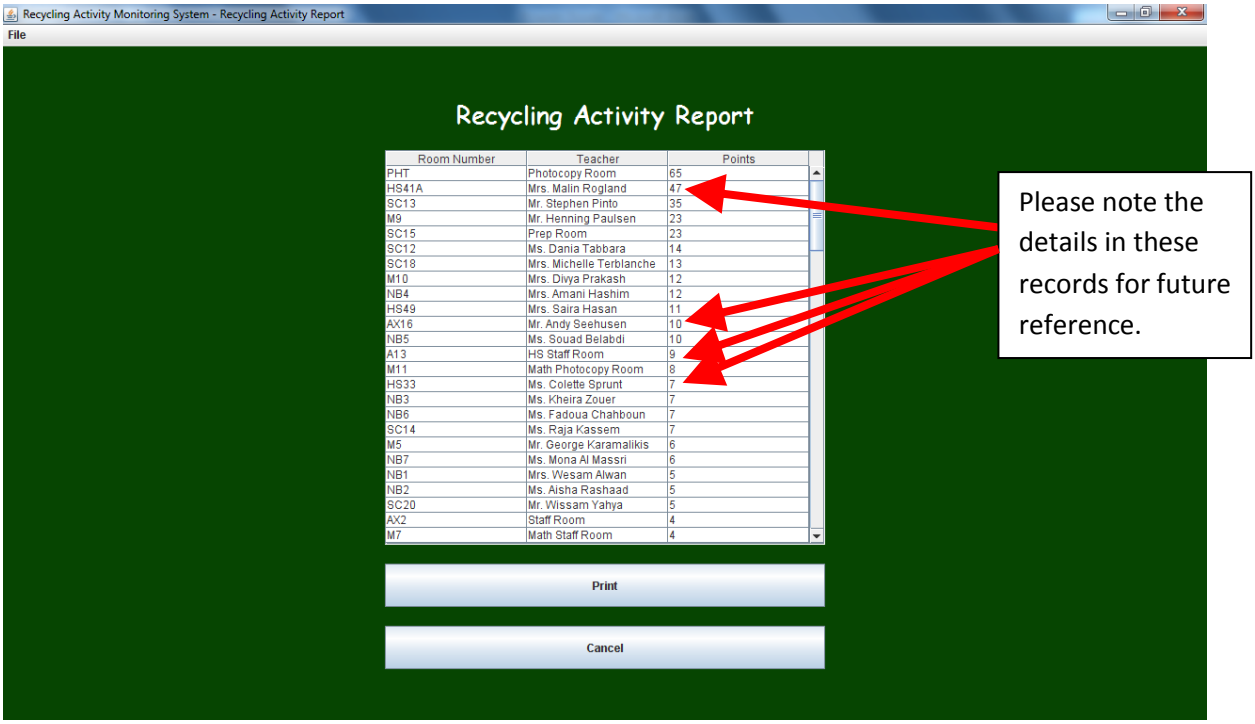


The menu screen allows the user to select from various functions. If the user has to input data of recycling statistics. They would have to click the Thursday Recycling Quota button. Having input some sample data already, the Recycler Attendance Report looks as follows.

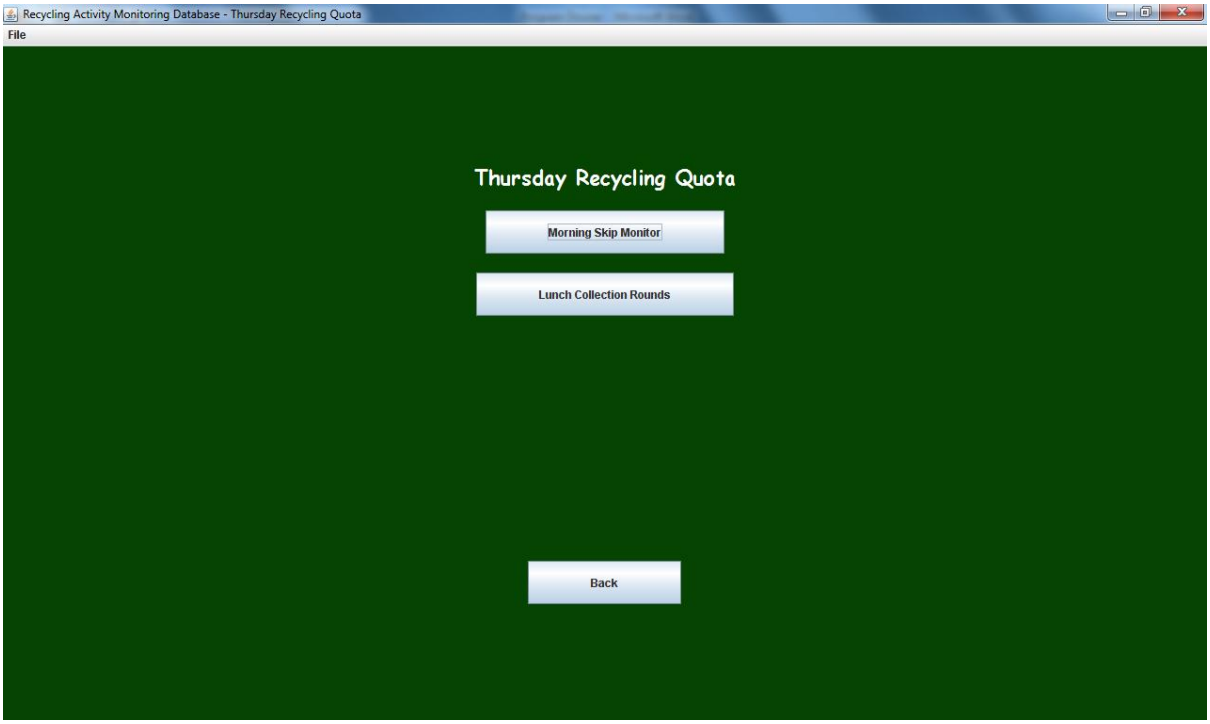


As can be seen, the report is sorted in descending order which is very useful to recognise the top recyclers for Mrs. Cooper when viewing this report. This report is the tally format that Mrs. Cooper wanted the system to produce.

The following is the second most crucial report of the system; the Recycling Activity Report.

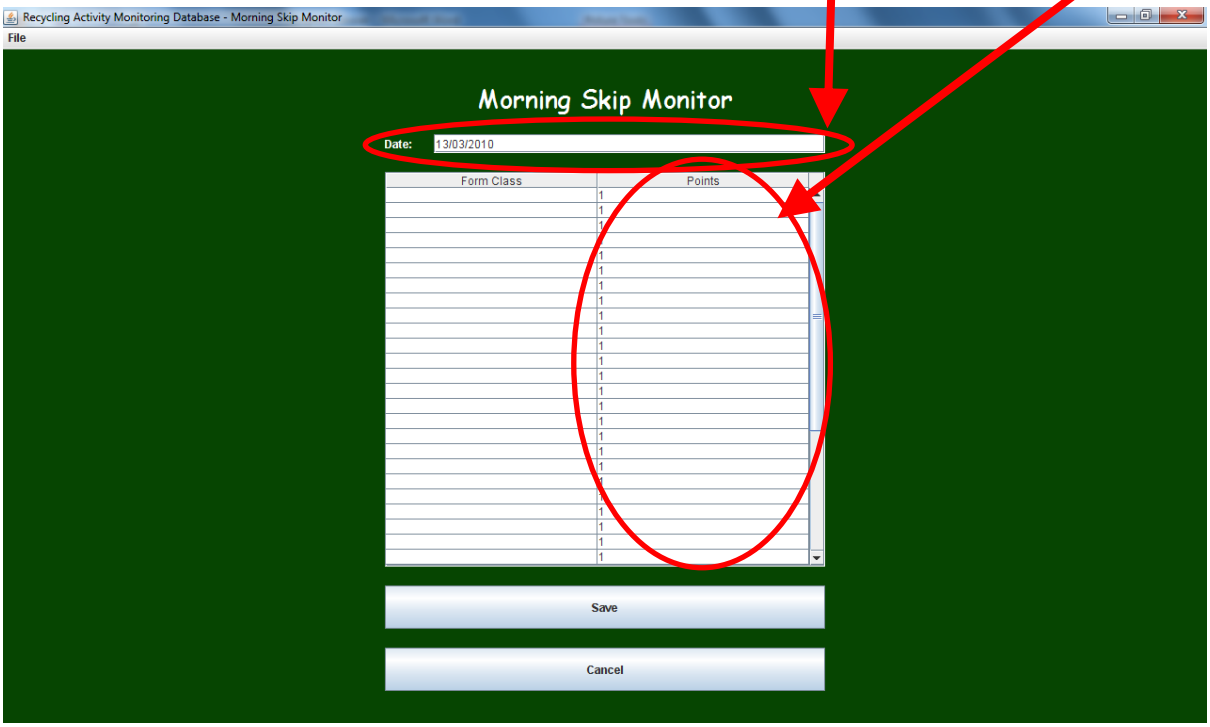


The next screenshot shows the Thursday Recycling Quota screen. It was essential to present the two previous screenshots because they will be used to verify that the program functions as planned.

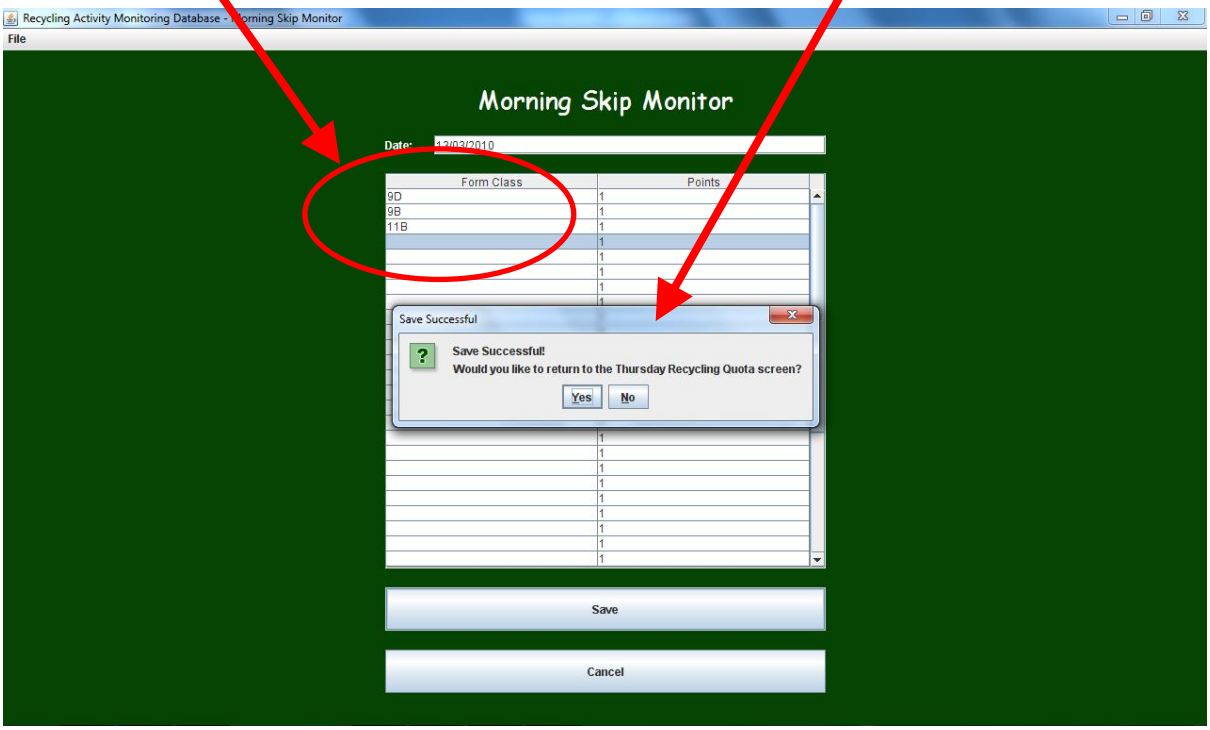


Every week, both the Morning Skip Monitor and Lunch Collection Rounds data is collected. Therefore, the user can choose to input them in any order by simply clicking on the relevant button. The screen on the following page shows the Morning Ship Monitor input screen.

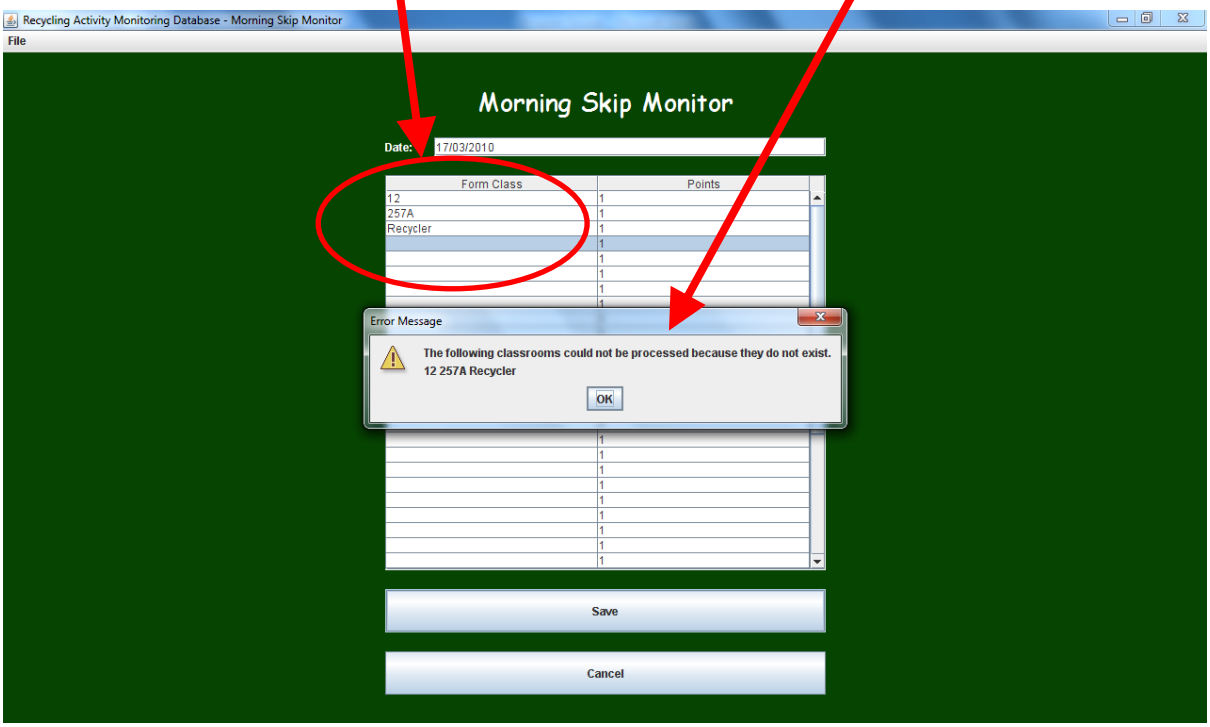
As can be seen, the system date is automatically set in the date field. If the date that the recycling statistics took place was different, it can simply be changed manually. The point list is also set to one by default but it can be changed if a form class achieved more points.



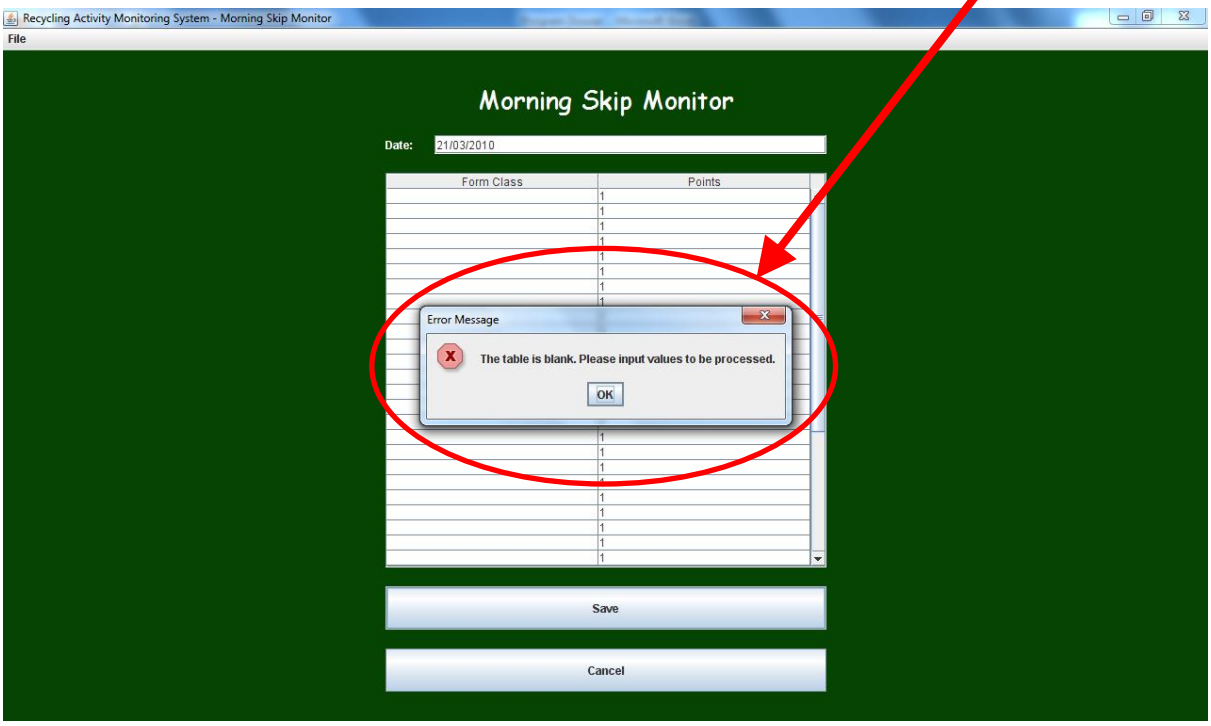
When data is input into the 'Form Class' column and the Save button is pressed, the following prompt appears from which the user can return to the Thursday Recycling Quota screen. In this instance, valid data is input into the system (9D, 9B, 11B).



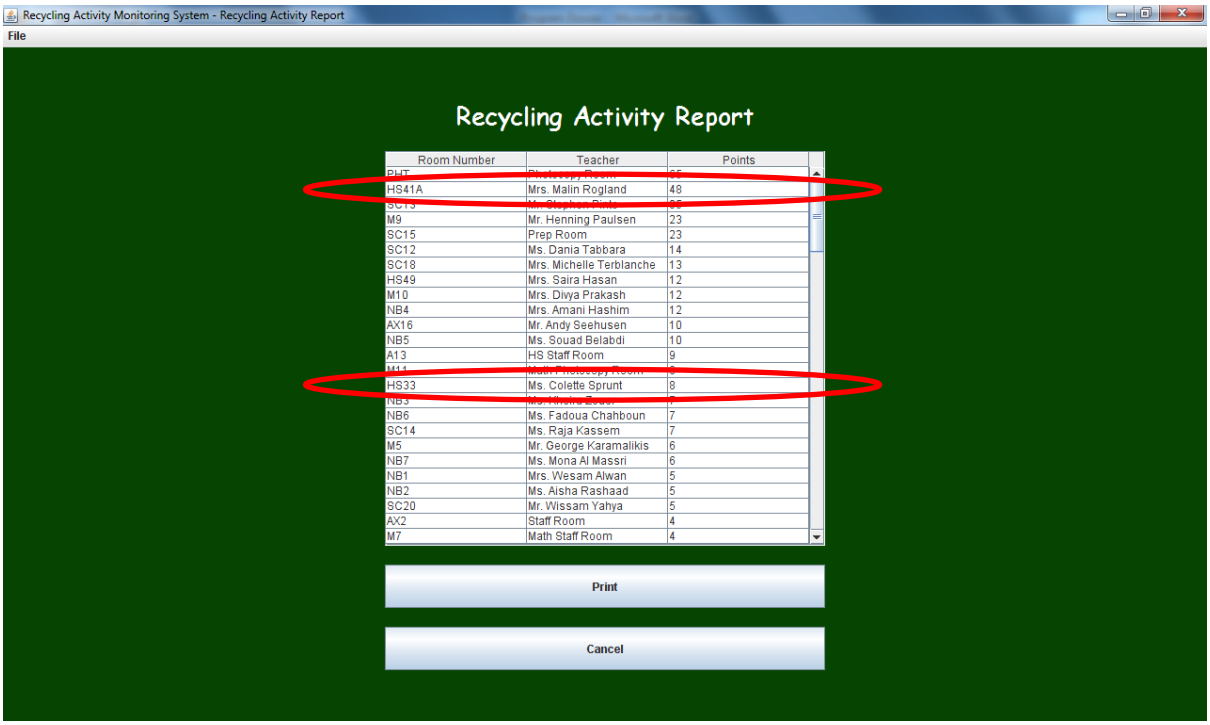
The next screen demonstrates the error handling of the Morning Skip Monitor input screen when invalid data is input (12, 257A, Recycler). These errors are caught by the program if they are not referenced to a room number.



In addition if the user attempts to click the Save button when the table is blank, the following error message will be returned.

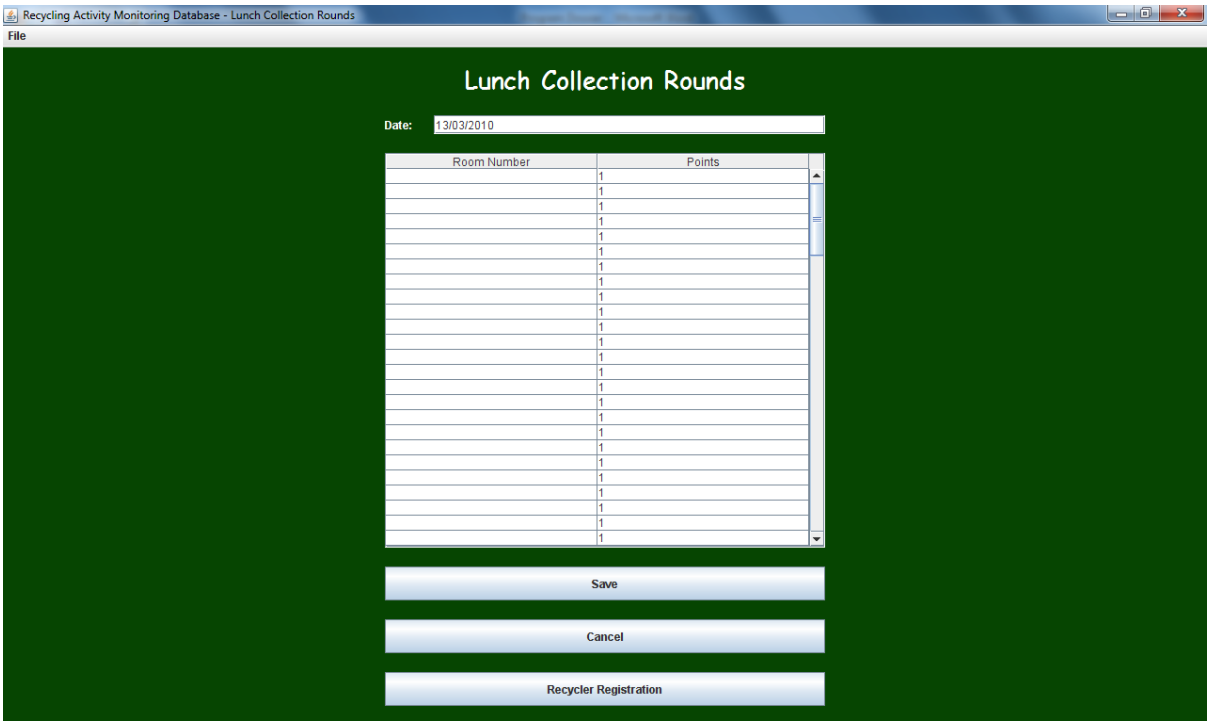


When the Morning Skip Monitor data has been added as shown before, the random access file is updated and the Recycling Activity Report updates as well.

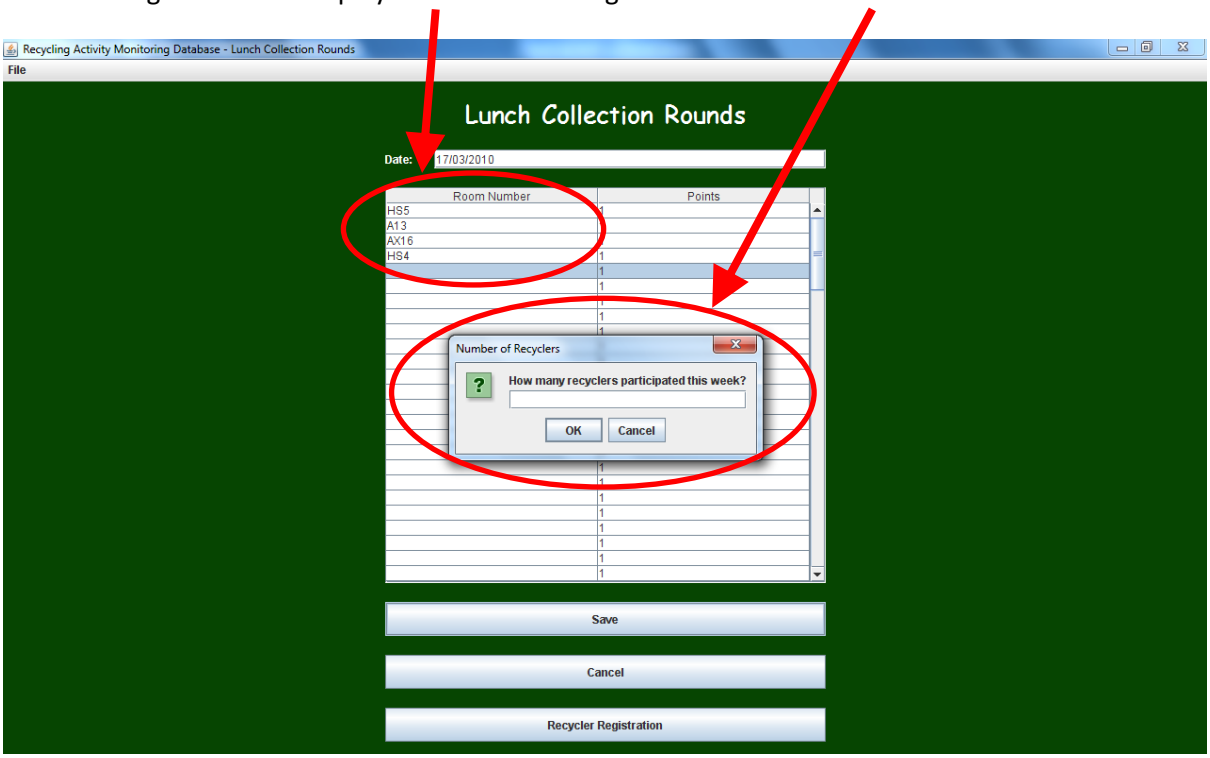


This screenshot above shows that HS41A – 9D and HS33 – 11B has an increased tally of one point. This can be compared to the first screenshot on Page 120. HS49 – 9B is further down the list but has also updated by one point.

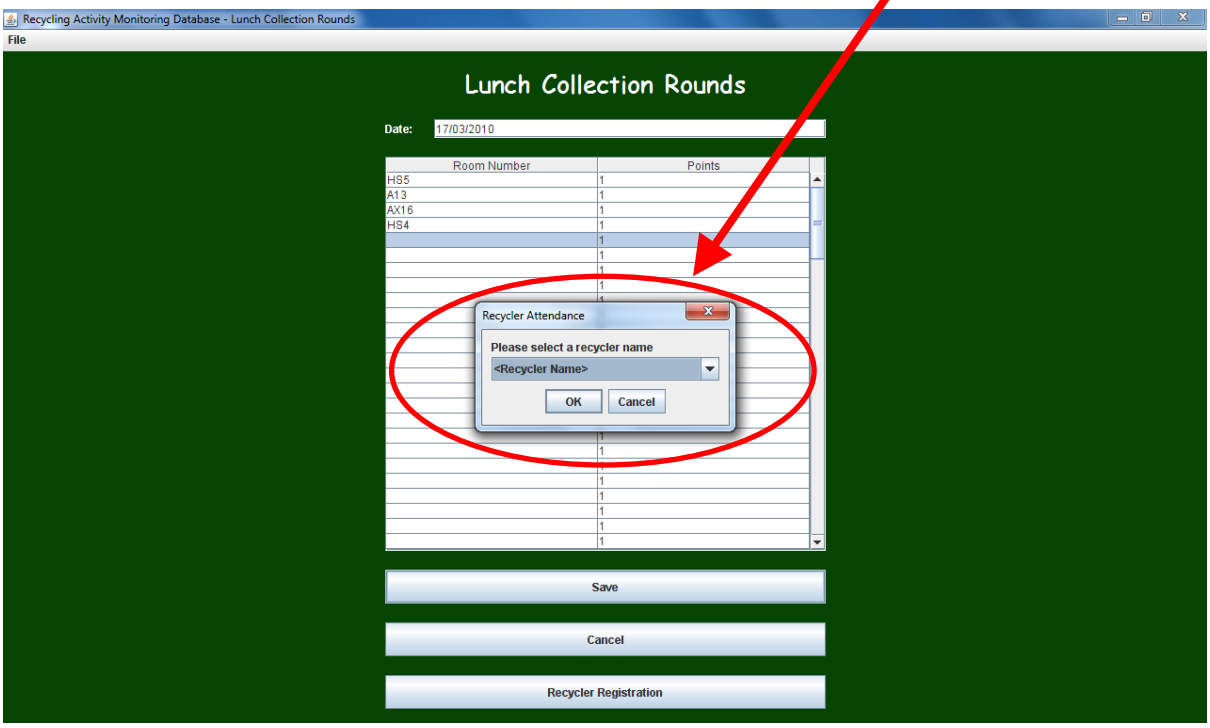
The next input screen in the Thursday Recycling Quota menu is the Lunch Collection Rounds screen. The following screenshot shows how the screen looks once accessed.



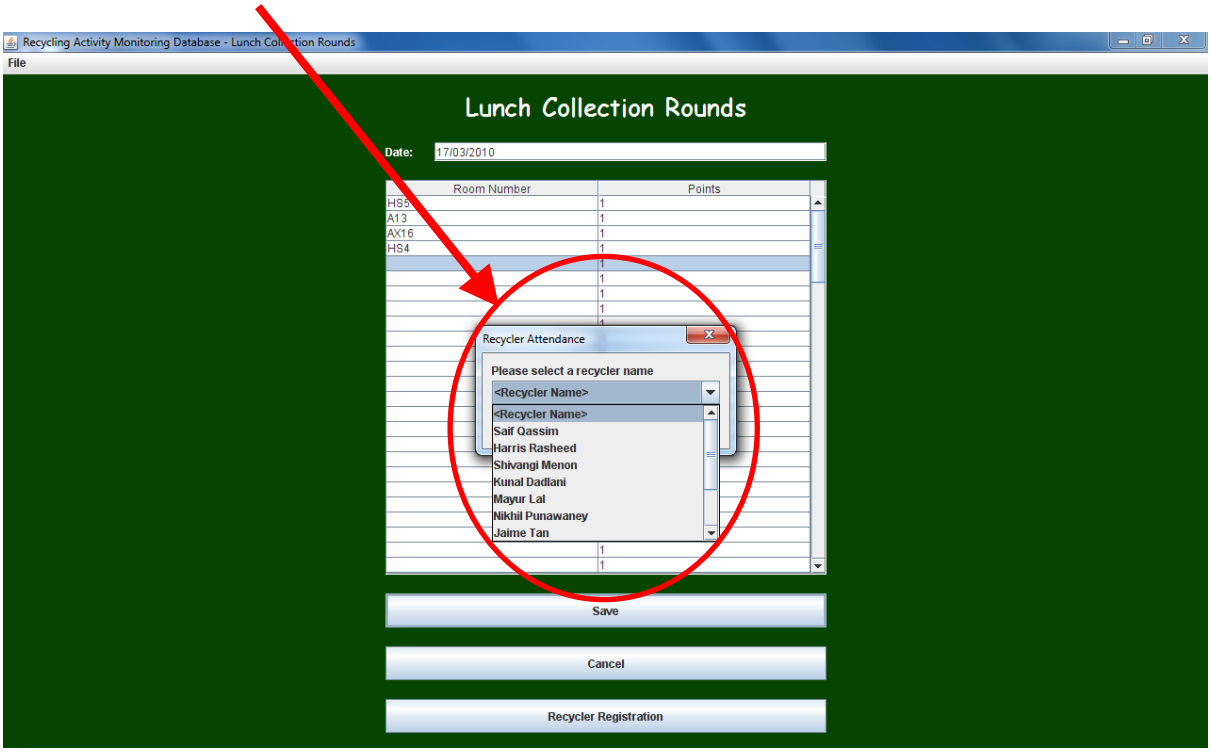
Data can be input as before with the Morning Skip Monitor screen, however, the room numbers must be input into the first column, in this case, as recyclers note down this data on the rota slips. The following screenshot displays a table containing valid data when the Save button was clicked.



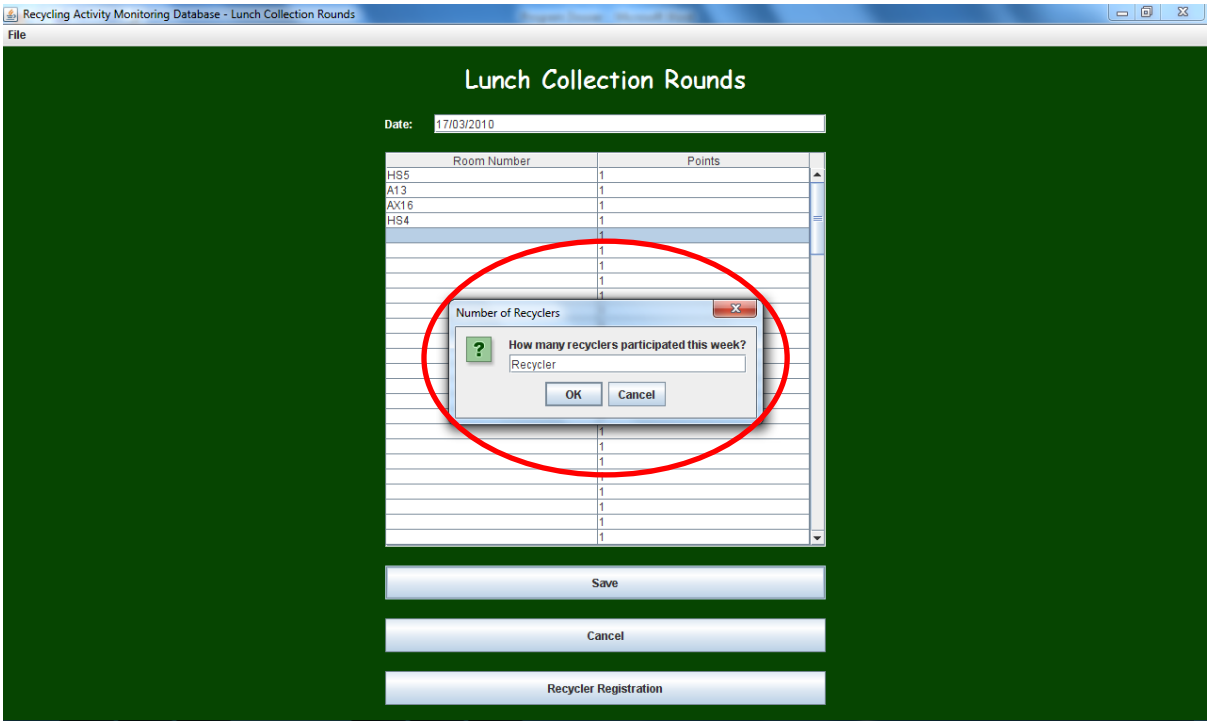
A prompt has appeared above asking how many recyclers participated in the collection for the week. A number needs to be input at this stage. Depending on the number input, the user will be prompted to enter the names of all recyclers that participated in the rounds. When the number five, for example, is input and the user presses 'OK', the screen changes to the following.



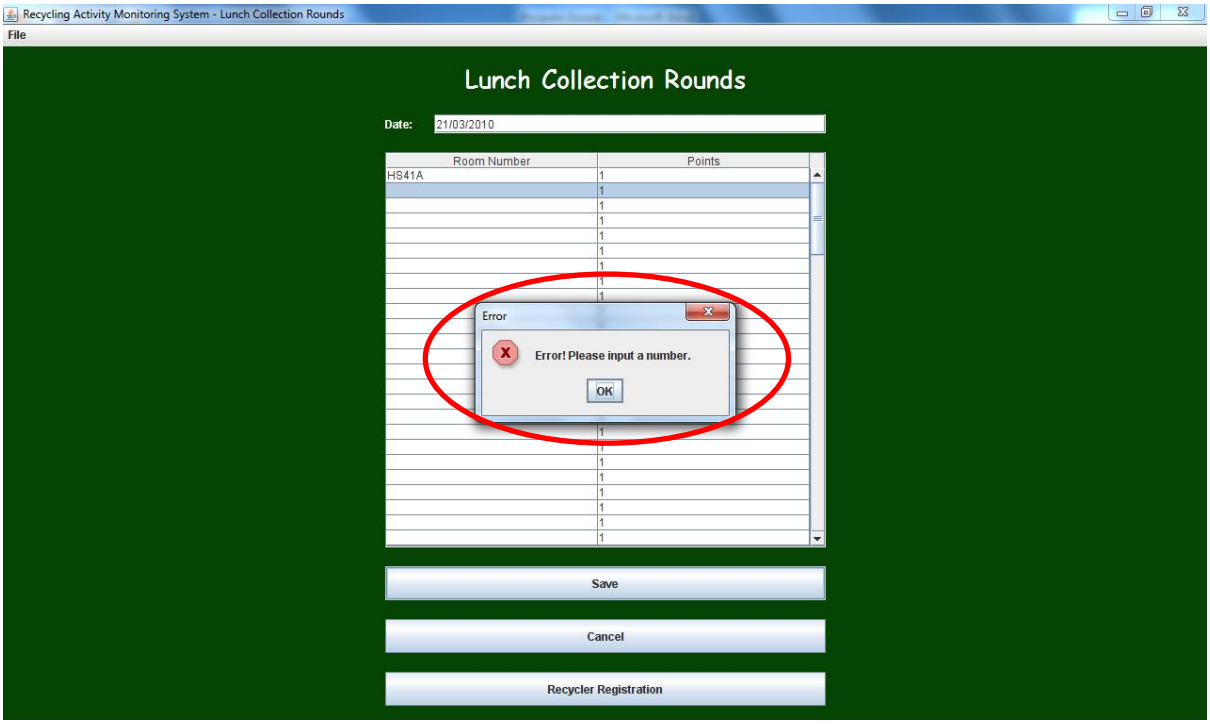
A recycler can be selected from the choice list. This list only includes the names of recyclers who are registered in the system.



After choosing a recycler and pressing 'OK', the same prompt will loop and appear four more times so that the four other recycler names can be input. At the end of the loop, the prompt disappears and the screen remains on the Lunch Collection Rounds screen. However, if the Save button is clicked and invalid data is input into the dialog box as shown below, an error message would appear.

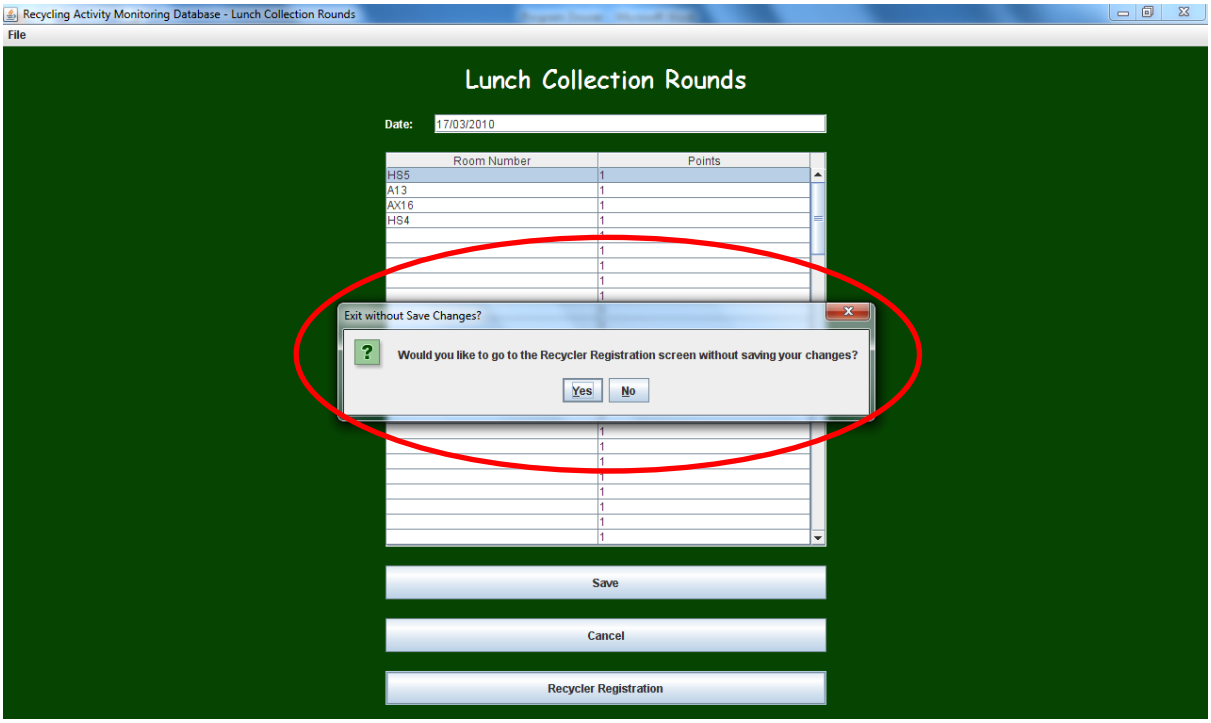


The error message is shown below.

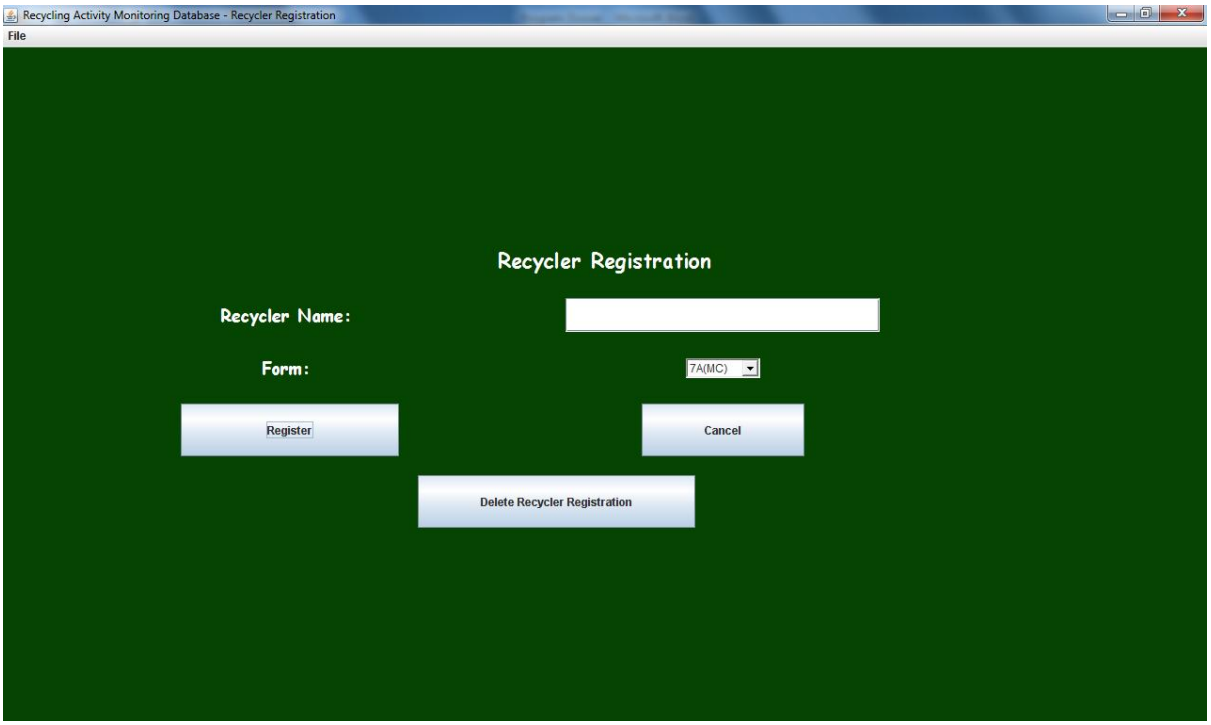


The following screenshot displays how the records have been processed when HS4, HS5, A13 and AX16 was input. The points have updated. The HS4 record can be viewed when scrolled down and it has been updated also. The HS5 statistic has increased and can now be seen on the top half of the table.

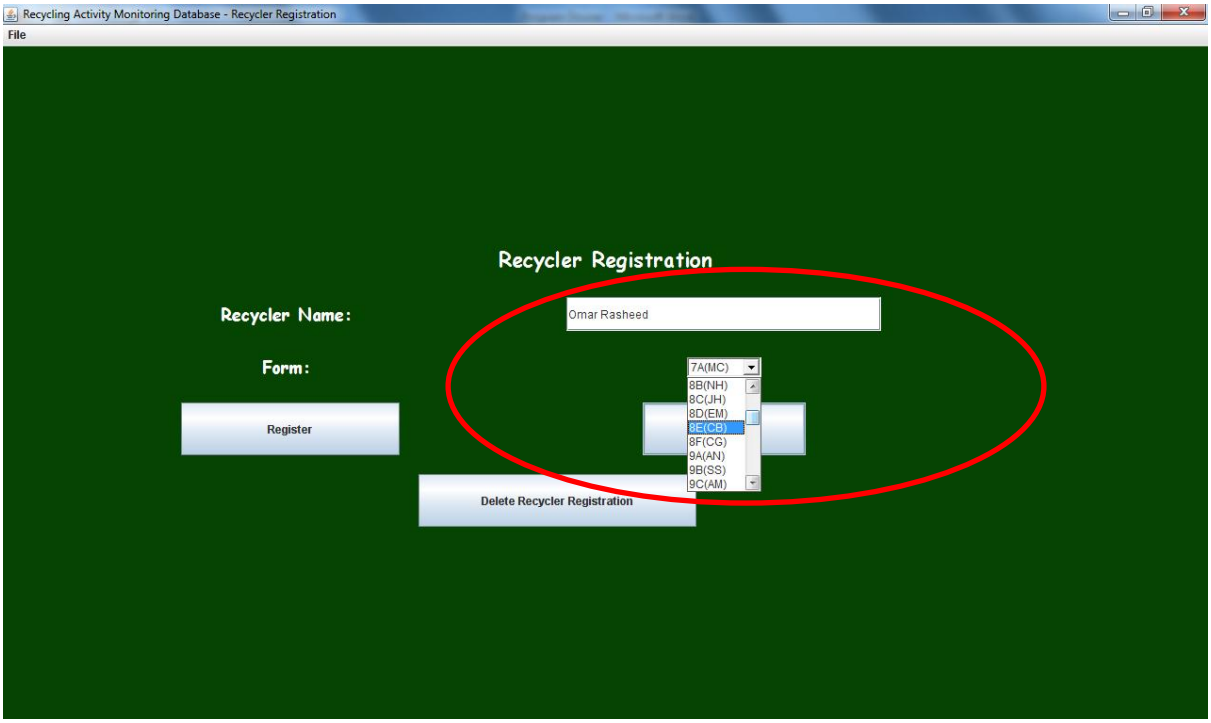
The Recycler Registration screen can also be pressed which will allow the user to add a new recycler to the recycler list. The following prompt appears when the Recycler Registration button is clicked.



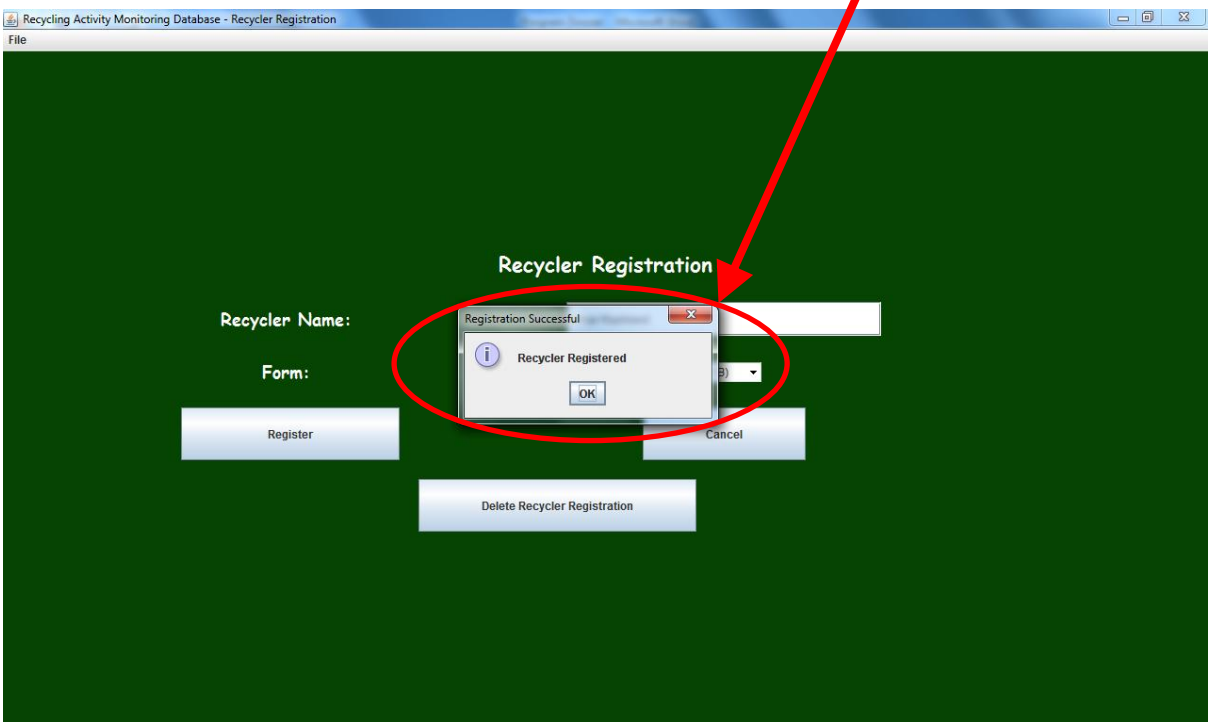
If the user presses No, the screen will stay on the Lunch Collection Rounds screen. If the users clicks Yes, the screen will change to the following.



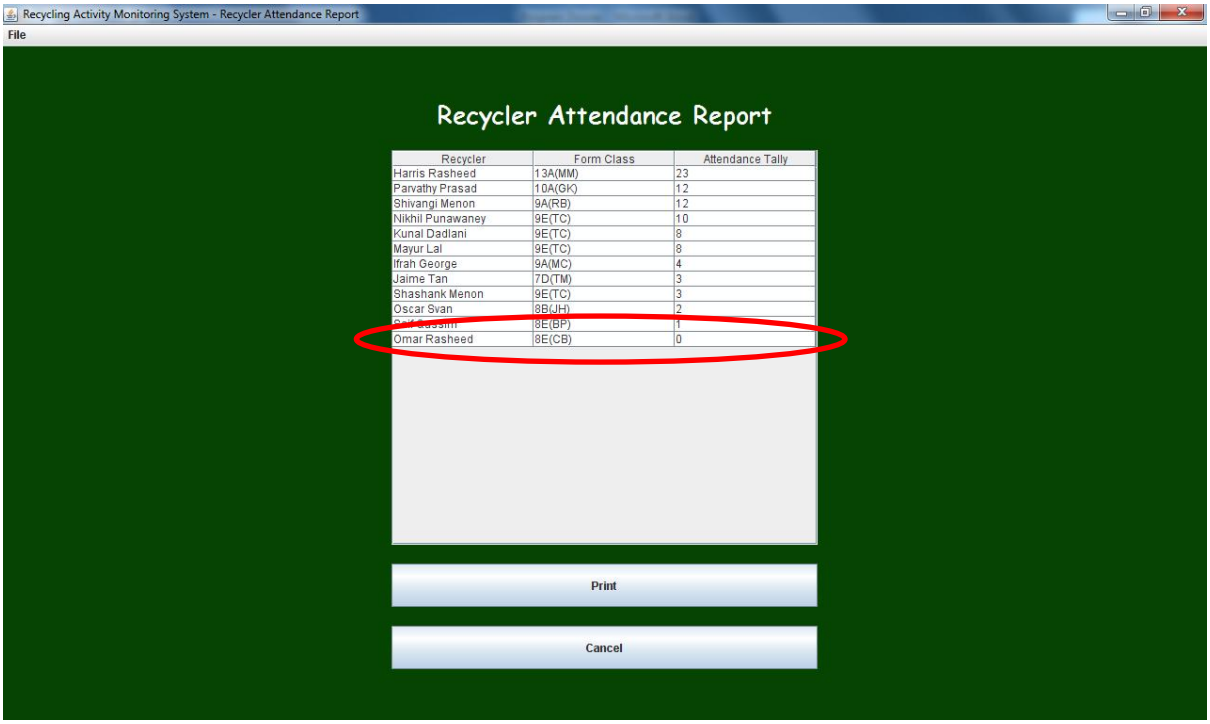
Although the layout of this screen is slightly messy, it functions perfectly. The user can input the new recycler name into the text field and select the form class of the student as well. The next screenshot presents a sample Recycler Registration.



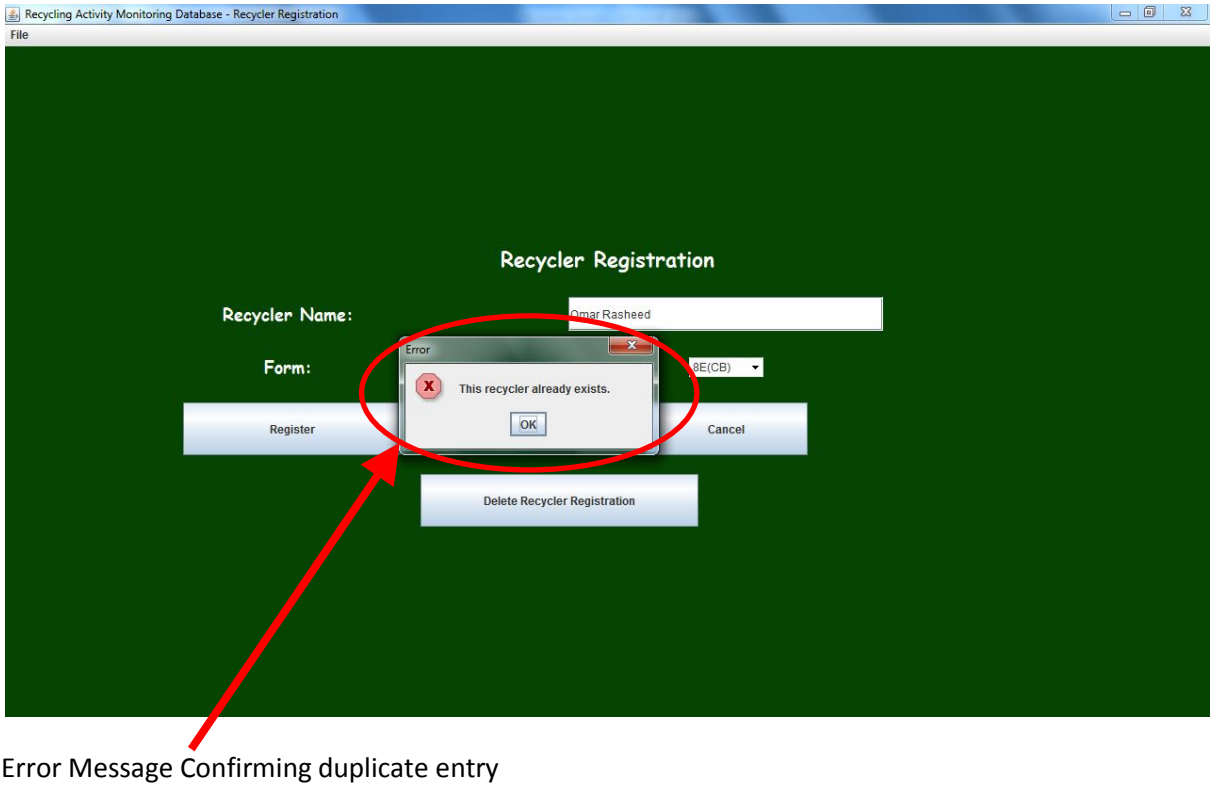
When the user registers the recycler, the following prompt appears confirming that the recycler has been registered into the system.



The next screenshot shows an updated version of the Recycler Attendance Report. Omar Rasheed is now included after his registration.

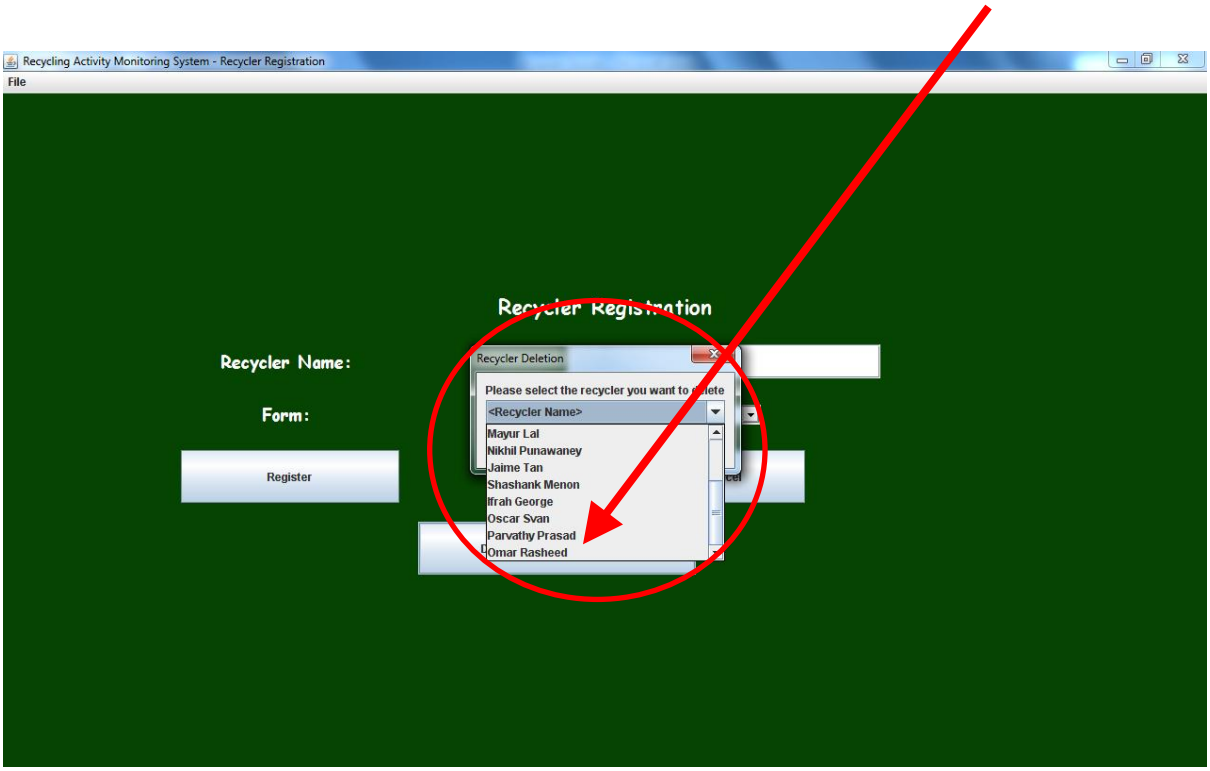


However if the user attempts to register a recycler that already exists, the screen will display as follows.

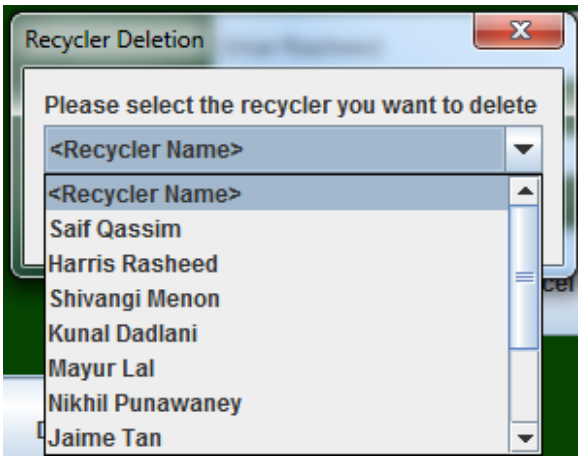


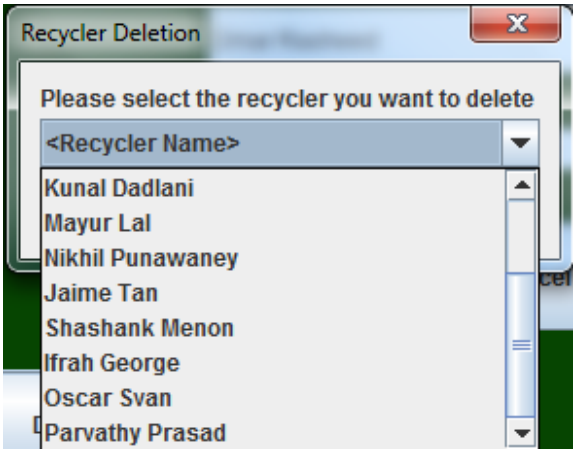
Error Message Confirming duplicate entry

The Delete Recycler Registration button is another option on this screen that can be pressed and an alternate prompt will appear allowing the user to select the recycler they want to delete from the system.

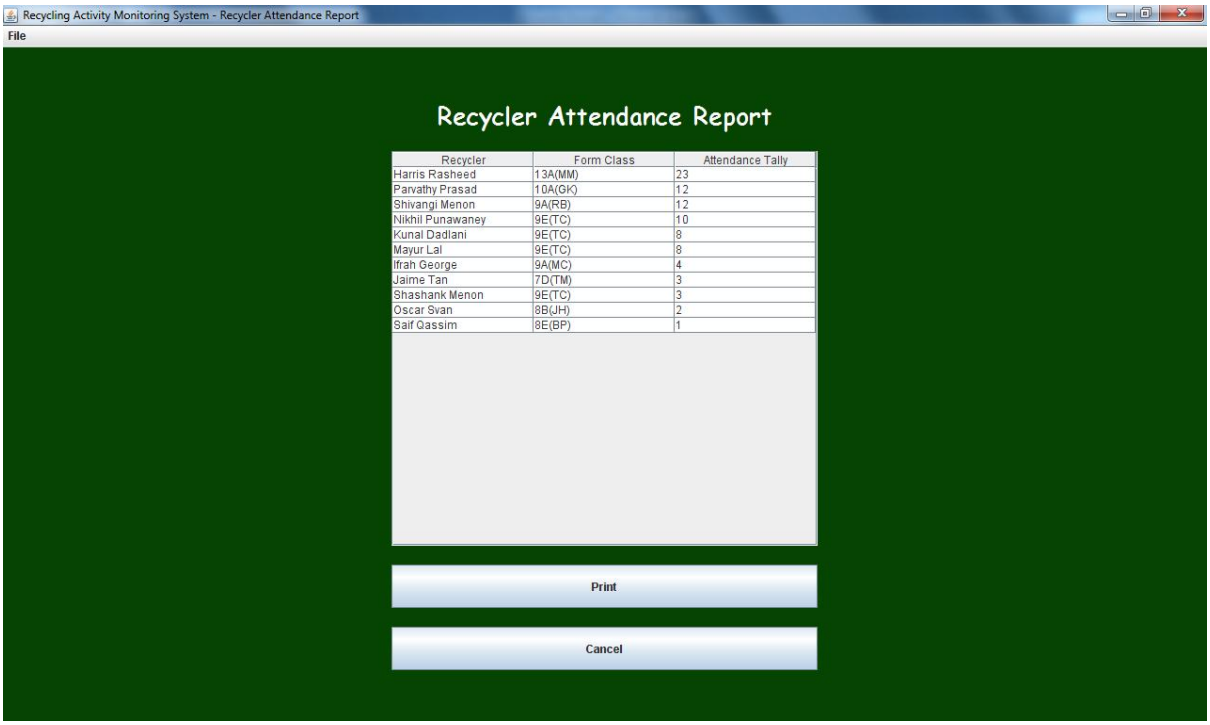


Omar Rasheed was selected and the OK button pressed. As a result, when I clicked Delete Recycler Registration button again, his name was not to be found on the list because the Deletion was successful.

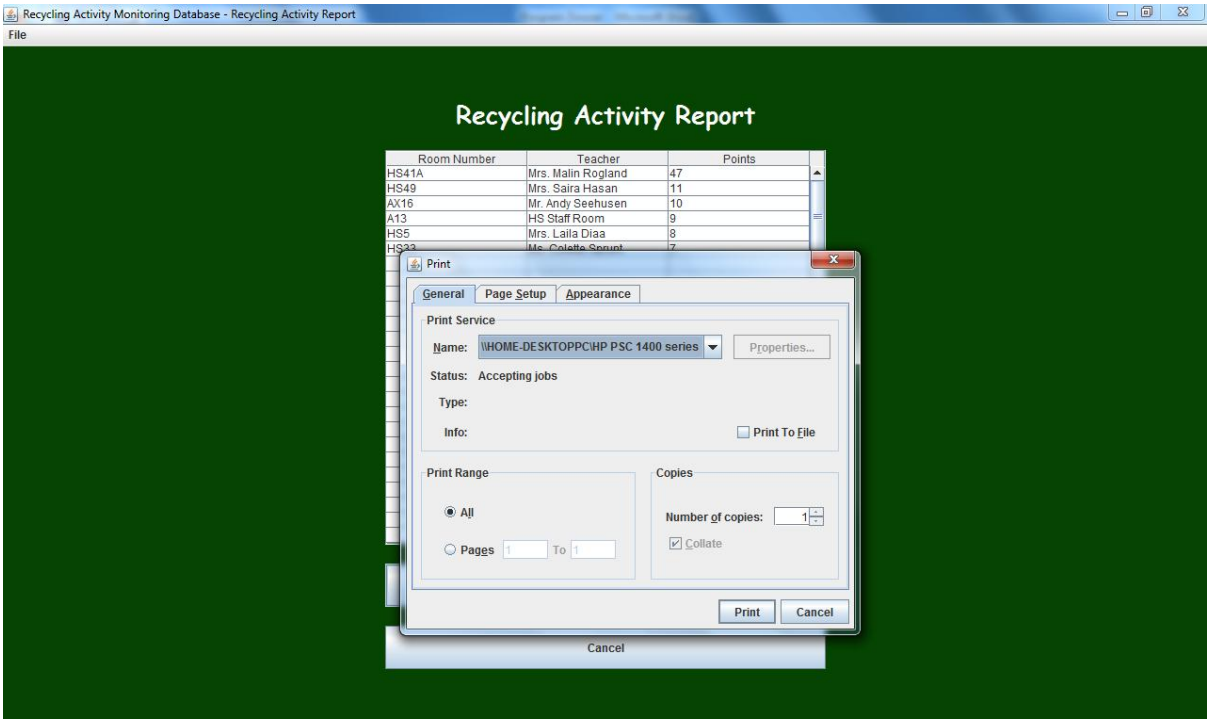




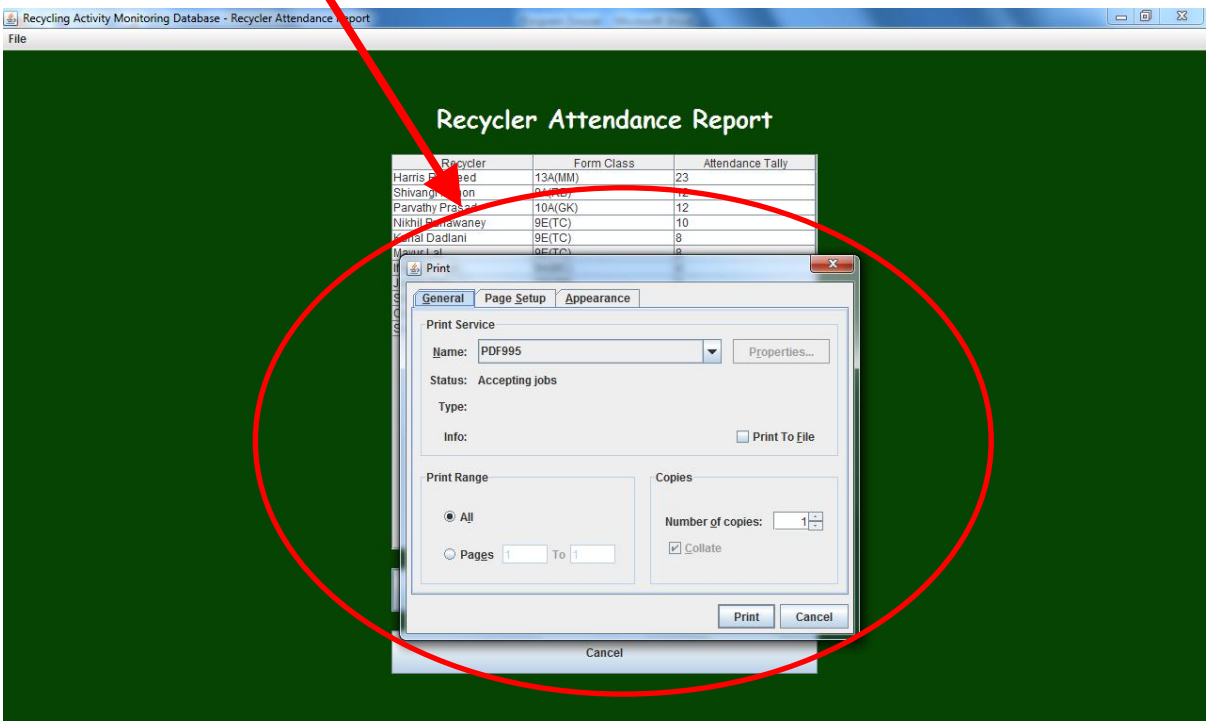
The screenshot below clearly shows that the recycler 'Omar Rasheed' has been deleted from the program's records and no longer exists.



Referring to the Recycling Activity Report, the following prompt appears when the print button is pressed. This allows the user to select printing preferences before printing. A printout of the report is displayed on the next few pages.

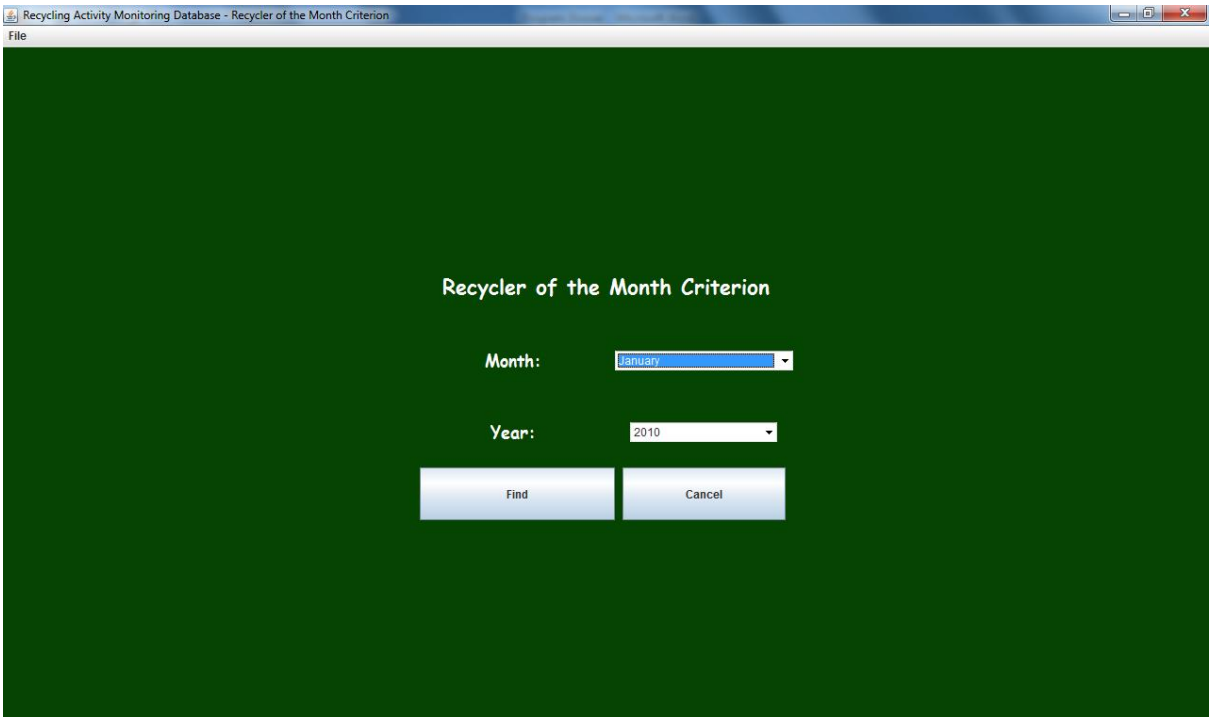


The same print dialog appears when the Print button is pressed for the Recycler Attendance Report.



The following page shows the printout of the Recycler Attendance Report.

The following screenshot shows the Recycler of the Month Criterion screen. From here, a month and year can be selected where a list of candidates will be produced on clicking Find.



After specifying the criterion as March 2010 and pressing the Find button, the following report was produced. This report produces a tally solely based on the month specified.

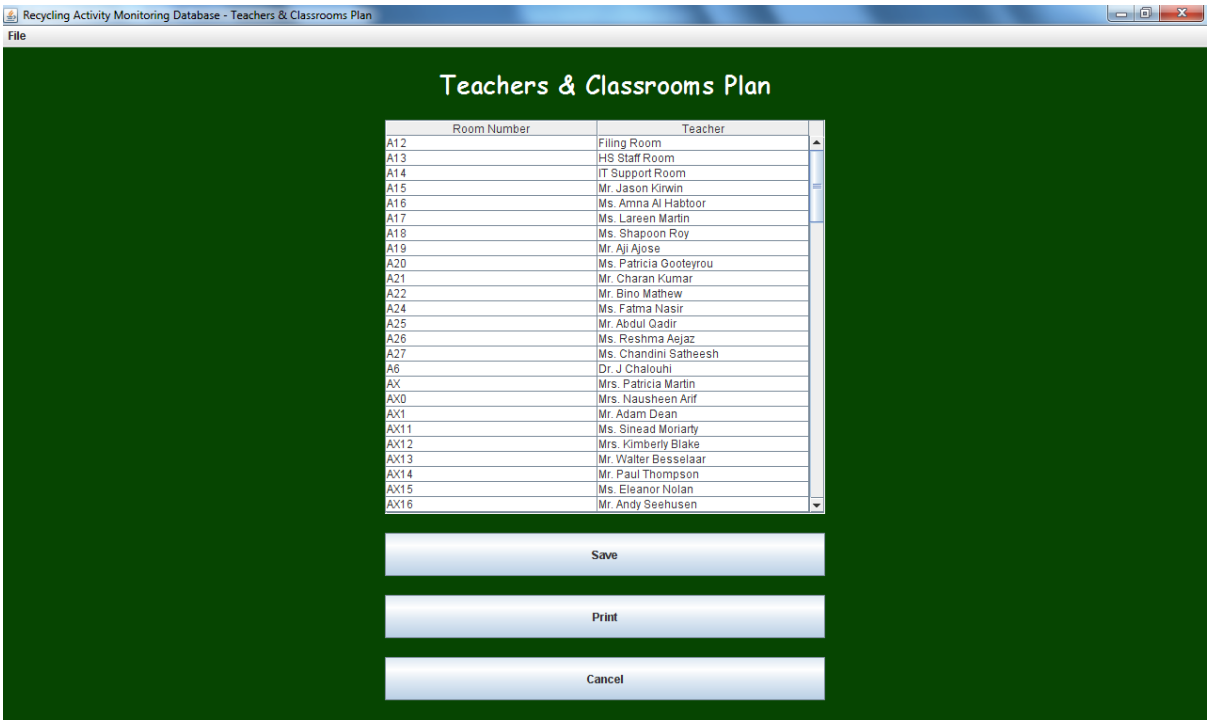
The above was produced based on the following data from a random access file:
Harris Rasheed 10/01/2010Shivangi Menon 17/03/2010Mayur Lal

17/03/2010Mayur Lal 17/03/2010Parvathy Prasad 17/03/2010Parvathy Prasad 17/03/2010

The report's results will always depend on the criterion specified. The current report displays a few records because only sample data has been entered into the system. When the system is in full implementation and data has been input, this report will grow larger in size and display all the possible candidates for the award.

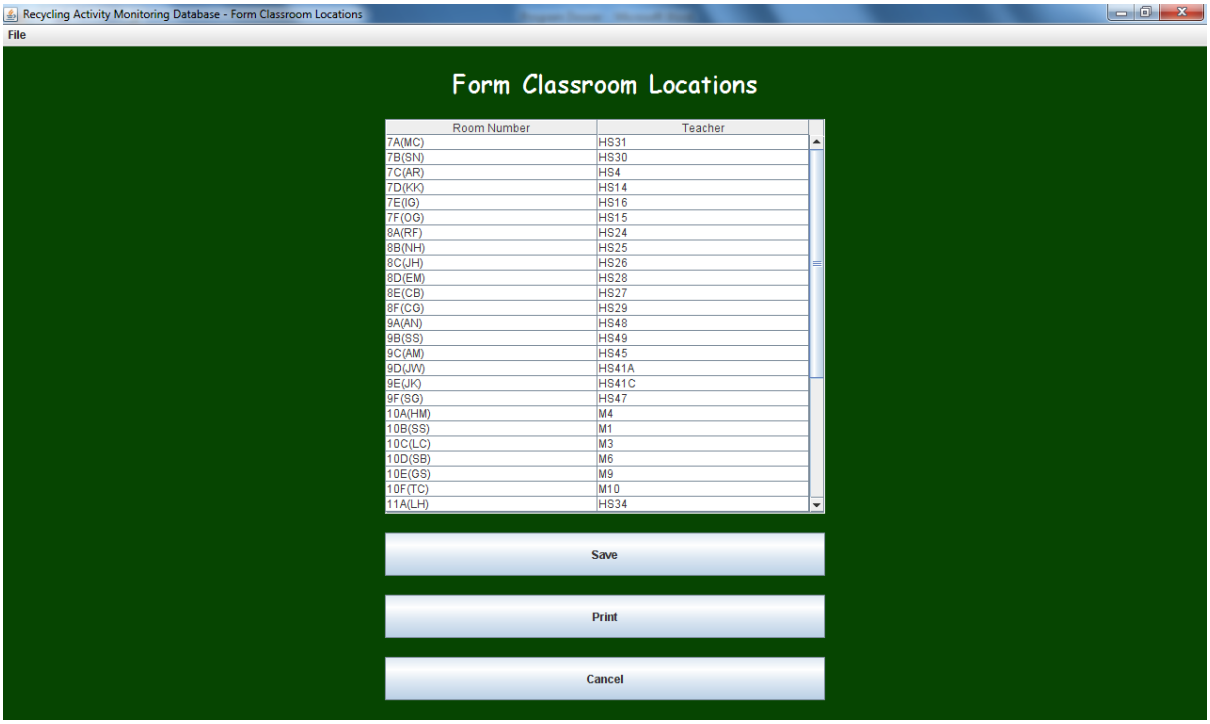
The following page shows a printout of the Recycler of the Month Candidate Report for March 2010.

The following screenshot displays the Teachers & Classrooms Plan reference table which can be updated and saved at anytime.



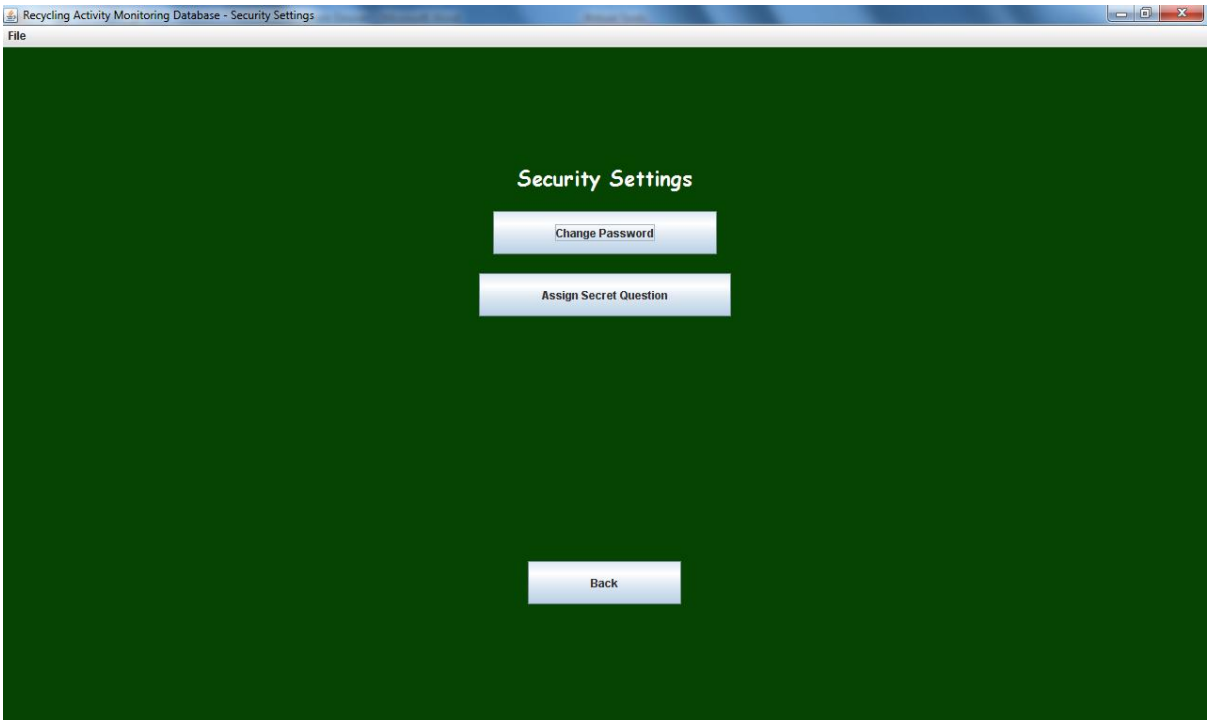
A printout of this table can be found on the next few pages.

The following screenshot displays the Form Classroom Locations reference table which is the most important reference table. Data in this table should be kept up-to-date at all times as when data is processed, cross-referencing is done with the data in this table and it is essential for it to be accurate so that results are also accurate.

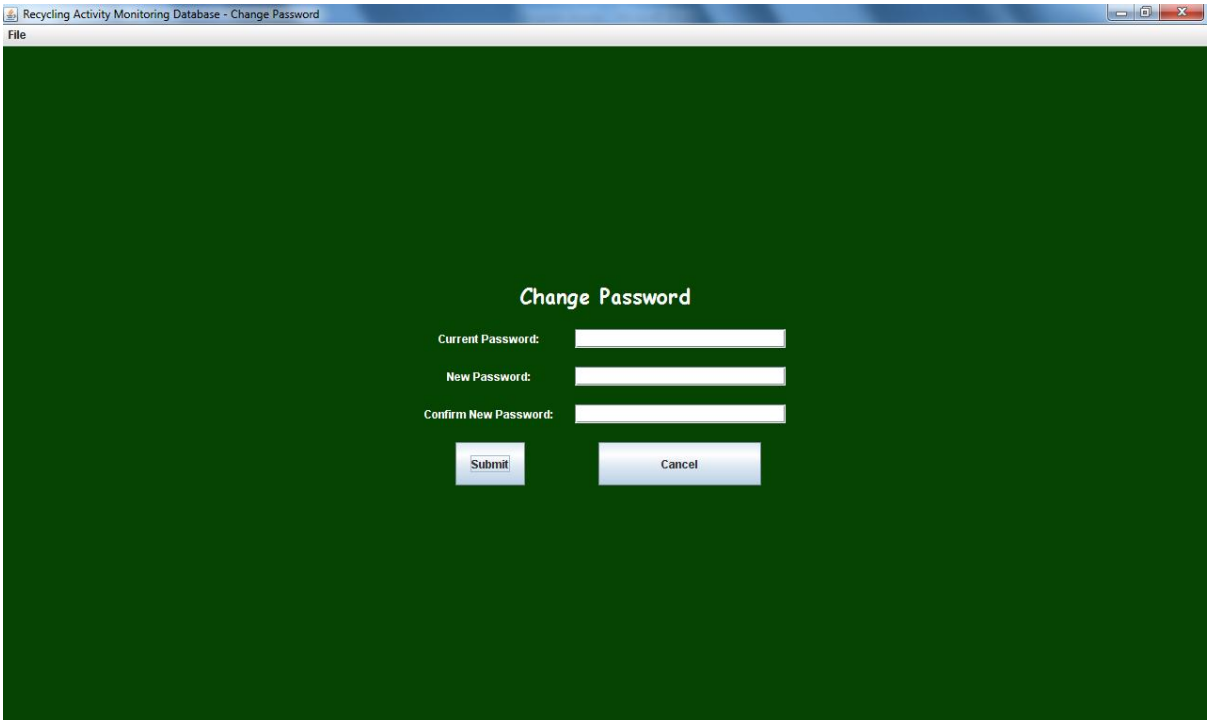


A printout of this table can be found on the next page.

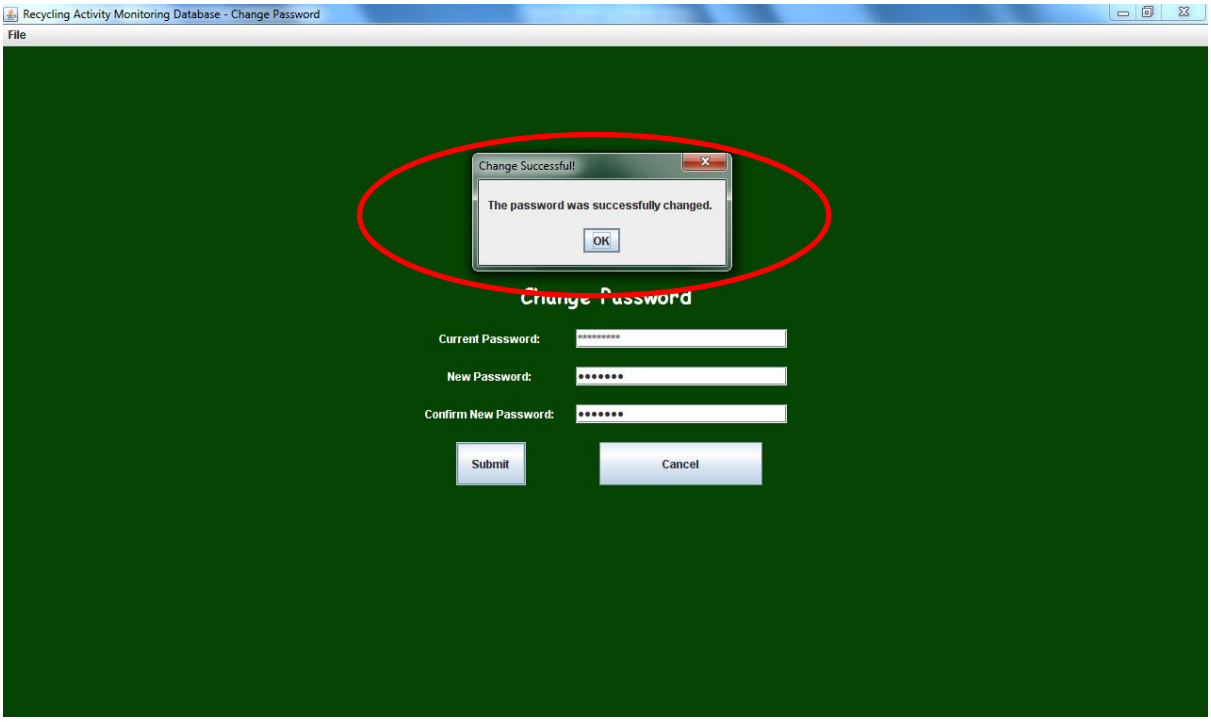
This screenshot shows the Security Settings screen. From here, the user has the option to change the system password, to assign a new secret question or to return to the main menu screen.



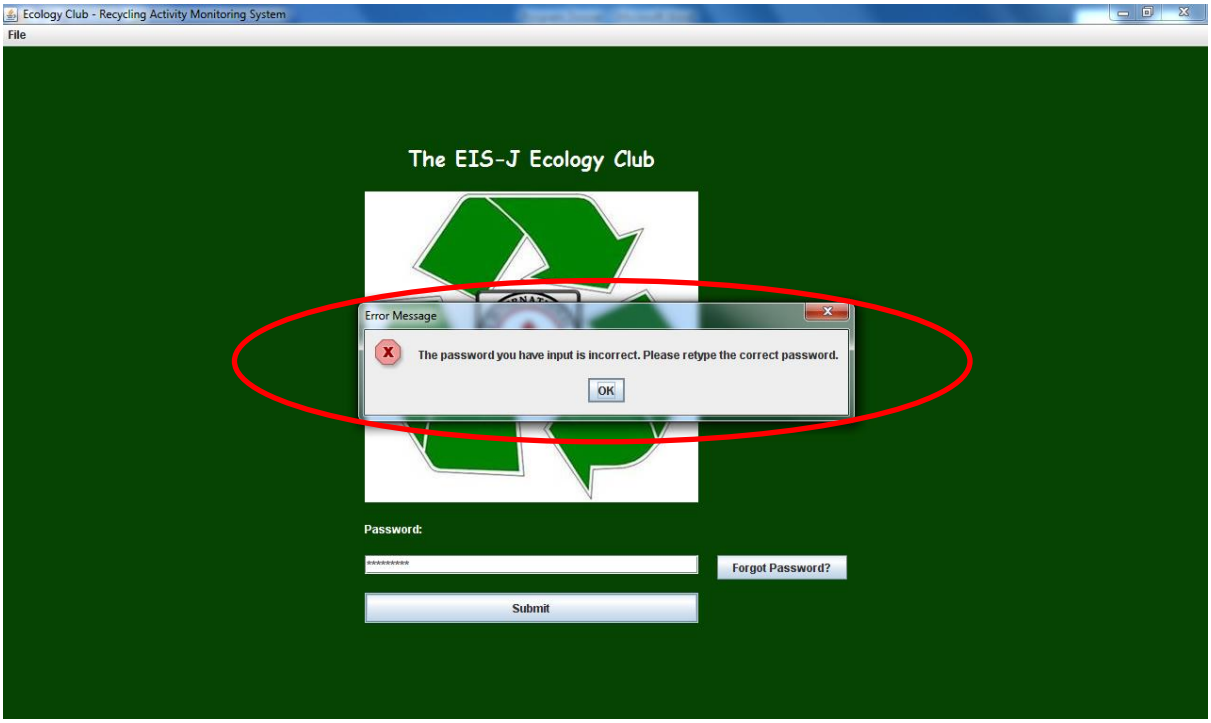
The next screenshot shows the screen that allows the user to change the password.



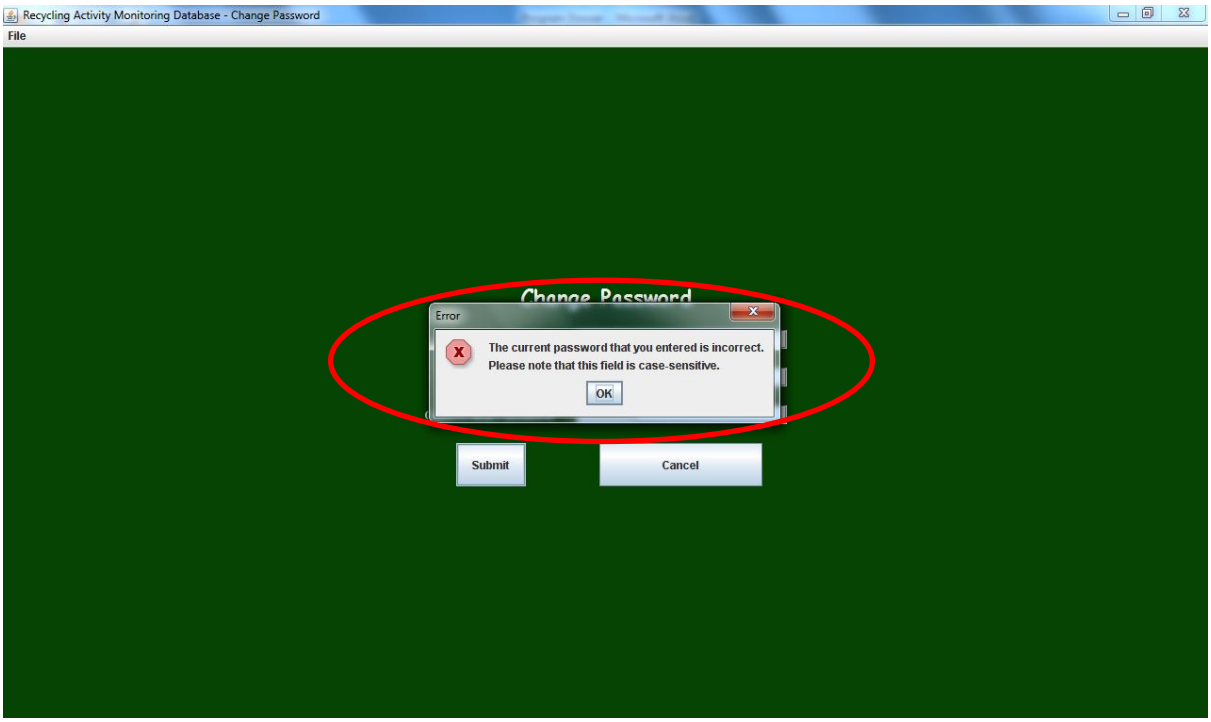
If valid data is provided i.e. the current password is authorised and the new password is input twice identically, the screen on the next page will display.



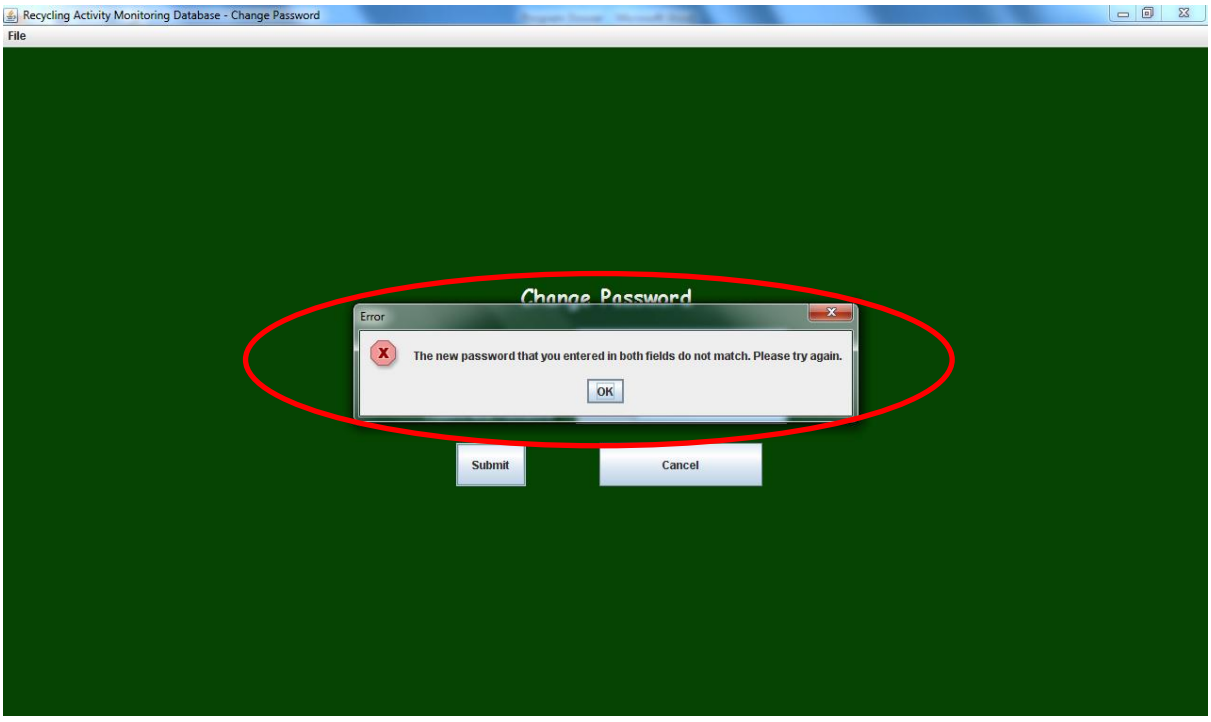
The password on the screen above has been changed from 'waterfall' to 'mcooper'. When the user attempts to input 'waterfall' during login from now. The following screen appears rather than authorisation.



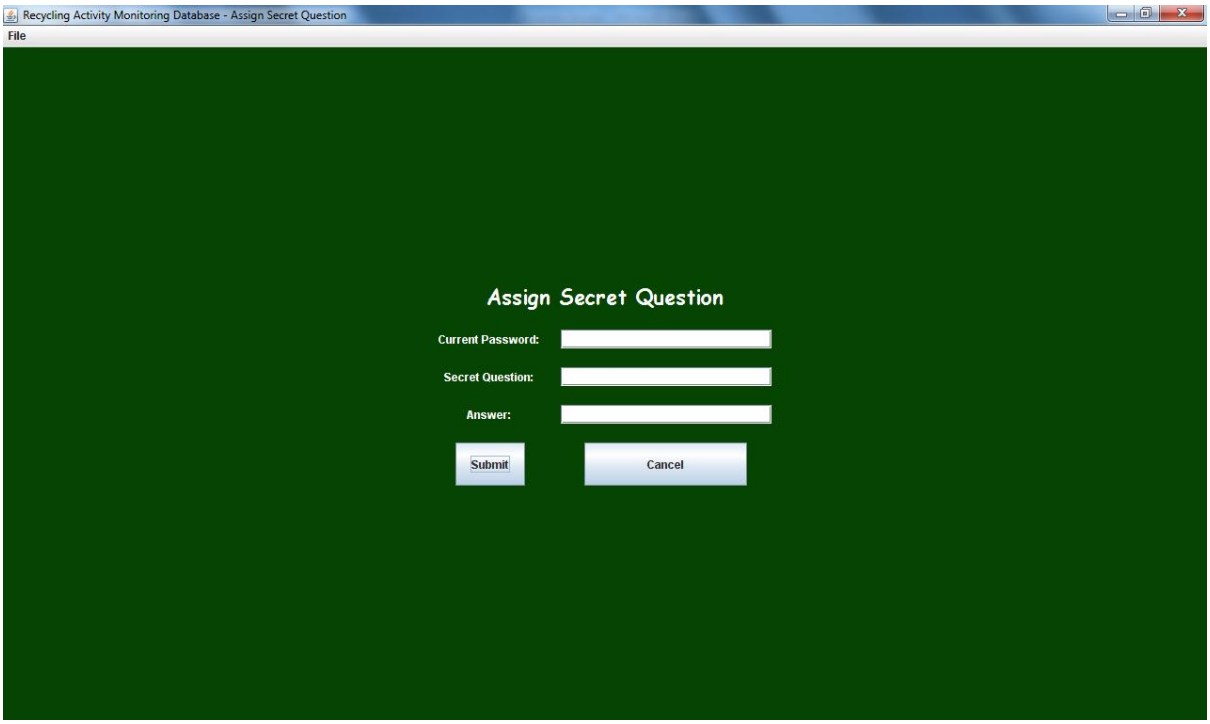
If on the Change Password screen the wrong current password is input for authorisation, the following message appears.



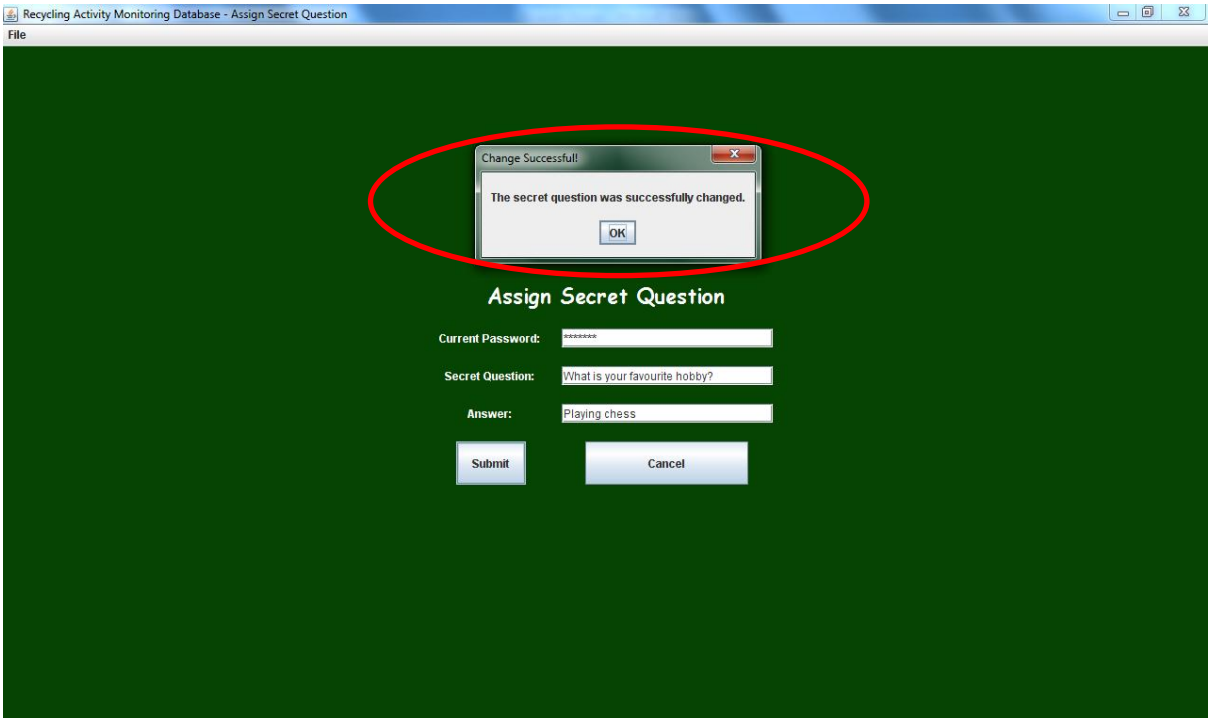
On the other hand if the new password typed in the two password fields do not match, the subsequent prompt appears.



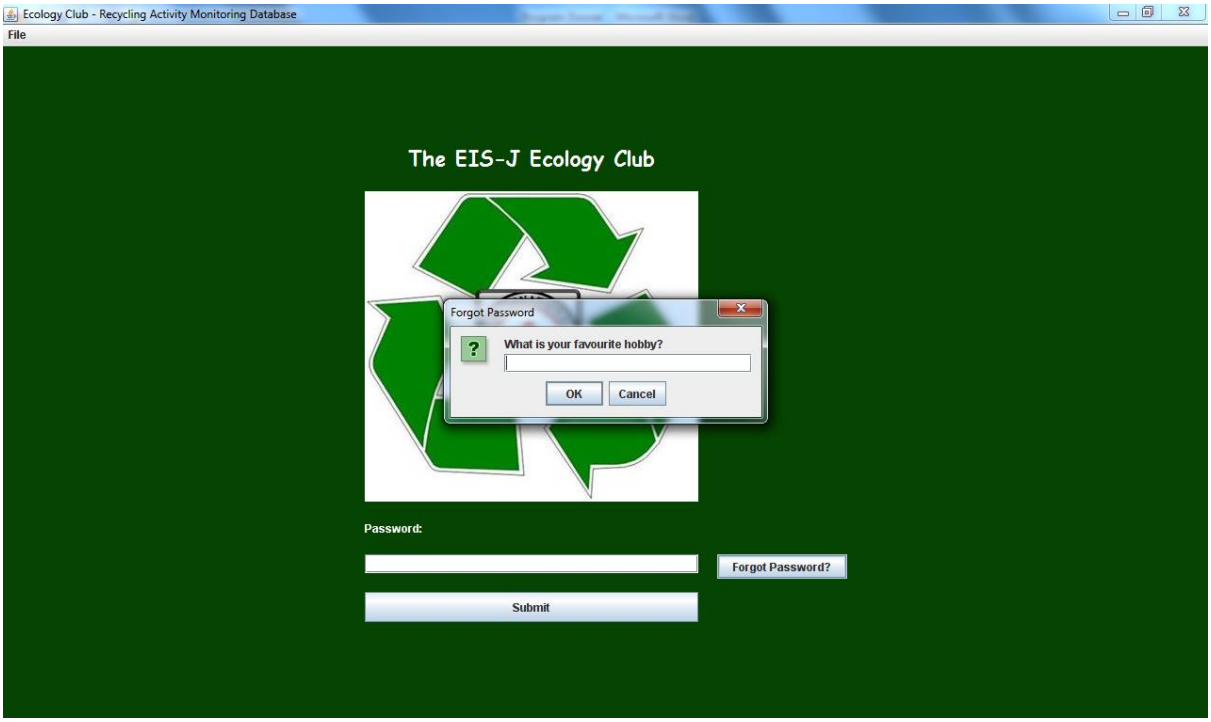
The screenshot below presents the 'Assign Secret Question' screen which allows the user to assign a secret question and its answer of their choosing.



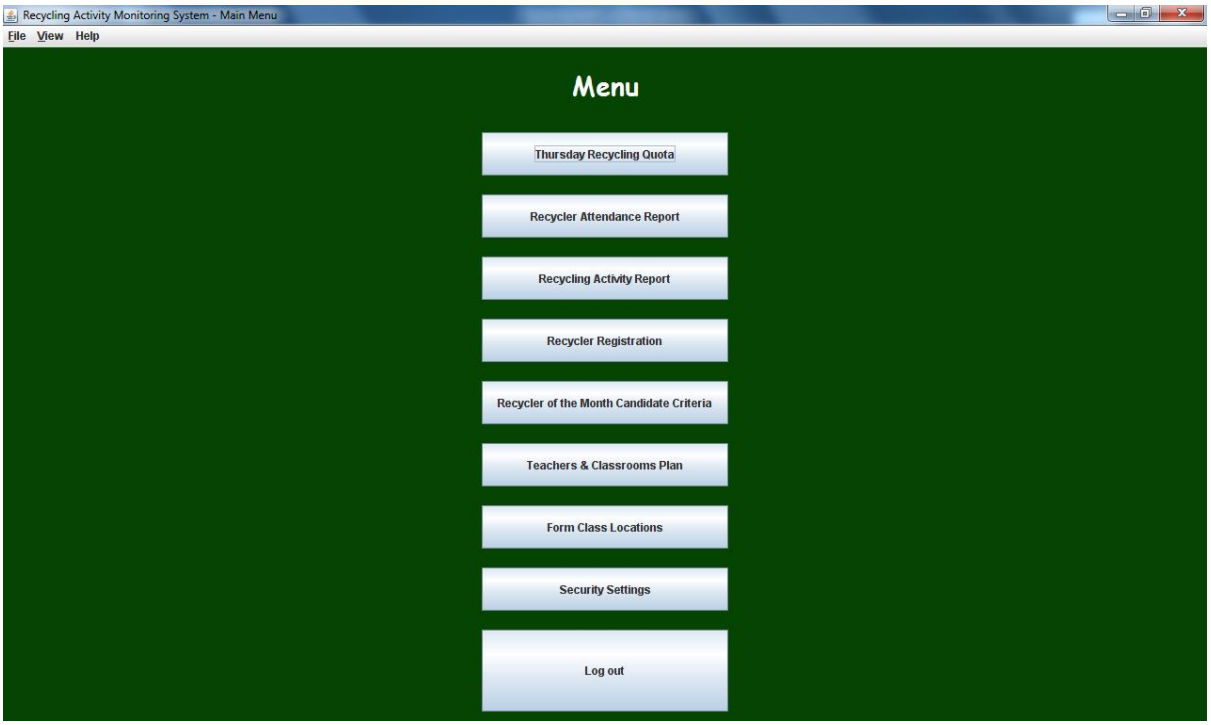
The screenshot displays the result when a new secret question is assigned and appropriate authorisation given.



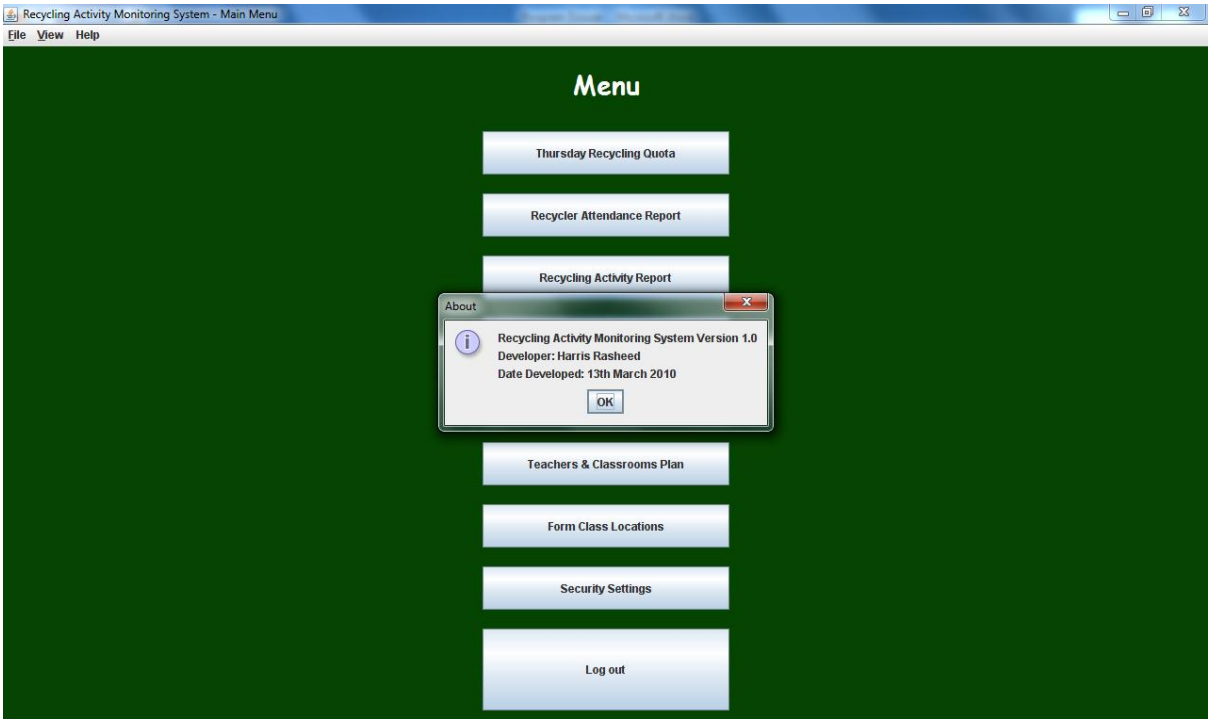
This next screenshot shows that the secret question has changed to the user assigned question.



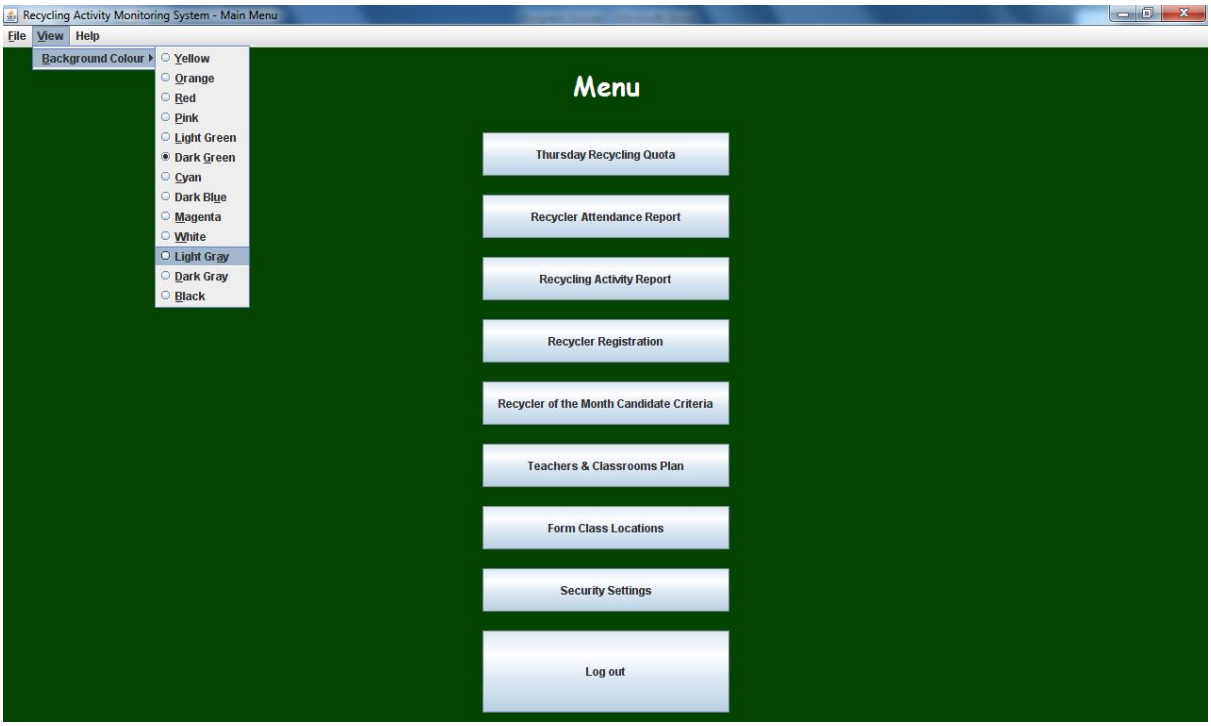
When 'Playing chess' is entered, the user is logged onto the system and is shown the menu below.



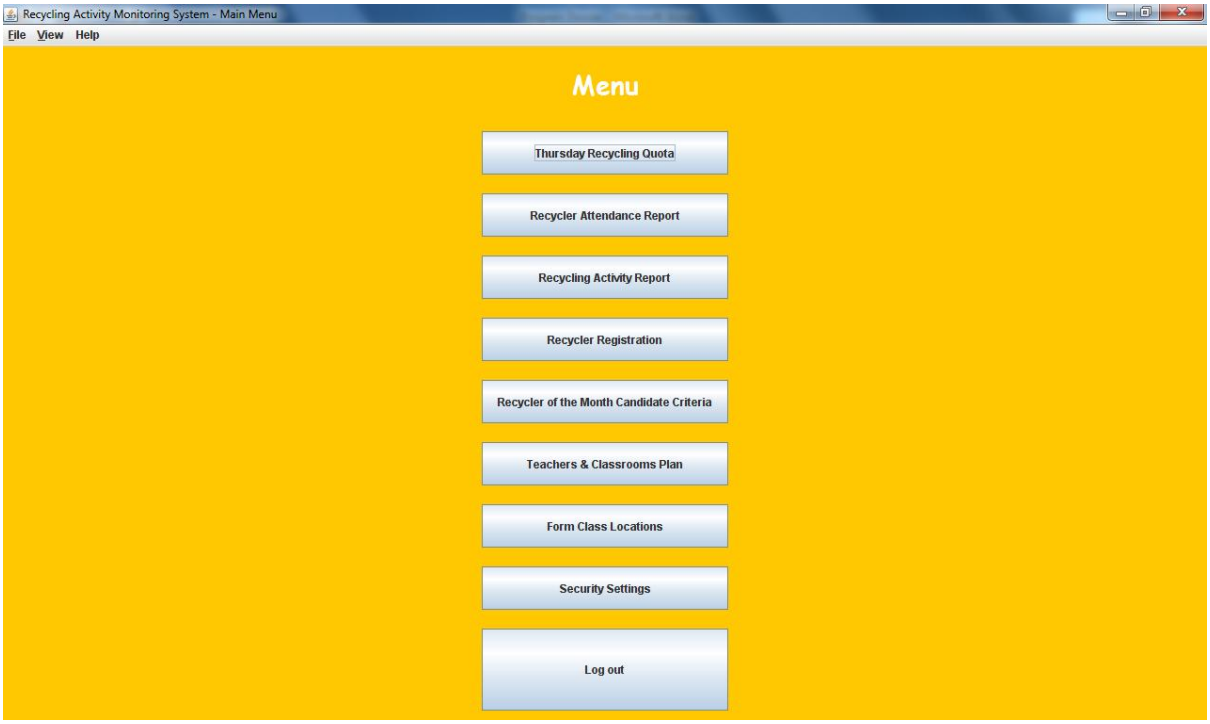
The 'Help' menu shown here displays the following dialog box.



Another user-friendly feature that I have added is that the user can select the background colour they wish to use the program in.



One such example is as follows for Orange.



Criterion D2: Evaluating Solutions

Judging the Perks and Criticising the Problems

Having completed the program and testing it for bugs, I will reflect and evaluate the success of my program.

Did it work?

Yes. The program worked exactly as planned if not better. I was able to add much more colour to the system than previously attended which Mrs. Cooper took very positively. She was very impressed by the program and believes that the system is much more efficient now that the computer handles all the processes and tallies the recycler attendance and the recycling activity. The random access file algorithms were difficult to understand at first but in the end, I was able to perfect them.

Did it address the criteria for success?

Yes. All the criterion for success were addressed. However one of the objectives, "keep a record of all the recycling activity statistics that have been processed" was met only to a certain extent in my opinion. Reviewing the program again with this in light, this element of the program is useless if Mrs. Cooper does not understand the contents of the log in the text file because of the set record lengths. It might have been better to create a report for this objective too so that Mrs. Cooper can read and interpret the data more easily. Please see Page 12 for the system objectives outlined in Criterion A2.

Did it work for some data sets, but not others?

No. This program worked for all data sets because the coding was fool-proof and allowed all forms of valid input.

Does the program in its current form have any limitations?

The algorithms written in the program could have been made more efficient. Some variables were declared global when they could have been declared locally. This would have reduced the system's memory usage and perhaps have made the program faster. In addition, there was repetition of many methods which could have easily have been accessed from other classes but the structure used prevents it for the majority of methods. This is because the GUI component and layout manager features were placed in the constructors of each class. Accessing each class then required the execution of the constructors before the desired method could be accessed. Future modifications would be to move the GUI components' layout manager code into a separate method called createAndShowGUI() rather than in a constructor. This would also make the program more efficient. The length of records in the RecyclerAttendanceStats.txt was very long. The name field was allotted 30 characters but the longest name existing in the system only takes up 16 characters.

What additional features could the program have?

The program could have had a reset button for each table to clear the contents in case the user needed to input recycling statistics from more than one week's worth of data. Also, a "New Academic Year" button would have been very useful to reset all the data in the system. I considered

this idea at the programming stage but I thought that if an unauthorised user found the system left open and logged in, they might tamper with the data or worse, clear it all using this function. Therefore, I omitted this function as Mrs. Cooper was more worried about the Recycler Attendance Report and the Recycling Activity Report. Also, a function to edit recycler details would have been more useful than the current system which only allows the user to delete them and then add them again. This can cause problems if the user misspells a recycler name during the registration process.

Was the initial design appropriate?

Yes. The initial design was appropriate however a few changes were made to most of the screens to make it more user-friendly and colourful. The overall design greatly aided the coding process as the prototypes were on paper and simply had to be digitalised on the computer with Java. The prototype design stage was instrumental in the production process of the system as the Mrs. Cooper's feedback was very useful in producing and satisfying the Eco Club Recycling Activity Monitoring System's new needs.

Was the program user-friendly?

Mrs. Cooper said "Good job with the new database. I have to admit that it is an excellent improvement over the Access one. Having reviewed your prototypes and see this masterpiece, you never fail to impress me. You are a star. Great work".

Was the program efficient?

The program was efficient with certain processes but inefficient with others. The quick sort function was utilised to sort the reports in descending order. This algorithm has a BigO Notation of $O(n\log_2 n)$ which makes it very efficient as the size of the list increases. This is likely to happen in the Eco Club Recycling Activity Monitoring System as a large number of recyclers register and participate throughout the academic year. Alternatively, a bubble sort was used to sort the Recycling Activity Report because a quick sort would be difficult to code with it. The array in this report had null values which could not be compared in a quick sort but it was easily filtered out in the bubble sort. A linear search was used throughout the program. In order to find data stored in the random access files, each record had to be read one by one before the relevant record was found. This gave the program BigO efficiency of O(n) which is not very efficient. Nevertheless, the size of these lists increase slowly so the time taken for a record to be searched will be relatively low. A small lag occurs with the program whenever the print function is initiated however, this is a system issue.

Conclusion

This program dossier has been a very interesting system analysis experience for me. The Eco Club system has needed a change for more than a year and now it has an efficient system. The majority of functions used in this program had to be researched online and was implemented into my program with ease. This program will be used by Mrs. Cooper for some time until bugs are found and the system cannot satisfy the changing needs of the EIS-J Ecology Club.

Note: This program dossier is based on a real-life situation and the program is now in actual implementation.

Mastery Aspect Index

The following pages outline how this program dossier has fulfilled a mastery factor of 1.0.

Standard Level	
1. Arrays	Arrays have been used throughout the program. In order to process the recycler attendance and store the data in the random access file, the data was passed as an array through a parameter. In addition, 1D arrays have been used to store column names of JTables. *Lines 177, 764, 1158, 1237, 1282, 1348, 1438, 1493*
2. User-defined objects	User-defined objects have been used in order to access methods of other classes and to link screens together when the user chooses to see a selected module. *Lines 564, 571, 578, 585, 592, 599, 606, 613, 625, 716*
3. Objects as data records	Not used.
4. Simple selection (if-else)	This has been used to test conditions such as if the password input is correct to the system set string. Also, the actionPerformed method makes extensive use of this structure. *Lines 3364-3408, 256-276, 2598-2610*
5. Complex selection (nested if, if with multiple conditions or switch)	A switch case has been written to convert the system date's month substring to the name of the month. Multiple conditions have been set to analyse the contents of a string so that the program can process input data without error. The currentPassword() method makes use of a nested if to test various conditions. *Lines 496-632, 2492-2542, 3591-3616*
6. Loops	For loops have been used to write records to random access file. Before they can be written, each field in a record must have a set field size. A routine for loop checks and processes each field so that the field size is set to the length it is meant to be. *Lines 151-155, 840-844, 868-921, 808-812, 1095-1101*
7. Nested loops	Nested loops have been used to process data in 2D arrays as well as to process data in random access files. *Lines 960-1012, 1043-1061, 1086-1092, 1247-1261*
8. User-defined methods	There has been extensive use of user-defined methods throughout the program. This includes referenceFormTable, changePassword, addRecyclers, storeRecyStats etc. *Lines 836-846, 859-941, 3353-3416, 1237-1216*
9. User-defined methods with parameters (the parameters have to be useful and used within the method body)	Methods such as referenceFormTable in class morningMonitor have String[][] return types. *Lines 1282-1342, 1348-1388, 1599-1653, 1815-1867*

10. User-defined methods with appropriate return values	Methods such as referenceFormTable in class morningMonitor have String[][] return types. *Lines 1815-1867, 2462-2561, 2625-2679, 2682-2703*
11. Sorting	Before reports are presented in the system, they are quick sorted or bubble sorted by attendance tally or points. *Lines 1599-1653, 1871-1898*
12. Searching	Search algorithms are implemented with random access files; linear search. *Lines 859-941, 1282-1342, 1815-1867*
13. File i/o	The use of a random access file in the program supports this. *Lines 953, 1036, 1241, 1287, 1353, 1556, 1774, 1821*
14. Use of additional libraries (such as utilities and graphical libraries not included in appendix Java Examination Tool Subsets)	Graphical libraries have been imported for GUI and swing objects and the utility library has been imported to access the system date to place a default time stamp on input screen data. *Lines 11-16*
15. Use of sentinels or flags	Flags were used for loops so that the loop only terminated when the record was found. *Lines 916, 1003, 2598*

Total Standard Level Mastery Aspects Achieved: 14/15

Note: The table above only demonstrates the Standard Level Mastery aspects that were implemented in the program. Please see the following table for documentation of Higher Level Mastery aspects.

Higher Level	
1. Adding data to an instance of the RandomAccessFile class by direct manipulation of the file pointer using the seek method	Recycling statistics and recycler attendance input into the system will be appended to the logs (random access files). *Lines 2127, 1059, 1373*
2. Deleting data from an instance of the RandomAccessFile class by direct manipulation of the file pointer using the seek method. (Data primitives or objects may be shuffled or marked as deleted by use of a flag field. Therefore files may be ordered or unordered).	The Recycler Registration screen allows the user to select an existing and previously registered recycler and to delete their name from the system. This will be useful for Mrs. Cooper at the end of each academic year so that a new list can be used. *Lines 983, 2174*
3. Searching for specified data in a file	In order to amend data in the random access files, each record had to be found first. *Lines 961-1002, 1315-1327, 976-1001*
4. Recursion	The quick sort function was used which uses the concept of recursion to function. *Lines 1599-1653*
5. Merging two or more sorted data structures	Not used.
6. Polymorphism	Not used.

7. Inheritance	Extending to JFrame and implementing the ActionListener are forms of inheritance that allow access to features of an in-built package. *Lines 19, 287, 637, 746, 1140, 1478, 1695*
8. Encapsulation	This has been implemented on the methods throughout the program in order to protect the code from outside access. Methods have been declared private and protected rather than public. *Lines 133, 171, 836, 859, 1282, 1348, 1552*
9. Parsing a text file or other data stream	Numerical data read from random access files and from text fields have been parsed in order to perform calculations and processes on it; *Lines 974, 991, 1606, 1610, 1616, 2490, 2609*
10. Implementing a hierarchical composite data structure. A composite data structure in this definition is a class implementing a record style data structure. A hierarchical composite data structure is one that contains more than one element and at least one of the elements is a composite data structure. Examples are, an array or linked list of records, a record that has one field that is another record, or an array.	An array of records have been used. In order to process data stored in JTable, the records must be stored in a 2D array. *Lines 948, 2625, 2682, 2829, 2865, 2927, 3062*
11. The use of any five standard level mastery factors – this can be applied only once	Please refer to the table on the previous page.
12. Up to four aspects can be awarded for the implementation of abstract data types (ADTs) according to the table entitled "Implementation of ADTs".	Not used.
	Not used.
	Not used.
	Not used.
16. Use of additional libraries (such as utilities and graphical libraries not included in appendix Java Examination Tool Subsets)	Please refer to SL Mastery Aspect 14 in the Standard Level Mastery aspects table on the previous page.
17. Inserting data into an ordered sequential file without reading the entire file into RAM.	Not used.
18. Deleting data from a sequential file without reading the entire file into RAM.	Not used.
19. Arrays of two or more dimensions.	2D arrays have been used to store multiple records. Evidence of this can be found in the program. *Lines 1031, 1083, 1398, 1552, 1772, 2576*

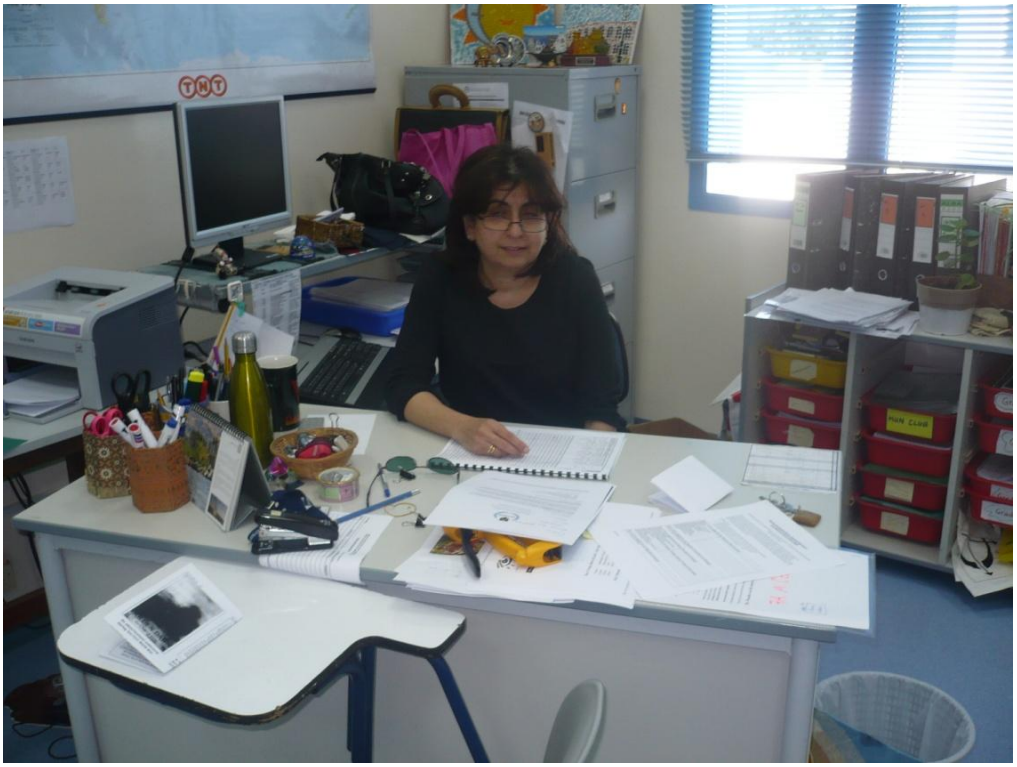
Total Higher Level Mastery Aspects Achieved: 11/19 - Mastery Factor 1.0

Appendix

The following screenshot shows the EIS-J's Eco Club website. It was created on Piczo, a free web host. The cost for hosting the website is free because advertisement is included on each webpage. This is not optional but after the website has crossed the milestone of 5000 visitors, adverts will be removed by the host as a token of achievement of attracting so many web surfers to their website.

EIS – Jumeirah Ecology Club Website: http://www.eisecoclub.piczo.com/

The picture below shows Mrs. Cooper, the EIS-J Ecology Club Co-ordinator at her desk.



The following screenshot displays the amount of storage space that the Recycling Activity Monitoring system takes up on a computer hard disk.