**(B) Definition, Investigation and Analysis**

1. **Definition – Nature of the Problem**

**Description of Organisation**

Beserah Primary School is an average elementary school which provides essential education to young children to prepare them for the challenges in life. It is situated an area named Beserah in Kuantan town, giving its name. This school was founded by in 1906. Despite being a primary school of average reputation, this school is fighting its way in creating its excellence in all aspects. One of the efforts made is the establishment of a variety of co-curricular activity clubs and societies to allow the students to choose to join and participate in activities of their interests.

I have decided to choose the English Language Society of this school as my project because Miss Narulhuda binti Yaacob is an acquaintance of my mother and had enquired whether I was interested in solving the problem faced by the club as my project. There are many co-curricular clubs and societies available in Beserah Primary School such as Science and Mathematics Society, Badminton Club, Hockey Club, Soccer Club, Astronomy Club, Drama Club, Choir Club etc. For this project, I have decided to computerise the club’s management system in order to ease the jobs of taking attendance and handling the member and event information.

The English Language Society of Beserah Primary School was formed when this school first started its operation. The primary objective of the constitution of this club is to hone the level of English language of the students. Membership is open to all students. The current chief advisor of this society is Miss Narulhuda binti Yaacob, who is also the Head of English Panel of this school. Mohammad Danny Hafiz bin Noor Hazly and Mohammad Amer Aimam bin Baderol Hisham, both from the class 6B, were elected as the chairperson and the vice chairperson of the society respectively. The secretary of the society is Nur Amira Aqilah binti Ibrahim from the class 6A while the treasurer is Hanis Daniela binti Hasbullah from the class 6C.

**Description of Problem**

The main problem faced by the advisors of the English Language Society contemporarily is the inefficiency of the current manual system. Whenever details of a particular member are required, the member registration form of that member must be searched. This takes up a considerable amount of time. The simple solution to this problem without changing its implementation method is the copy of member details in another paper which ease the searching of data but causes the data to be duplicated and redundant. Furthermore, backup process takes up a large amount of time as data need to be copied manually. Thus, data may be lost before the backup process is complete, causing data stored to be incomplete which may cause the system to break down.

Next, the storage of data files is another major problem faced. As all the data need to be filed and stored separately, the data files need to be labeled carefully. Any mistake may result in the “lost” of data. The amount of physical space took up by the data files is also a problem which can be solved.

In addition, the illegible handwritings of either the members or the advisors can cause failure in readability. This may cause difficulties in cases where data need to be referred and when the guardian of the member needs to be contacted. Also, re-filling of member form is difficult when the advisors have no idea where to find the member to allow him or her to perform the mentioned task.

**Methods Currently in Use**

The English Language Society of Beserah Primary School is currently using a manual system for members, events and attendance management.

For the management of members, when a student wishes to join the club, he or she will be given a member registration form to fill up. After completing the form, he or she will submit it to his or her English language teacher to check whether the data filled up is valid. After that, the teacher will pass the completed form to any advisor of the society to be kept in the Member file.

On the other hand, for the management of events, after an event has been held, its details are recorded on an event form, which is then filed in the Event file. The attendance of the members to an event is taken on another form, which is the member attendance form. All the member attendance forms are filed in the Attendance file.

The members and events each have unique IDs to allow easier and faster reference to look back the attendance of members in each event.

**Origin and Form of Data**

Member Registration Form

All the data in the table below is filled up, handwritten, by the students and checked by their respective English language teachers. Any changes in the content of data which is filled up are made by either the teacher or the student.

A copy of the member registration form is included in page 5 after the tables of contents describing the fields to be filled up in the form.

|  |  |
| --- | --- |
| **Name** | **Description** |
| Member ID | A five digit number, which is also the student ID of the student to be registered as a member of the society, used to distinguish every student or member. |
| Member Name | The name of the student to be registered as a member of the society. |
| Member Class | The class of the student to be registered as a member of the society. |
| Position | The position held by the student to be registered as a member of the society in the society. |
| Highest English Grade | The highest grade achieved by the student to be registered as a member of the society in English language.  If the student do not have any record of English grade in this school then tick “F” in this field. |
| Potential Contestant | The indication whether the student to be registered as a member of the society is eligible to be a contestant of competitions or quizzes on English language. |
| Guardian | The name of the parent or guardian of the student to be registered as a member of the society. |
| Contact Number | The reachable contact number of the parent or guardian of the student to be registered as a member of the society.  Mobile phone number is preferable. |
| Signature | The signature of the student to be registered as a member of the society indicating he or she agrees to join the society on his or her own will and promises to obey all its rules. |
| Date | The date of the day when the student registered to be a member of the society. |



Sekolah Kebangsaan Beserah

English Language Society

Member registration form

Member ID :

*(Student ID)*

Member Name :

Member Class :

Position :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| English Grade | A | B | C | D | E | F |
| Highest English Grade |  |  |  |  |  |  |
| Potential Contestant | **/** | **/** |  |  |  |  |

*Note: - Please tick in the appropriate box for Highest English Grade.*

Guardian :

*(Name)*

Contact Number :

Hereby, I agree to join this society and will obey all of its rules and regulations as an official member.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_

Event Form

A copy of the event form is included in page 7 after the tables of contents describing the fields to be filled up in the form.

|  |  |
| --- | --- |
| **Name** | **Description** |
| Event ID | A four digit number given to the event in ascending order which is used to distinguish every event. |
| Event Name | The name of the event, including the year if it is held annually or in a specific time period. |
| Event Venue | The venue where the event was held.  If the venue is not within the school compound then this field should be filled up as detailed as possible. |
| Event Date | The date when the event was held. |
| Event Level | The level of the event held. |
| Event Chief Advisor | The name of the chief advisor of the event held. |
| Event Advisor | The name of the advisor of the event held who assisted the chief advisor. |
| Remarks | The additional information regarding the event held. |
| Signature | The signature of any advisor of the event held indicating he or she was watching over the members when the event was held. |

Attendance Form

A copy of the event form is included in page 8 after the tables of contents describing the fields to be filled up in the form.

|  |  |
| --- | --- |
| **Name** | **Description** |
| Number | The ascending number of the members in the society. |
| Member Name | The name of the member of the society. |
| Member ID | The five digit number which is the ID of the member of the society, which is also the student ID of the member, used to distinguish every student or member. |
| Event ID | The four digit number which is the ID of the event that was held which is used to distinguish every event. |
| Class | The class of the member of the society. |
| Date | The date when the attendance of the members of the society was taken. |
| Attendance | The attendance of the members of the society.  Tick if the member was present, fill “0” if the member was absent. |
| Signature of the advisor in charge | The signature of the advisor of the event held who took the attendance of the members for the event. |



Sekolah Kebangsaan Beserah

English Language Society

Event form

Event ID :

Event Name :

Event Venue :

Event Date :

Event Level :

Event Chief Advisor :

Event Advisor :

Remarks: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Note: - Please mark \* beside the party who signed.*



Sekolah Kebangsaan Beserah

English Language Society

Member Attendance form

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Member Name  (Member ID) | Event ID | Attendance | | | | | | | |
|  |  |  |  |  |  |  |  |
| Class | 4  /  2 | 11 /  2 | 18 /  2 | 25 /  2 | 4  /  3 | 11/  3 | 18/  3 | 25  /  3 |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |  |  |  |  |
| 19 |  |  |  |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |  |  |  |
| Signature of the Advisor In Charge | | |  |  |  |  |  |  |  |  |

1. **Investigation and Analysis**

**Plan of Information Gathering**

In order to obtain a clearer view of the problem faced by this organisation, I set up a two day schedule to investigate and understand the management of members, events and attendance more thoroughly.

The first method of information gathering that I planned to use was by conducting an interview with Miss Narulhuda binti Yaacob after acquiring her schedule and fixing the date and time when both she and I were free. I chose her as my interviewee because she was holding the highest position in the English Language Society and was most likely the one who understands the operation and imperfections of the current system the most. She may also suggest ways to improve the current system. Besides that, she will also be one of the end users of the system. Therefore, it would be best for her to know and communicate on the development of the system.

As there are many aspects of the new system that I would like to communicate and discuss on, I planned to conduct two interviews with Miss Narulhuda. The first one will be a discussion on the problems, storage, backup and data processing whereas the second interview will be a discussion on the details including the input and output information, requirements and possible solutions.

Furthermore, I planned to request Miss Narulhuda binti Yaacob to allow me to observe and then understand how the current manual system handles all the different data after the first interview. This will help me grasp the operation of the current manual system. Moreover, I may also encounter and discover problems which were not mentioned or stated before.

By integrating these methods and combining all the information gathered, I will be able to identify the exact problems and flaws in the current management system of the English Language Society. Thus, I can work on counteracting these problems in the solution. Additionally, a list of problems which are to be solved, the goals of the new system and the technical specifications can be summarised, giving ease for me to produce a solution which can satisfy the requirement and preference of Miss Narulhuda.

**Interview 1**

I conducted my first interview with Miss Narulhuda binti Yaacob on the 16th of July 2015 at 9 a.m. The aim of the first interview was to see how the current system works, to understand more about the current system, to identify the problems and difficulties in the current system and to produce a rough guideline on how the new system should work.

Below is a script of the mentioned interview.

Me : *Can you describe the process of member registration*

*briefly?*

Ms Narulhuda : *The student interested in joining the society needs to come*

*to any advisor of the society, then we will hand him or her*

*a member registration form to be filled up. After that, he or*

*she needs to submit the completed form to his or her*

*English teacher to be checked. If the information filled up*

*is valid and logical, then the teacher will hand it to us for*

*us to keep it in our Member file.*

Me : *How do you differentiate the members who have graduated*

*or transferred from this school with those that are still here?*

Ms Narulhuda : *If the member has graduated or transferred, we move their*

*record to a separate Past Member file. We archive their*

*records in case they come back in search of their*

*co-curriculum details.*

Me : *What about the events? How do you record them?*

Ms Narulhuda : *The event form is completed by the event’s chief advisor*

*then it is kept in the Event file.*

Me : *I noticed that most schools conduct co-curricular*

*activities on a weekly basis. Is it the same for your society?*

Ms Narulhuda : *Yes, our society’s activities are also run weekly. In fact, the*

*time for all society activities are fixed, which is from 3p.m.*

*to 5p.m. That is why the “Time” field is not available in the*

*event form. For activities like quizzes, competitions and*

*talks which are held at times which different from the usual*

*society activities, the time is stated under “Remarks”.*

Me : *I see, actually that was what I intended to ask next. I will*

*move on to the attendance part then. How is the member*

*attendance taken?*

Ms Narulhuda : *The member attendance form was designed for that*

*purpose. During every event, an advisor takes the*

*attendance of members. Usually the event advisor besides*

*the chief advisor who is free will be taking the attendance.*

*That advisor will be in charged to solve the issues*

*regarding the attendance of the members for that*

*particular event if they appear.*

Me : *How is the advisor in-charged identified?*

Ms Narulhuda : *The advisor will be identified through his or her signature*

*on the member attendance form.*

Me : *Where are all the files stored?*

Ms Narulhuda : *They are all stored in the English Language Society room*

*which is locked at all time. Only the chief advisor of the*

*society and the school principal own the key to the room.*

Me : *So the data files are relatively safe then. Have you met any*

*difficulty with the current system?*

Ms Narulhuda : *When a member requests to edit his or her details, we need*

*to flip through Member file to find the particular member’s*

*record so that it can be updated.*

Me : *That is very inconvenient for you all. How is the backup*

*done for the current system?*

Ms Narulhuda : *The advisors need to copy all the data of the members,*

*events and attendance. This is only done by the advisors of*

*the society to prevent anyone else from changing the data*

*without our consciences.*

Me : *Is there any difficulty faced during the backup process?*

Ms Narulhuda : *Yes, the illegible handwritings in the member and event*

*records affect the readability of the details. Therefore, we*

*chose to make a copy of the member or event record other*

*than the backup copy to replace the original record if we*

*find the original one difficult to read. However, this can be*

*a difficult task for us if we cannot find the member due to*

*his or her inactiveness in the society activities since his or*

*her class may not be identified due to the problem*

*mentioned.*

Me : *Based on the problems and difficulties mentioned*

*regarding the current system, what do you want the new*

*system to improve on in order to minimise these problems?*

Ms Narulhuda : *I would like to have a system which allows me to register,*

*view, edit and delete the member and event details. It*

*should also allow attendance of members for events to be*

*taken. Most importantly, it should be easy to use.*

Me : *I think that is sufficient for today. Thank you for sharing.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

**Direct Observation**

The observation was carried out on the 16th of July 2015 as planned. Miss Narulhuda gave me a demonstration and a detailed explanation on how the registration of member is usually carried out at 2.30 p.m. before the society activity starts.

Around 3.30 p.m., I witnessed Miss Sharifah Nor Mala binti Hassan, the event advisor in charged, take the attendance of the members first-hand. After the event ended and all the members have left the event venue at 5.15 p.m., Miss Sharifah filled up the event form with me watching. Then, she filed the event and member attendance forms into their respective files.

I believe that the amount of members of the society is the main reason large problem has yet to arise. From my observation, I noticed that the current manual system is running quite smoothly despite the problems that are noticeable:

1. Some of the member registration forms and event forms are messy because corrections have been made either by crossing the errors out or by using correction fluid or tape. Furthermore, bad handwritings also exist in some of the forms. This makes the writings on the form difficult to read and the contents difficult to interpret. Hence, some members are required to fill up the member registration form again, the same goes for the event form.
2. Searching of data is quite fast because the number of members who joined the society is relatively small. However, if many members joined this society, the searching time will increase by a great amount and the inefficiency of the current system will be a threat for the society to continue its operation.
3. The advisors spend a considerable amount of time to check the written contents of all the forms again and again. This can be overcome by getting more people to check the same set of data as it is unlikely that everyone oversees the mistake in the data.

**Interview 2**

On the 18th of July at 9 a.m., I conducted a follow-up interview with Miss Narulhuda after observing the operation of the current system and summarising the first interview’s points. In this interview, the proposed new system was discussed in detail.

Below is the script of this follow-up interview.

Me : *After observing the operation of the current system and*

*based on the information gathered in the last interview, I*

*noticed that the new system should be able to produce a*

*number of inputs which are the member records, event*

*records and attendance records. Is there any other data to*

*be inputted?*

Ms Narulhuda : *Yes, the committee board of the year should be inputted*

*and saved too. Additionally, I would like the attendance to*

*be taken based on a certain event so that it is easier to read*

*off.*

Me : *I see. What data should be outputted from the system?*

Ms Narulhuda : *If possible, it should be able to list out all the potential*

*contestants and the members who were present in a certain*

*event. The committee board of the year should also be*

*generated as an output when required so that a printed*

*version of the committee board can be obtained easily.*

Me : *Based on all your proposals, there are a few methods to*

*overcome the problems faced by the current system. The*

*first method is to distribute the responsibilities between the*

*advisors and use a word processor software. For example,*

*advisor A will be handling filing, advisors B and C will be*

*backing up the member and event records, advisor D will*

*be listing out the potential contestants, members present in*

*a certain event and committee members while advisor E is*

*the chief advisor. All the data of all the member, event and*

*attendance records will be saved in a word processor*

*software. For this method, the drawback is the jobs may be*

*disorientated when the advisor in charged is away, and*

*human error and illegible handwritings may still occur in*

*the paper works. Moreover, the filtering of data still needs*

*to be done manually. The second method is by using a*

*spreadsheet software linked to a database. This allows data*

*to be stored neatly in the spreadsheet software by utilising*

*its build-in functions so that they can be read off easily. The*

*database will be used to search and query data. However,*

*it will include extra functions which are not needed as the*

*software used are off-the-shelf software which are readily*

*available in the market and not targeted to solve your*

*problem. The final method is by using a programme coded*

*from Visual Basic 2012 connected to Access. The*

*programme will be designed by me so I will create it based*

*on your requirements and proposals solely without*

*inserting extra functions so that it will be relatively easier*

*to use. However, you will need to learn how to use this*

*software as it is different from other software available.*

Ms Narulhuda : *I think I would prefer to use the last option since everything*

*will be done according to my wishes. Besides, no extra*

*functions will be in the programme so it is unlikely that*

*unexpected results or interfaces will appear after we have*

*learned how to operate the programme.*

Me : *For your information, a software with the features that you*

*wish to have requires a computer with moderate*

*specifications. Does your budget allow the development of*

*this system?*

Ms Narulhuda :  *The laptop provided by the school to our society is quite*

*new so I believe it is powerful enough to run the software.*

*Yes, I believe our budget will be enough to cover the*

*development of the system.*

Me : *What input method would you prefer the new system to*

*implement?*

Ms Narulhuda : *The basic keying in using keyboard and selecting using*

*mouse will be sufficient. I do not wish for a flashy and*

*complicated system.*

Me : *I see. What security measures would you like to apply on*

*the new system?*

Ms Narulhuda : *Since the laptop is user-authenticated, the new system can*

*be accessed like other software directly. In fact, I would*

*prefer to have it that way, simpler and easier to use. I think*

*no special measures are necessary. The event advisor who*

*will be writing the event record will bring the laptop back*

*after the event. On other days, it will be under the*

*possession of the chief advisor of the society until the*

*co-curriculum day every week, which is Wednesday.*

Me : *How would you like to backup the data in the new system?*

Ms Narulhuda : *Flash drive is the best choice, I think. The data will be*

*copied to a few flash drives for backup purpose, then some*

*will be under the possession of the different advisors, and*

*one will be kept in the English Language Society room.*

Me : *How often do you think the backup files should be updated?*

Ms Narulhuda : *Once a week should be sufficient since the society activities*

*are held weekly and the members only edit their details*

*when necessary, which is quite rare.*

Me : *I see. I am very grateful for your time and effort spent on*

*helping me design this system. I believe that I will be able*

*to work on this project after acquiring this amount of*

*information from the discussions between us. I will try to*

*get the new system done according to your requirements*

*and proposals. Thank you.*

Ms Narulhuda : *I am looking forward to it. Good luck and all the best.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

**Analysis**

After spending two days gathering information about the current manual system through observation and two interviews, I made a summary of the requirements of the solution.

The data inputted into the system are the member and event records, which are done handwritten by the respective parties. Then, the written data are checked for their presence and validity before they are kept in their respective files. Another data inputted is the attendance of the members to the events. The attendance taking process can only occur after confirming the members who are currently in the society and the event held.

On the other hand, the data outputted are the list of members present, potential contestant list and the committee board. These outputs all require the data to be filtered out carefully and accordingly, which takes up a considerable amount of time.

The current system has its weaknesses. Firstly, it is prone to human errors as the data are all collected, checked and inputted by humans. Any error in the data may not be identified until the data is in immediate need.

Next, the current system does not support the editing of data. As all the data are handwritten on their respective forms, the editing of data can only be done either by changing the old data using a correction fluid, correction tape or by crossing out the wrong or outdated data and writing the updated data on any available space, or by filling up another form with the updated data. The former will cause the form to be messy and may cause the data to be unreadable while the latter is not environmental friendly and takes up a larger amount of space as all the data needs to be filed up for reference.

Furthermore, the amount of data requires a significant amount of physical space for storage purpose. This also causes the searching of data to be difficult and slow, especially when the data needed is labeled wrongly due to human error.

However, this manual system also has its own merits. Electronic devices are not required in this system. This causes the preparation and operation costs to be lower. It also eases the staff training process as no sophisticated machine or software is involved.

Context Diagram

Please refer to page 62.

Data Flow Diagram

Please refer to page 63.

**Alternative Solutions**

There are a number of alternative solutions which can be applied to solve the existing problems. The following are the methods that I proposed to Miss Narulhuda to overcome the difficulties faced by the society:

1. **Distribute the responsibilities and use a word processor software**

If the responsibilities are divided and distributed between the advisors and the data of all the member, event and attendance records saved in a word processor software, a half-computerised system can be established with the advisors more specialised in their respective jobs. Additionally, data can be stored in an electronic device such as laptop or computer, leading to fast and easy backup of files by making a copy of the data files in another device such as flash drive or CD. However, the jobs may be disorientated when the advisor in charged is on leave or away. Also, human error and illegible handwritings may still occur in the paper works besides the filtering of data still needs to be done manually.

1. **Using a spreadsheet software linked to a database**

All the data can be saved neatly using the spreadsheet software by utilising its build-in functions so that they can be read off easily while the database can be used to search and query data. This will be a fully-computerised solution. Paper works can be reduced to minimum using this solution but it may include extra functions which are not needed as the software used are off-the-shelf software which are not designed to solve the problem faced by this society.

1. **Using a programme coded from Visual Basic 2012 connected to Access**

By using a programme specially created for the society, all its problems can be solved. The process of saving, editing, searching and deleting is eased. Furthermore, the filtering of data can be done using special features. Additionally, all the features, interfaces and contents can be designed and created according to the client’s proposals. However, the end users will have to learn how to operate the new software as it is different from other software available.

**Proposition of New Solution**

After considering the pros and cons, I have decided to select the third alternative solution which is by using a programme coded from Visual Basic 2012 connected to Access to solve the existing problems

Since my client wishes to have a new system which is easy to use, a fully-computerised system can fulfill this proposal. Besides that, my client already has a laptop with moderate specifications, which will be powerful enough to run the programme. Furthermore, this solution is less risky as anyone who knows how to use the programme can handle the saving, updating, searching and deleting of records. The absence of an advisor will not affect the operation of the new system provided all the advisors had learnt to use the programme.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

**Requirement Specifications**

1. **Functional Requirements**

After having a discussion with my client, it has been decided that the new system must have the following functions:

1. Able to register or insert, update, delete and search for the member details
2. Able to register or insert, update, delete and search for the event details
3. Able to take attendance of members for a certain event
4. Able to search for the attendance list of members for a certain event
5. Able to insert, update and search for and generate the committee board of a certain year and print it
6. Able to generate a list of members present for a certain event and print it
7. Able to generate a potential contestant list and print it

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

1. **Input Requirements**

The following data are to be inputted into the system.

Category: **Member Details**

|  |  |
| --- | --- |
| **Field Name** | **Field Description** |
| Member ID | A five digit number, which is also the student ID of the member of the society, used to distinguish every student or member. |
| Member Name | The name of the member of the society. |
| Member Class | The class of the member of the society. |
| Position | The position held by the member of the society in the society. |
| Highest English Grade | The highest grade achieved by the member of the society in English language.  If the member do not have any record of English grade in this school then tick “F” in this field. |
| Potential Contestant | The indication whether the member of the society is eligible to be a contestant of competitions or quizzes on English language. |
| Guardian | The name of the parent or guardian of the member of the society. |
| Contact Number | The reachable contact number of the parent or guardian of the member of the society.  Mobile phone number is preferable. |
| Condition | The current condition of the member.  Condition is “Inactive” only if the member has transferred or graduated from this school. |

Category: **Event Details**

|  |  |
| --- | --- |
| **Field Name** | **Field Description** |
| Event ID | A four digit number given to the event in ascending order which is used to distinguish every event. |
| Event Name | The name of the event, including the year if it is held annually or in a specific time period. |
| Event Venue | The venue where the event was held.  If the venue is not within the school compound then this field should be filled up as detailed as possible. |
| Event Date | The date when the event was held. |
| Event Level | The level of the event held. |
| Event Chief Advisor | The name of the chief advisor of the event held. |
| Event Advisor | The name of the advisor of the event held who assisted the chief advisor. |
| Remarks | The additional information regarding the event held. |

Category: **Attendance Details**

|  |  |
| --- | --- |
| **Field Name** | **Field Description** |
| Event ID | A four digit number given to the event in ascending order which is used to distinguish every event. |
| Member ID | The five digit number which is the ID of the member of the society, which is also the student ID of the member. |
| Attendance | The indication whether the member of the society was present in the event. |

Category: **Committee Members Details**

|  |  |
| --- | --- |
| **Field Name** | **Field Description** |
| Chief Advisor | The name of the chief advisor of the society. |
| Advisor 1 | The name of one of the three advisors of the society who is in charge of assisting the chief advisor. |
| Advisor 2 | The name of one of the three advisors of the society who is in charge of assisting the chief advisor. |
| Advisor 3 | The name of one of the three advisors of the society who is in charge of assisting the chief advisor. |
| Year | The year in which the committee board is effective. |
| Chairperson | The name of the member of the society who holds the position of chairperson. |
| Vice Chairperson | The name of the member of the society who holds the position of vice chairperson. |
| Secretary | The name of the member of the society who holds the position of secretary. |
| Treasurer | The name of the member of the society who holds the position of treasurer. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

1. **Output Requirements**

The following data are required to be output by the system.

Document Name : **Potential Contestant List**

Description : A list of active members who are eligible and qualified to

be a contestant of competitions or quizzes on English

language.

|  |  |
| --- | --- |
| **Field Name** | **Field Description** |
| Highest English Grade | The highest grade achieved by the member of the society in English language.  If the member do not have any record of English grade in this school then tick “F” in this field. |
| Member ID | A five digit number, which is also the student ID of the member of the society, used to distinguish every student or member. |
| Member Name | The name of the member of the society. |

Document Name : **List of Members Present**

Description : A list of members who were present in a certain event.

|  |  |
| --- | --- |
| **Field Name** | **Field Description** |
| Member ID | A five digit number, which is also the student ID of the member of the society, used to distinguish every student or member. |
| Member Name | The name of the member of the society. |
| Event ID | A four digit number given to the event in ascending order which is used to distinguish every event. |
| Event Name | The name of the event, including the year if it is held annually or in a specific time period. |
| Event Venue | The venue where the event was held.  If the venue is not within the school compound then this field should be filled up as detailed as possible. |
| Event Date | The date when the event was held. |

Document Name : **Committee Board**

Description : A list of committee members of a particular year.

|  |  |
| --- | --- |
| **Field Name** | **Field Description** |
| Chief Advisor | The name of the chief advisor of the society. |
| Advisor 1 | The name of one of the three advisors of the society who is in charge of assisting the chief advisor. |
| Advisor 2 | The name of one of the three advisors of the society who is in charge of assisting the chief advisor. |
| Advisor 3 | The name of one of the three advisors of the society who is in charge of assisting the chief advisor. |
| Year | The year in which the committee board is effective. |
| Chairperson | The name of the member of the society who holds the position of chairperson. |
| Vice Chairperson | The name of the member of the society who holds the position of vice chairperson. |
| Secretary | The name of the member of the society who holds the position of secretary. |
| Treasurer | The name of the member of the society who holds the position of treasurer. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

1. **Software Requirements**

The following software will be required in order to develop and run this new system.

1. Visual Studio 2012

Used to create the user interface, create the programme code and debug it. It has the ability to create maintainable codes using indentation, annotation, meaningful variable names, white spaces and other techniques.

1. Access 2007

A database management system which allows a relational database to be created, updated and maintained. Microsoft Access is considered as one of the simplest database creator and has all the necessary functions for the problems to be solved. It also provides data definition language and data manipulation language which allows queries to be created to search the database.

1. Windows 7

Used as the operating system to run the new system because the laptop of the client is using a Windows 7 operating system. Therefore, it provides the environment for the software to be executed. This operating system also consists of the .NET framework v4.5 which is essential for the other software.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

1. **Hardware Requirements**

The client is going to use a laptop with build in touch pad and keyboard.

The following hardware is recommended in order to run the new system smoothly and efficiently. Hardware with lower specification may lower the performance of the new system and cause problems such as lagging.

|  |  |
| --- | --- |
| **Hardware Name** | **Justification** |
| Intel Core i5-4200U Processor with a supported motherboard | 4 GB of RAM and 2.66 GHz processor which meets the minimum hardware specification of 1 GB of RAM and 1.6 GHz for Visual Studio 2012. Thus, multitasking may be performed without lagging. |
| 4 GB USB flash drive | To save backups of the database and the software, so that data can be restored if the original one is lost or corrupted. |
| 500 GB internal hard disk | To store files of the operating system, Microsoft Office, database and other necessary software, and to provide the space for installing the necessary software. |
| Inkjet printer | To print out the document outputs produced by the software when necessary.  For instance, the committee board output is printed to post it on the English Language Society’s notice board. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

**(C) Design**

1. **Nature of the Solution**

**Objectives of the System**

After inspecting the details obtained from the interviews and the observation and analysing the needs and requirements of the client, the objectives of the new system has been set. The programme will be created to fulfill the following objectives:

1. Able to register or insert, update, delete and search for the member details
2. Able to register or insert, update, delete and search for the event details
3. Able to take attendance of members for a certain event
4. Able to search for the attendance list of members for a certain event
5. Able to insert, update and search for and generate the committee board of a certain year and print it
6. Able to generate a list of members present for a certain event and print it
7. Able to generate a potential contestant list and print it

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

**Entity Relationship (ER) Diagram**

\_\_\_\_\_ Primary Key

Event ID

Event Name

Event Venue

Event Date

Event Level

Event Chief Advisor

Event Advisor

Remark

Event ID

Member ID

Attendance

Member ID

Member Name

Member Class

Condition

Position

Highest English Grade

Potential Contestant

Guardian

Contact Number

contains

contains

has

has

EVENT

EVENT ATTENDANCE

MEMBER

3NF

tblMemberRecord (Member ID, Member Name, Member Class, Condition, Position,

Highest English Grade, Potential Contestant, Guardian,

Contact Number)

tblEventRecord (Event ID, Event Name, Event Venue, Event Date, Event Level,

Event Chief Advisor, Event Advisor, Remark)

tblEventAttendance (Event ID, Member ID, Attendance)

There is another table (tblCommitteeBoard) which has no relationship with the other tables, created by the request of the client, Miss Narulhuda to store the committee boards of each year so as to ease the production of printed output of the committee board every year. The data for the last four fields of this table is generated from the tblMemberRecord.

tblCommitteeBoard (Year, Chief Advisor, Advisor 1, Advisor 2, Advisor 3, Chairperson,

Vice Chairperson, Secretary, Treasurer)

**Data Dictionary**

The data dictionary includes all the tables in the database, with their descriptions.

Tables

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| tblMemberRecord | A table to store all the details of members. It is linked to tblEventAttendance via Member ID. |
| tblEventRecord | A table to store all the details of events. It is linked to tblEventAttendance via Event ID. |
| tblEventAttendance | A table to store the attendance of members to an event. It is linked to tblMemberRecord via Member ID and tblEventRecord via Event ID. |
| tblCommitteeBoard | A table to store the committee advisors and members of the year. |

Member Record Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length (bytes)** | **Sample Data** | **Constraints** | **Validation Check** | **Comment** |
| Member ID  (Primary Key) | Integer | 5 | 32100 | Digits only | Presence, Length, Character | Must contain 5 digits |
| Member Name | String | 50 | Ali Amir bin Dani | Characters only | Presence, Length Character |  |
| Member Class | String | 2 | 4B | Only within the choices available | Presence | Values set in combo box |
| Condition | String | 8 | Active | ‘Active’ or ‘Inactive’ | Presence | Values set in combo box |
| Position | String | 16 | None | Only within the choices available | Presence | Values set in combo box |
| Highest English Grade | String | 1 | D | Only within the choices available | Presence | Values set in combo box |
| Potential Contestant | String | 20 | Potential Contestant | ‘Potential Contestant’ or null | - | Depends on data in “Highest English Grade” and “Condition” fields |
| Guardian | String | 50 | Dani bin Ahmad Rahim | Characters only | Presence, Length, Character |  |
| Contact Number | String | 11 | 0197263722 | Numbers only | Presence, Length, Character | Between 9 to 11 numbers |

Event Record Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length (bytes)** | **Sample Data** | **Constraints** | **Validation Check** | **Comment** |
| Event ID  (Primary Key) | Integer | 4 | 1106 | Digits only | Presence | A four digit number generated automatically in ascending order starting from 1100 |
| Event Name | String | 30 | Happy English Talk | Characters and numbers | Presence, Length |  |
| Event Venue | String | 30 | School Hall | Characters and numbers | Presence, Length |  |
| Event Date | Date | 8 | 6/6/2015 | Short date only | - | Today’s date is generated automatically |
| Event Level | String | 20 | Zone | Characters only | Presence, Length, Character |  |
| Event Chief Advisor | String | 50 | Lim Yu Li | Characters only | Presence, Length, Character |  |
| Event Advisor | String | 50 | Norhaimi binti Yusof | Characters only | Presence, Length, Character |  |
| Remark | String | 100 | Held from 9 a.m. to 11 a.m. | Characters and numbers | Length |  |

Event Attendance Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length**  **(bytes)** | **Sample Data** | **Constraints** | **Validation Check** | **Comment** |
| Event ID  (Primary Key) | Integer | 4 | 1106 | Digits only | Presence | A four digit number generated automatically in ascending order starting from 1100 |
| Member ID  (Primary Key) | Integer | 5 | 32100 | Digits only | Presence, Length, Character | Must contain 5 digits |
| Attendance | String | 1 | 1 | ‘1’ or null | - | Generated according to the list box where the member is placed in the attendance form |

Committee Board Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length (bytes)** | **Sample Data** | **Constraints** | **Validation Check** | **Comment** |
| Year  (Primary Key) | Integer | 4 | 2015 | Digits only | Presence, Length, Character | Today’s year is generated automatically |
| Chief Advisor | String | 50 | Umairah binti Talib | Characters only | Presence, Length, Character |  |
| Advisor 1 | String | 50 | Lim Yu Li | Characters only | Presence, Length, Character |  |
| Advisor 2 | String | 50 | Norhaimi binti Yusof | Characters only | Presence, Length, Character |  |
| Advisor 3 | String | 50 | Tan Geun | Characters only | Presence, Length, Character |  |
| Chairperson | String | 50 | Amira binti Baderol | Characters only | Presence, Length, Character | Searched from tblMemberRecord |
| Vice Chairperson | String | 50 | Siti binti Ahmad | Characters only | Presence, Length, Character | Searched from tblMemberRecord |
| Secretary | String | 50 | Danial bin Sulaiman | Characters only | Presence, Length, Character | Searched from tblMemberRecord |
| Treasurer | String | 50 | Laili binti Azmi Osman | Characters only | Presence, Length, Character | Searched from tblMemberRecord |

**Input Design**

All data is entered into the database using forms designed using Visual Basic. Different forms will be used to link to different tables unless when the data is related. Then, the data inputted will be saved into their respective tables in the database. The following are the initial design of input forms proposed:

Member Record Form

Uses:

* To add a new member record into the database
* To edit an existing member record in the database
* To search for an existing member record from the database
* To delete a wrong member record from the database

1



10

8

6

4

2

12

11

9

7

5

3

15

14

13

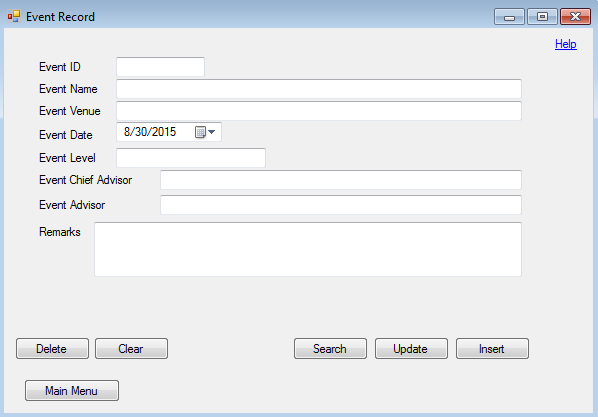
16

|  |  |
| --- | --- |
| **Number** | **Function** |
| 1 | Link label to allow the help page to be called. |
| 2 | Textbox to input the member ID, which is also the student ID, of the student/member to be registered as a member or whose details are to be edited/searched/deleted. |
| 3 | Combo box to choose the current condition of the member in the society. |
| 4 | Textbox to input the name of the student/member to be registered as a member or whose details are to be edited/searched. |
| 5 | Combo box to choose the class of the member. |
| 6 | Combo box to choose the position of the member in the society. |
| 7 | Combo box to choose the highest English grade achieved by the member in examinations. |
| 8 | Textbox to display whether the member is a potential contestant in the society. |
| 9 | Textbox to input the name of the parent/guardian of the member who easy to reach. |
| 10 | Textbox to input the contact number of the parent/guardian of the member which is reachable. |
| 11 | Button to allow the wrong member record to be deleted from the database.  (the student/member did not join the society, for example) |
| 12 | Button to clear and reset all the textboxes and choices made for combo boxes. |
| 13 | Button to allow a specific member record to be searched from the database via the member ID. |
| 14 | Button to allow a specific member record to be updated in the database via the member ID. |
| 15 | Button to allow a student to be registered as a member of the society and have his/her details to be saved in the database |
| 16 | Button to return to the main menu. |

Event Record Form

Use:

* To add a new event record into the database
* To edit an existing event record in the database
* To search for an existing event record from the database
* To delete a wrong event record from the database



14

7

6

5

4

3

2

1

8

9

10

11

12

13

15

|  |  |
| --- | --- |
| **Number** | **Function** |
| 1 | Link label to allow the help page to be called. |
| 2 | Textbox to input the ID of the event, which is generated automatically when the form is loaded, in which its details are to be edited/searched/deleted. |
| 3 | Textbox to input the name of the event. |
| 4 | Textbox to input the venue of the event. |
| 5 | Date time picker to select the date of the event. |
| 6 | Textbox to input the level of the event. |
| 7 | Textbox to input the name of the chief advisor who is in charge of the event. |
| 8 | Textbox to input the name of the advisor who is in charge of assisting the chief advisor of the event. |
| 9 | Textbox to input the remark or additional details of the event if there is any. |
| 10 | Button to allow the wrong event record to be deleted from the database.  (the event inputted with the wrong set of event details, for example) |
| 11 | Button to clear and reset all the textboxes and the date selected at the date time picker. |
| 12 | Button to allow a specific event record to be searched from the database via the event ID. |
| 13 | Button to allow a specific event record to be updated in the database via the event ID. |
| 14 | Button to allow a new event to be registered and have its details to be saved in the database. |
| 15 | Button to return to the main menu. |

Event Attendance Form

Uses:

* To take attendance of the members in a specific event
* To search for the attendance of the members in a specific event
* To update the attendance of the members in a specific event

7

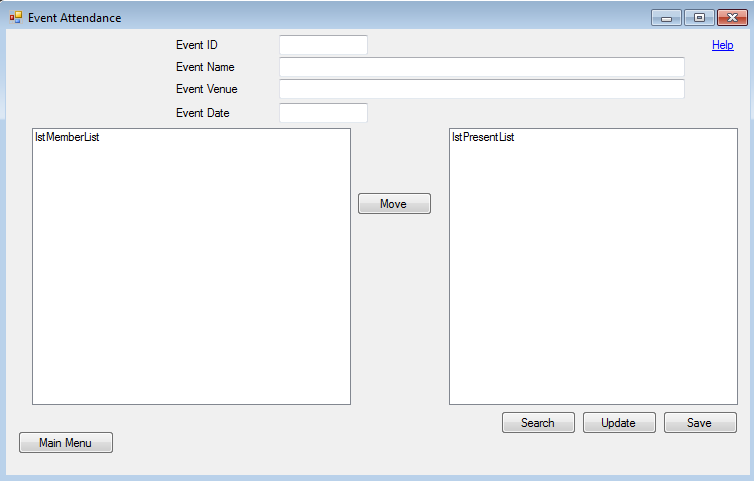
5

4

1

3

2



12

11

6

10

8

9

|  |  |
| --- | --- |
| **Number** | **Function** |
| 1 | Link label to allow the help page to be called. |
| 2 | Textbox to input the ID of the event, in which its details are already saved in the database. |
| 3 | Textbox to display the name of the event, which is generated automatically after the event ID is inputted. |
| 4 | Textbox to display the venue of the event, which is generated automatically after the event ID is inputted. |
| 5 | Textbox to display the date of the event, which is generated automatically after the event ID is inputted. |
| 6 | List box which contains the members who were absent in the event. |
| 7 | List box which contains the members who were present in the event. |
| 8 | Button to move the selected member from one list box to another. |
| 9 | Button to return to the main menu. |
| 10 | Button to allow a specific event’s attendance to be searched from the database via the event ID. |
| 11 | Button to allow a specific event’s attendance to be updated in the database via the event ID. |
| 12 | Button to allow a new event attendance to be saved in the database. |

Committee Board Form

Uses:

* To save the list of committee members of a particular year
* To search for the list of committee members of a particular year



16

11

10

9

8

6

5

4

3

2

1

7

12

13

14

15

|  |  |
| --- | --- |
| **Number** | **Function** |
| 1 | Link label to allow the help page to be called. |
| 2 | Link label to allow the committee board form to be print previewed and printed. |
| 3 | Textbox to input the year in which the committee board is effective. |
| 4 | Textbox to input the name of the chief advisor of the society. |
| 5 | Textbox to input the name of one of the advisors of the society who is in charge of assisting the chief advisor. |
| 6 | Textbox to input the name of one of the advisors of the society who is in charge of assisting the chief advisor. |
| 7 | Textbox to input the name of one of the advisors of the society who is in charge of assisting the chief advisor. |
| 8 | Textbox to display the name of the member who holds the position of the chairperson of the society, which is searched from the database. |
| 9 | Textbox to display the name of the member who holds the position of the vice chairperson of the society, which is searched from the database. |
| 10 | Textbox to display the name of the member who holds the position of the secretary of the society, which is searched from the database. |
| 11 | Textbox to display the name of the member who holds the position of the treasurer of the society, which is searched from the database. |
| 12 | Button to return to the main menu. |
| 13 | Button to clear and reset all the textboxes. |
| 14 | Button to allow a specific year’s committee board to be searched from the database via the year. |
| 15 | Button to allow a specific year’s committee board to be updated in the database via the year. |
| 16 | Button to allow a new committee board of a specific year to be saved in the database. |

**Feedback on the Input Design**

Miss Narulhuda questioned where the validation texts will appear if the data inputted is invalid. After learning that they will pop out as message boxes, she suggested to have labels on the input screens to ease their input processes.

Next, she wishes all the designs to be similar, so they should all have the same background colours. Since she wishes the output screens, shown in the later part, to have white backgrounds, then the input screens should also have white backgrounds.

She also insisted to have the event record form and event attendance form merged to simplify their work besides having a greater understanding of the attendance part. For the event attendance form, she would like to remove the “Update” function as it was unnecessary.

For the committee board form, she suggested to have the textboxes for the positions held by the members, not the teachers, to be hidden from view when the form is loaded since they will be searched from the database. If the committee board of a certain year is to be searched from the database then the textboxes for the positions held by the members should be shown with the respective data.

The following are the modified input designs:

Member Record Form



Newly added label to display validation texts when data inputted is invalid.

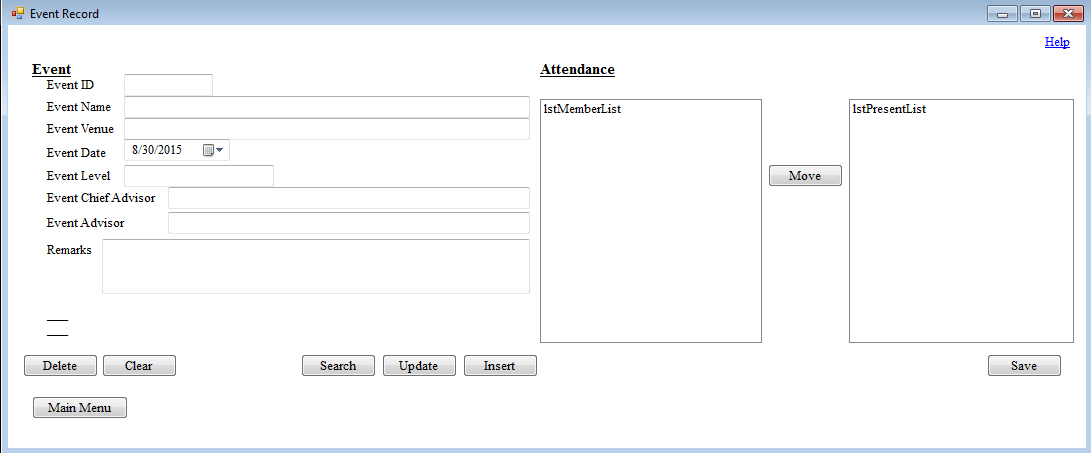
White background.

Newly added label to display validation texts when important fields are left blank.

Event Record Form

Merged event record form and event attendance form.

White background.

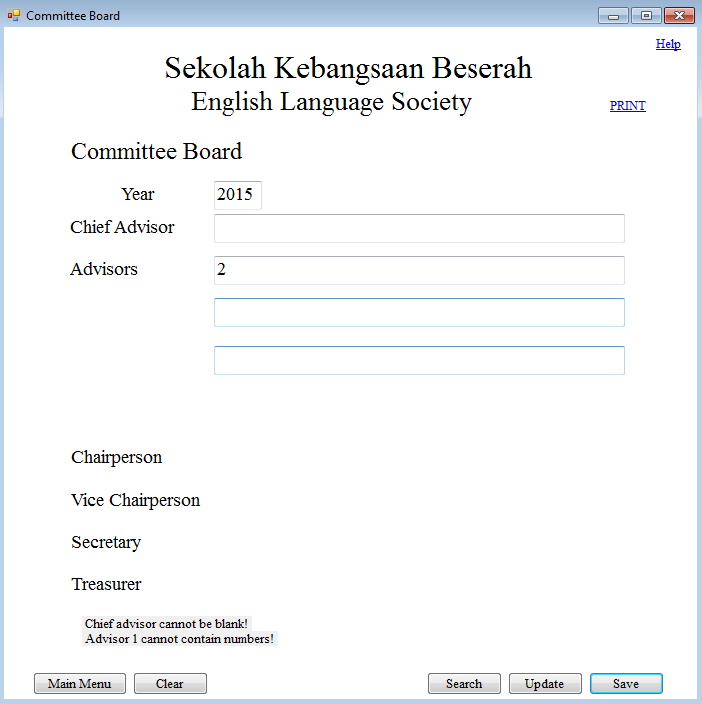


Newly added label to display validation texts when data inputted is invalid.

Newly added label to display validation texts when important fields are left blank.

Combined search function which works for both the event and the attendance.

Committee Board Form



Newly added label to display validation texts when important fields are left blank.

Textboxes are invisible when data is to be inputted.

White background.

Newly added label to display validation texts when data inputted is invalid.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

**Output Design**

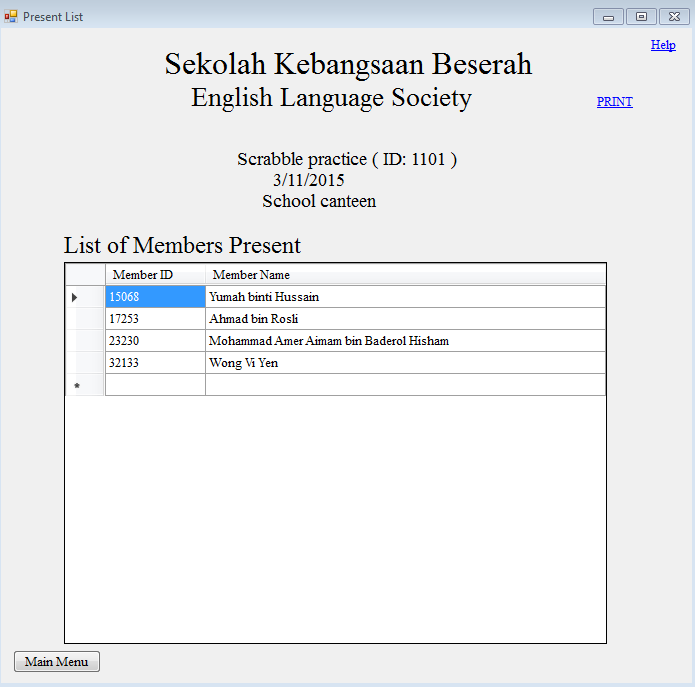
The following output screens are designed according the client’s wishes. They are also expected to be produced by the system.

List of Members Present

Use: To produce a list of members who were present in a specific event. 

Link label to allow the help page to be called.

Space to input the event ID of the specific event.



Button to return to the main menu.

A data grid view to display the Member ID and Member Name of the members who were present in a specific event.

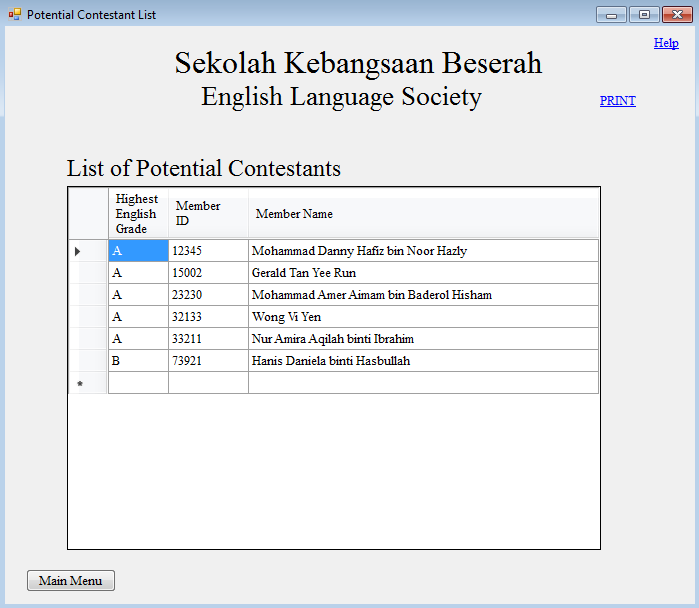
Labels to display the Event Name, Event ID, Event Date and Event Venue of the specific event.

Link label to allow the potential contestant list to be print previewed and printed.

Potential Contestant List

Use: To produce a list of members who are the potential contestants with their highest English grades of A or B.

Link label to allow the help page to be called.



Button to return to the main menu.

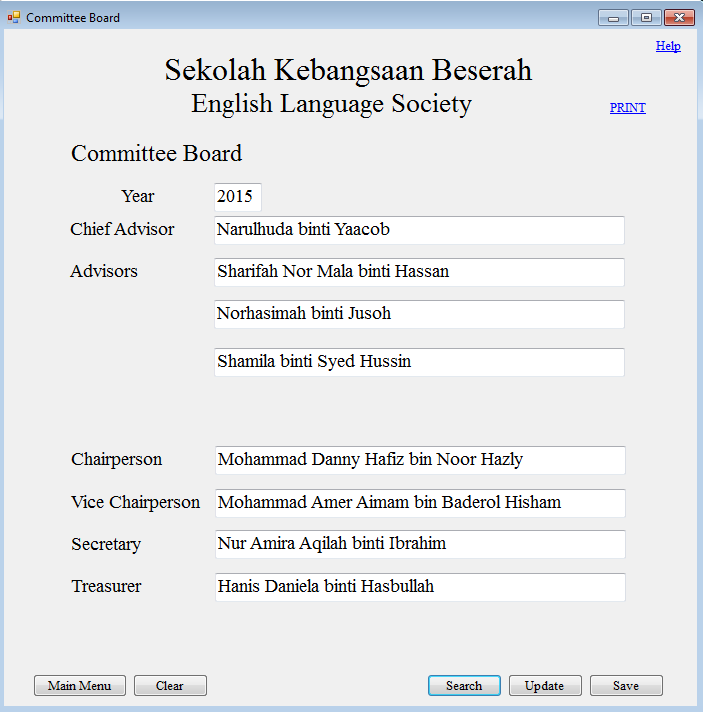
A data grid view to display the Highest English Grade, Member ID and Member Name of the potential contestants.

Link label to allow the potential contestant list to be print previewed and printed.

List of Committee Members

Use: To produce a printed output of committee board of a particular year

Link label to allow the help page to be called.



Button to allow a new committee board of a specific year to be saved in the database.

Textboxes to display the year in which the committee board is effective and names of the committee members.

Link label to allow the list of committee members to be print previewed and printed.

Button to allow a specific year’s committee board to be updated in the database via the year.

Button to return to the main menu.

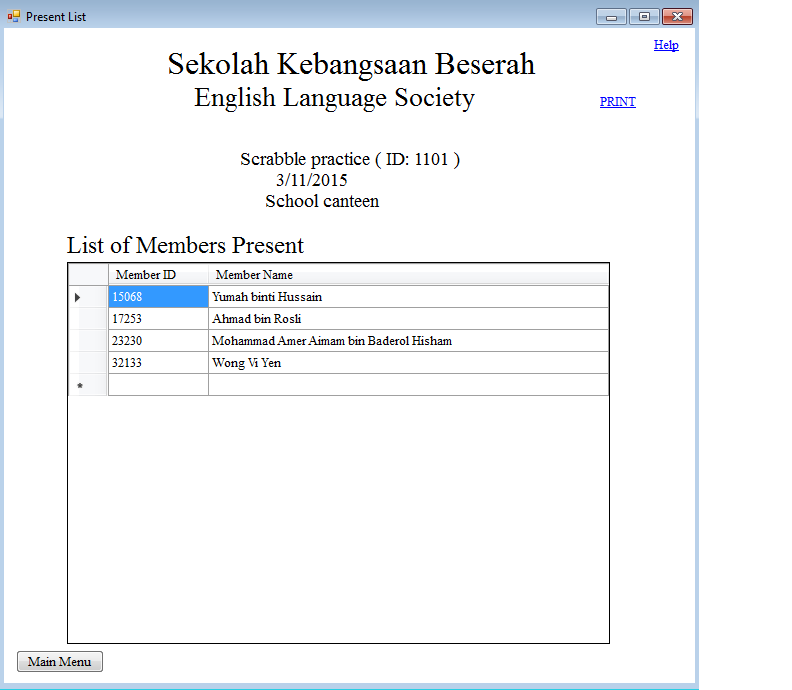
Button to allow a specific year’s committee board to be searched from the database via the year.

Button to clear and reset all the textboxes.

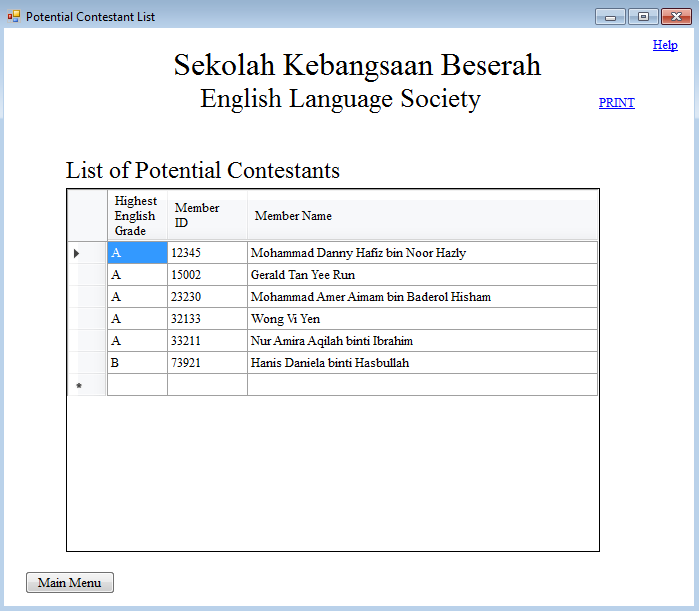
**Feedback on the Output Design**

Miss Narulhuda was satisfied with the layouts and designs of the outputs, but she wants the background colour of the outputs to be white to save cost on printing, as all the outputs will be printed on white A4 sized papers.

List of Members Present

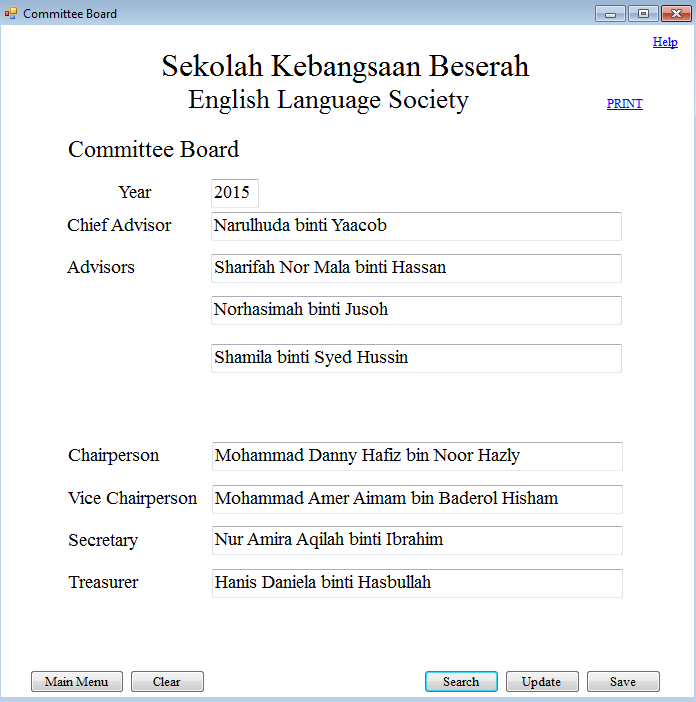
****Potential Contestant List

White background.

****

White background.

List of Committee Members

****

White background.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

**Test Plan**

All data inputted requires validation before they are saved in the database. However, some validation processes are repeated for similar fields. For instance, Member ID in Member Record form and Event Attendance form, Event ID in Event Record form and Event Attendance form, .

Member Record Form

Validation routines for member name and guardian are similar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Field** | **Test Type** | **Test Data** | **Expected Result** |
| 1 | Member ID | Valid | 11029 | 11029 |
| 2 | Member ID | Boundary | 99999 | 99999 |
| 3 | Member ID | Invalid | 2312 | Error |
| 4 | Condition | Valid | Active | Active |
| 5 | Member Name | Valid | Tan Jiyen | Tan Jiyen |
| 6 | Member Name | Boundary | Mohammad Amirul Khan bin Uzza Nafiz Nabirul Baderi | Mohammad Amirul Khan bin Uzza Nafiz Nabirul Baderi |
| 7 | Member Name | Invalid | Mohammad Ali Amirul Khan bin Uza Nafiz Nabirul Baderi | Error |
| 8 | Member Class | Valid | 4A | 4A |
| 9 | Member Class | Invalid |  | Error |
| 10 | Position | Valid | None | None |
| 11 | Position | Invalid |  | Error |
| 12 | Highest English Grade | Valid | A | A |
| 13 | Highest English Grade | Invalid |  | Error |
| 14 | Contact Number | Valid | 0129045745 | 0129045745 |
| 15 | Contact Number | Boundary | 01119053677 | 01119053677 |
| 16 | Contact Number | Invalid | 018-5546775 | Error |

Event Record Form

Validation routines for event name and event venue, event chief advisor and event advisor are the same while that of event ID and member ID are the same as that in the event record form and the member record form respectively.

Event ID does not need to be validated as it is generated automatically when this form is loaded.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Field** | **Test Type** | **Test Data** | **Expected Result** |
| 17 | Event Name | Valid | Happy English Talk 2014 | Happy English Talk 2014 |
| 18 | Event Name | Boundary | Year 14/15 Happy English Talk | Year 14/15 Happy English Talk |
| 19 | Event Name | Invalid | Year 2014/2015 Happy English Talk | Error |
| 20 | Event Date | Valid | 6/6/2015 | 6/6/2015 |
| 21 | Event Level | Valid | School | School |
| 22 | Event Level | Boundary | School Zone together | School Zone together |
| 23 | Event Level | Invalid | Zone 17/09 | Error |
| 24 | Event Chief Advisor | Valid | Tan Eng Eng | Tan Eng Eng |
| 25 | Event Chief Advisor | Boundary | Siti Aisyah Nuraini Shah binti Wakhrul Zani Osman | Siti Aisyah Nuraini Shah binti Wakhrul Zani Osman |
| 26 | Event Chief Advisor | Invalid | Ami Rusiah Fatimah binti Gural1 | Error |
| 27 | Remark | Valid | Held from 9 to 10 a.m. | Held from 9 to 10 a.m. |
| 28 | Remark | Boundary | The talk was conducted by an Australian Lecturer from Cambridge University from 12 p.m. to 1.30 p.m. | The talk was conducted by an Australian Lecturer from Cambridge University from 12 p.m. to 1.30 p.m. |
| 29 | Remark | Invalid | Talk was conducted by an Australian Lecturer from Cambridge University. It was held from 12 p.m. to 1.30 p.m. and is compulsory for all the year 4, 5 and 6 students to attend. | Error |
| 30 | Attendance | Functional | 1 | 1 |

Committee Board Form

Validation routines of chief advisor, advisor 1, advisor 2 and advisor 3 are the same as those of the event chief advisor and event advisor in the event record form.

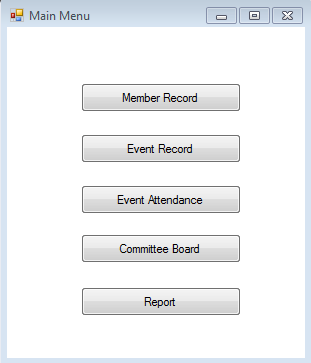
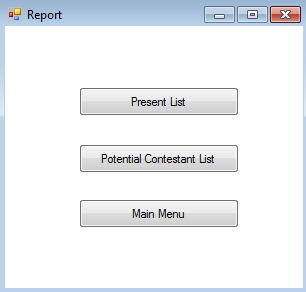
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Field** | **Test Type** | **Test Data** | **Expected Result** |
| 31 | Year | Valid | 2014 | 2014 |
| 32 | Year | Boundary | 1000 | 1000 |
| 33 | Year | Invalid | 2ooo | Error |

**Menu Design**

There are two menus for this system, in which one links to another.

Main Menu Form

Use: To allow the user to select the input operation to be performed or to output a certain report.



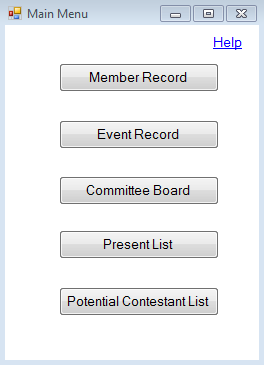
Report Form

Use: To allow the user to select the report to be outputted or to return to the main menu form.

**Feedback on the Menu Design**

After seeing the menu design, Miss Narulhuda proposed that she would like a main menu form which can link to all the operations available, including both input and output operations. Help should also be provided on the main menu form.

Main Menu Form

Use: To allow the user to select the operation to be performed from all that are available.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

1. **Intended Benefits**

The system designed above will bring several benefits to the client which makes it better than the previous manual system implemented by the client. These benefits are:

1. Easier storage and editing of data as all data are computerised and stored in the database. If data is to be edited, the required data is searched from the database then changed. This process takes less time compared to the previous system.
2. Backup is faster. This is done by making a set of copy of the programme and the database in the backup device. If the backup device is a flash drive, then it is possible to operate the system in other computers provided the programme can be installed into them. The previous paper works and paper documents are still filed up and stored for reference and in case the computerised data is lost.
3. Illegible handwritings can be minimised as all data are inputted into the system directly without going through the paper works. The searching processes are also done by the computers so any illegible handwritings would not be dealt with in this process.
4. More time-efficient as most processes are done by the computers. The operator only needs to select the correct button to perform the intended task.
5. Satisfaction of the client is achieved. This is because the system is designed according to the client’s proposals and wishes to allow them to learn, use and utilise the system to its full potential.
6. **Limits to the Scope**

**System Limitation**

Nothing is absolute, nothing is perfect. Limitations can failures. Therefore, it is important to learn the limitations and weaknesses of the system in order to produce a successful system.

One significant limitation is the software used to create this system, Visual Studio 2012. It is not a common software like Microsoft Word or Microsoft Excel which are installed in most computers sold in the market. In fact, it is quite difficult to find shops which sell the installer of this software. Also, an unfamiliar software can be hard to operate, especially for technology illiterates.

Next, the hardware and software equipped may cause limitations if they are of lower specification than the recommended hardware and software requirements. Then, the overall performance of the system may drop.

Lastly, the system is not capable of handling a large amount of data. It is able to handle the processing of data in a society as the society only contains a small amount of data. Problems regarding this issue will not appear as long as only small amount of data is dealt with by the system.

**File Size Estimation**

The estimated size of data stored in the database after a year is calculated below.

tblMemberRecord

Currently, there are 12 members active in the society.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Size for One Record (bytes)** |
| Member ID | Integer | 4 |
| Condition | String | 8 |
| Member Name | String | 50 |
| Member Class | String | 3 |
| Position | String | 16 |
| Highest English Grade | String | 1 |
| Potential Contestant | String | 20 |
| Guardian | String | 50 |
| Contact Number | String | 11 |

Total estimated bytes for one record = 163 bytes

Total estimated bytes for 12 records = 163 \* 12 = 1956 bytes

Total estimated kilobytes = 1956 / 1024 = 1.91 kilobytes

Total estimated kilobytes + 10 % = 1.91 \* 1.1 = 2.10 kilobytes

tblEventRecord

The size for this table is calculated by assuming that 20 events are held in a year.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Size for One Record (bytes)** |
| Event ID | Integer | 4 |
| Event Name | String | 30 |
| Event Venue | String | 30 |
| Event Date | Date | 8 |
| Event Level | String | 20 |
| Event Chief Advisor | String | 50 |
| Event Advisor | String | 50 |
| Remark | String | 100 |

Total estimated bytes for one record = 292 bytes

Total estimated bytes for 20 records = 292 \* 20 = 5840 bytes

Total estimated kilobytes = 5840 / 1024 = 5.70 kilobytes

Total estimated kilobytes + 10 % = 5.70 \* 1.1 = 6.27 kilobytes

tblEventAttendance

The size for this table is calculated by assuming that 20 events are held in a year and 12 members are active in the society throughout the year.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Size for One Record (bytes)** |
| Event ID | Integer | 4 |
| Member ID | Integer | 4 |
| Attendance | String | 1 |

Total estimated bytes for one record = 9 bytes

Total estimated bytes for records of one event = 9 \* 12 = 108 bytes

Total estimated bytes for records of one year = 108 \* 20 = 2160 bytes

Total estimated kilobytes = 2160 / 1024 = 2.11 kilobytes

Total estimated kilobytes + 10 % = 2.11 \* 1.1 = 2.32 kilobytes

tblCommitteeBoard

The size for this table is fixed unless more positions are added into the committee board.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Size for One Record (bytes)** |
| Year | Integer | 4 |
| Chief Advisor | String | 50 |
| Advisor 1 | String | 50 |
| Advisor 2 | String | 50 |
| Advisor 3 | String | 50 |
| Chairperson | String | 50 |
| Vice Chairperson | String | 50 |
| Secretary | String | 50 |
| Treasurer | String | 50 |

Total estimated bytes for one record = 404 bytes

Total estimated kilobytes = 404 / 1024 = 0.39 kilobytes

Total estimated kilobytes + 10 % = 0.39 \* 1.1 = 0.43 kilobytes

Grand Total

Total size of database in a year = 2.10 + 6.27 + 2.32 + 0.43 = 11.12 kilobytes

**(D) Software Development, Testing and**

**Implementation**

1. **Software Development**

**Development of Database**

Member Record Table

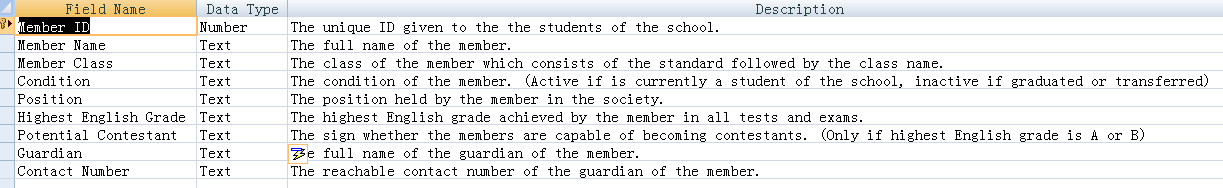


Diagram 1: Member Record Table Design View

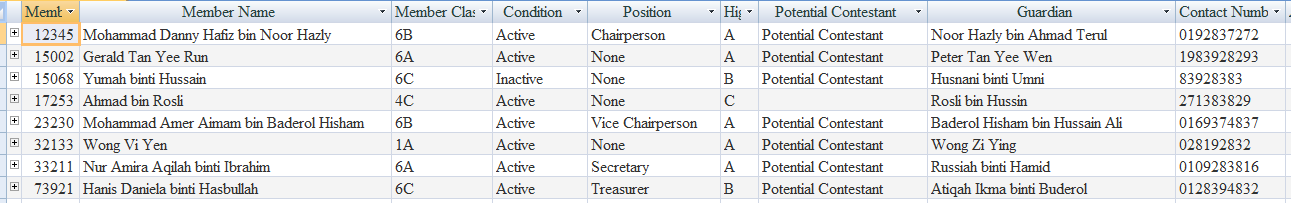


Diagram 2: Member Record Table Datasheet View

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length (bytes)** | **Description** | **Constraints** | **Validation Check** |
| Member ID  (Primary Key) | Integer | 5 | The unique ID given to the students of the school. | Digits only | Presence, Length, Character |
| Member Name | String | 50 | The full name of the member. | Characters only | Presence, Length Character |
| Member Class | String | 2 | The class of the member which consists of the standard followed by the class name. | Only within the choices available | Presence |
| Condition | String | 8 | The condition of the member. (Active if is currently a student of the school, inactive if graduated or transferred) | ‘Active’ or ‘Inactive’ | Presence |
| Position | String | 16 | The position held by the member in the society. | Only within the choices available | Presence |
| Highest English Grade | String | 1 | The highest English grade achieved by the member in all tests and exams. | Only within the choices available | Presence |
| Potential Contestant | String | 20 | The sign whether the members are capable of becoming contestants. (Only if highest English grade is A or B) | ‘Potential Contestant’ or null | - |
| Guardian | String | 50 | The full name of the guardian of the member. | Characters only | Presence, Length, Character |
| Contact Number | String | 11 | The reachable contact number of the guardian of the member. | Numbers only | Presence, Length, Character |

Event Record Table



Diagram 3: Event Record Table Design View



Diagram 4: Event Record Table Datasheet View

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length (bytes)** | **Description** | **Constraints** | **Validation Check** |
| Event ID  (Primary Key) | Integer | 4 | The unique ID given to the event which is continuous. | Digits only | Presence |
| Event Name | String | 30 | The name given to the event which is meaningful. | Characters and numbers | Presence, Length |
| Event Venue | String | 30 | The venue of which the event was held. | Characters and numbers | Presence, Length |
| Event Date | Date | 8 | The date of which the event was held. | Short date only | - |
| Event Level | String | 20 | The level of event which determines the importance of the event. | Characters only | Presence, Length, Character |
| Event Chief Advisor | String | 50 | The full name of the chief advisor who is in charge of the event. | Characters only | Presence, Length, Character |
| Event Advisor | String | 50 | The full name of the advisor who assists the chief advisor. | Characters only | Presence, Length, Character |
| Remark | String | 100 | Any other information or descriptions of the event. | Characters and numbers | Length |

Event Attendance Table

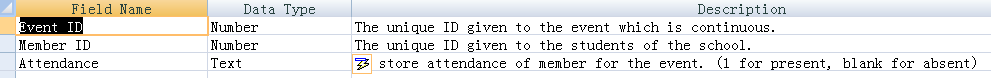


Diagram 5: Event Attendance Table Design View

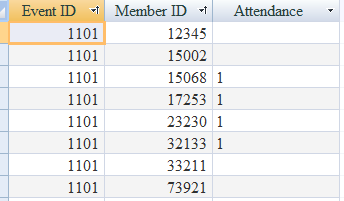


Diagram 6: Event Attendance Table Datasheet View

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length**  **(bytes)** | **Description** | **Constraints** | **Validation Check** |
| Event ID  (Primary Key) | Integer | 4 | The unique ID given to the event which is continuous. | Digits only | Presence |
| Member ID  (Primary Key) | Integer | 5 | The unique ID given to the students of the school. | Digits only | Presence, Length, Character |
| Attendance | String | 1 | To store attendance of member for the event. (1 for present, blank for absent) | ‘1’ or null | - |

Committee Board Table

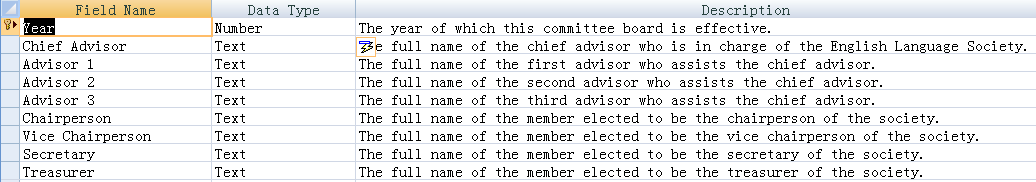


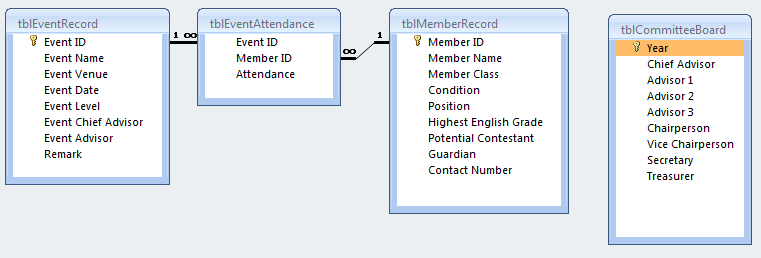
Diagram 7: Committee Board Table Design View

com 1.PNG

Diagram 8: Committee Board Table Datasheet View

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length (bytes)** | **Description** | **Constraints** | **Validation Check** |
| Year  (Primary Key) | Integer | 4 | The year of which this committee board is effective. | Digits only | Presence, Length, Character |
| Chief Advisor | String | 50 | The full name of the chief advisor who is in charge of the English Language Society. | Characters only | Presence, Length, Character |
| Advisor 1 | String | 50 | The full name of the first advisor who assists the chief advisor. | Characters only | Presence, Length, Character |
| Advisor 2 | String | 50 | The full name of the second advisor who assists the chief advisor. | Characters only | Presence, Length, Character |
| Advisor 3 | String | 50 | The full name of the third advisor who assists the chief advisor. | Characters only | Presence, Length, Character |
| Chairperson | String | 50 | The full name of the member elected to be the chairperson of the society. | Characters only | Presence, Length, Character |
| Vice Chairperson | String | 50 | The full name of the member elected to be the vice chairperson of the society. | Characters only | Presence, Length, Character |
| Secretary | String | 50 | The full name of the member elected to be the secretary of the society. | Characters only | Presence, Length, Character |
| Treasurer | String | 50 | The full name of the member elected to be the treasurer of the society. | Characters only | Presence, Length, Character |

The image below shows the relationships of all the tables in the database.



**Development of Input Methods**

Member Record Form

Uses:

* To add a new member record into the database
* To edit an existing member record in the database
* To search for an existing member record from the database
* To delete a wrong member record from the database



15

14

13

10

9

8

6

7

4

5

3

2

1

16

11

12

17

18

|  |  |
| --- | --- |
| **Number** | **Function** |
| 1 | Link label to allow the help page to be called. |
| 2 | Textbox to input the member ID, which is also the student ID, of the student/member to be registered as a member or whose details are to be edited/searched/deleted. |
| 3 | Combo box to choose the current condition of the member in the society. |
| 4 | Textbox to input the name of the student/member to be registered as a member or whose details are to be edited/searched. |
| 5 | Combo box to choose the class of the member. |
| 6 | Combo box to choose the position of the member in the society. |
| 7 | Combo box to choose the highest English grade achieved by the member in examinations. |
| 8 | Textbox to display whether the member is a potential contestant in the society. |
| 9 | Textbox to input the name of the parent/guardian of the member who easy to reach. |
| 10 | Textbox to input the contact number of the parent/guardian of the member which is reachable. |
| 11 | Button to allow the wrong member record to be deleted from the database.  (the student/member did not join the society, for example) |
| 12 | Button to clear and reset all the textboxes and choices made for combo boxes. |
| 13 | Button to allow a specific member record to be searched from the database via the member ID. |
| 14 | Button to allow a specific member record to be updated in the database via the member ID. |
| 15 | Button to allow a student to be registered as a member of the society and have his/her details to be saved in the database |
| 16 | Button to return to the main menu. |
| 17 | Label to display validation texts when important fields are left blank. |
| 18 | Label to display validation texts when data inputted is invalid. |

Event Record and Attendance Form

Use:

* To add a new event record into the database
* To take attendance of the members in a specific event
* To edit an existing event record in the database
* To search for an existing event record and its attendance of members from the database
* To delete a wrong event record from the database

1

7

3

9

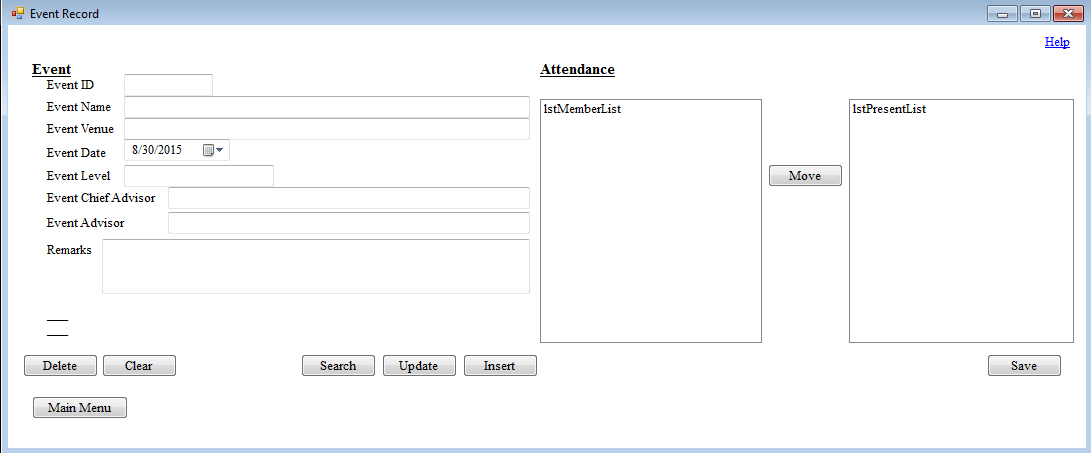
8

6

5

4

2



18

20

21

19

17

16

15

14

13

12

11

10

|  |  |
| --- | --- |
| **Number** | **Function** |
| 1 | Link label to allow the help page to be called. |
| 2 | Textbox to input the ID of the event, which is generated automatically when the form is loaded, in which its details are to be edited/searched/deleted. |
| 3 | Textbox to input the name of the event. |
| 4 | Textbox to input the venue of the event. |
| 5 | Date time picker to select the date of the event. |
| 6 | Textbox to input the level of the event. |
| 7 | Textbox to input the name of the chief advisor who is in charge of the event. |
| 8 | Textbox to input the name of the advisor who is in charge of assisting the chief advisor of the event. |
| 9 | Textbox to input the remark or additional details of the event if there is any. |
| 10 | Button to allow the wrong event record to be deleted from the database.  (the event inputted with the wrong set of event details, for example) |
| 11 | Button to clear and reset all the textboxes, the date selected at the date time picker and reload the attendance list. |
| 12 | Button to allow a specific event record and its attendance of the members to be searched from the database via the event ID. |
| 13 | Button to allow a specific event record to be updated in the database via the event ID. |
| 14 | Button to allow a new event to be registered and have its details to be saved in the database. |
| 15 | List box which contains the members who were absent in the event. |
| 16 | List box which contains the members who were present in the event. |
| 17 | Button to move the selected member from one list box to another. |
| 18 | Button to allow a new event attendance to be saved in the database. |
| 19 | Button to return to the main menu. |
| 20 | Label to display validation texts when important fields are left blank. |
| 21 | Label to display validation texts when data inputted is invalid. |

Committee Board Form

Uses:

* To save the list of committee members of a particular year
* To search for the list of committee members of a particular year



1

2

3

4

5

6

7

8

9

14

11

10

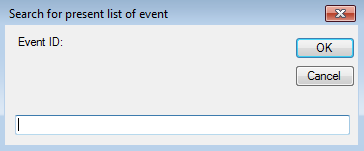
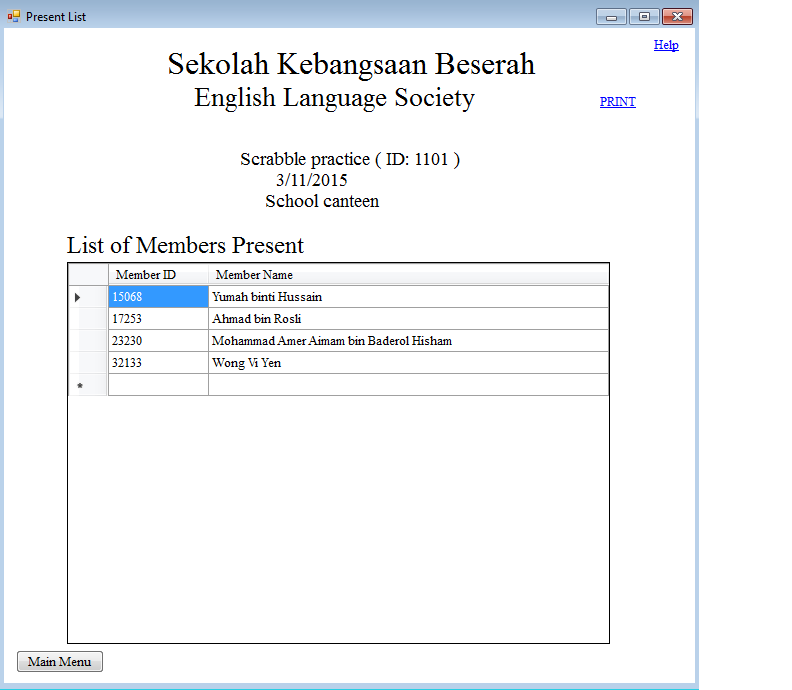
13

12

|  |  |
| --- | --- |
| **Number** | **Function** |
| 1 | Link label to allow the help page to be called. |
| 2 | Link label to allow the committee board form to be print previewed and printed. |
| 3 | Textbox to input the year in which the committee board is effective. |
| 4 | Textbox to input the name of the chief advisor of the society. |
| 5 | Textbox to input the name of one of the advisors of the society who is in charge of assisting the chief advisor. |
| 6 | Textbox to input the name of one of the advisors of the society who is in charge of assisting the chief advisor. |
| 7 | Textbox to input the name of one of the advisors of the society who is in charge of assisting the chief advisor. |
| 8 | Label to display validation texts when important fields are left blank. |
| 9 | Label to display validation texts when data inputted is invalid. |
| 10 | Button to return to the main menu. |
| 11 | Button to clear and reset all the textboxes. |
| 12 | Button to allow a specific year’s committee board to be searched from the database via the year. |
| 13 | Button to allow a specific year’s committee board to be updated in the database via the year. |
| 14 | Button to allow a new committee board of a specific year to be saved in the database. |

**Development of Output Methods**

List of Members Present

****Potential Contestant List

Labels to display the Event Name, Event ID, Event Date and Event Venue of the specific event.

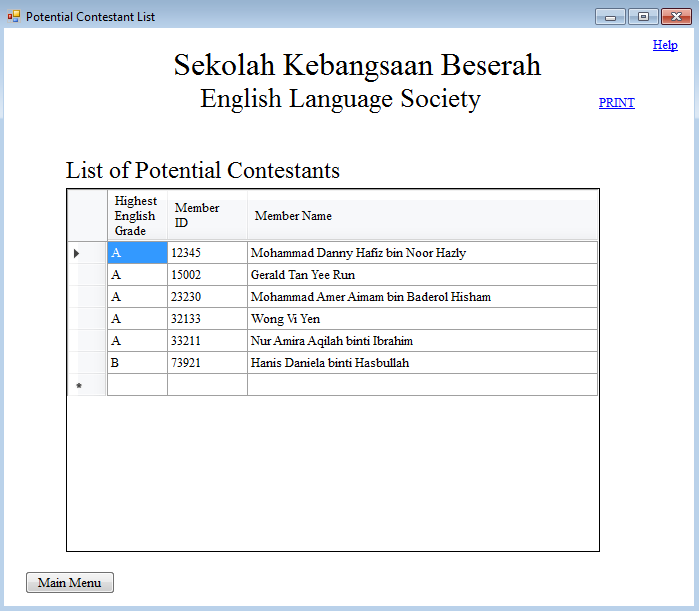
A data grid view to display the Member ID and Member Name of the members who were present in a specific event.

Button to return to the main menu.

Link label to allow the potential contestant list to be print previewed and printed.

Link label to allow the help page to be called.

Space to input the event ID of the specific event.

****

A data grid view to display the Highest English Grade, Member ID and Member Name of the potential contestants.

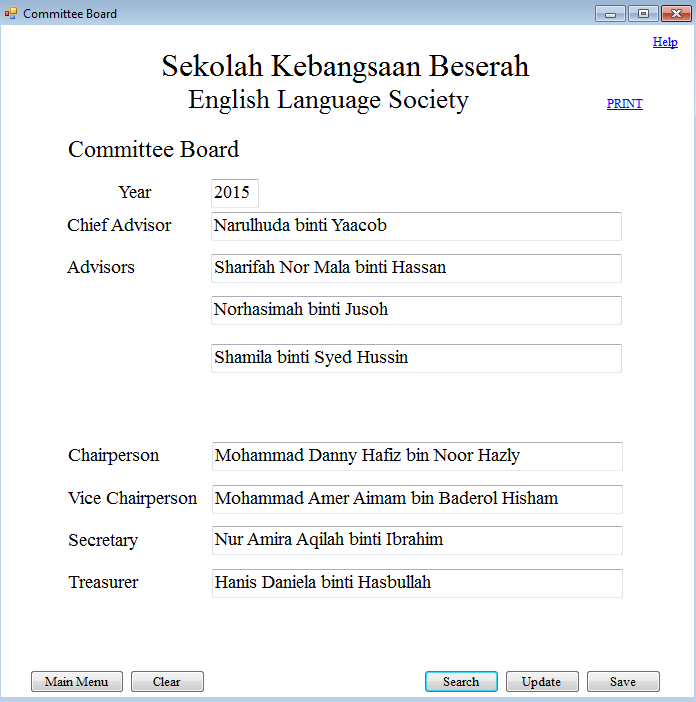
Link label to allow the potential contestant list to be print previewed and printed.

Link label to allow the help page to be called.

Button to return to the main menu.

List of Committee Members

Link label to allow the help page to be called.

****

Link label to allow the list of committee members to be print previewed and printed.

Button to allow a new committee board of a specific year to be saved in the database.

Button to allow a specific year’s committee board to be updated in the database via the year.

Button to allow a specific year’s committee board to be searched from the database via the year.

Button to clear and reset all the textboxes.

Textboxes to display the year in which the committee board is effective and names of the committee members.

Button to return to the main menu.

**Development of Database Queries**

1. "INSERT INTO tblMemberRecord([Member ID], [Condition], [Member Name], [Member Class], [Position], [Highest English Grade], " & \_

"[Potential Contestant], [Guardian], [Contact Number]) " & \_

"VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/Criteria** |
| Member Record Form | 314 | Insert parameter query | Inserts all data of a member inputted into the form into the database | tblMemberRecord | All | - |

1. "UPDATE tblMemberRecord " & \_

"SET [Member Name]='" & txtMemberName.Text & "'," & \_

"[Condition]='" & cboCondition.Text & "'," & \_

"[Member Class]='" & cboMemberClass.Text & "'," & \_

"[Position]='" & cboPosition.Text & "'," & \_

"[Highest English Grade]='" & cboHighestEnglishGrade.Text & "'," & \_

"[Potential Contestant]='" & txtPotentialContestant.Text & "'," & \_

"[Guardian]='" & txtGuardian.Text & "'," & \_

"[Contact Number]='" & txtContactNumber.Text & "' " & \_

"WHERE [Member ID]=" & MemberID

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/Criteria** |
| Member Record Form | 347 | Update parameter query | Updates all data of a member in the database according to the member ID | tblMemberRecord | All | - |

1. "SELECT \* FROM tblMemberRecord " & \_

"WHERE [Member ID]=" & MemberID

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Member Record Form | 377 | Select parameter query | Selects all data of a member from the database according to the member ID | tblMemberRecord | All | - |

1. "DELETE FROM tblMemberRecord " & \_

"WHERE [Member ID]=" & MemberID

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Member Record Form | 413 | Delete parameter query | Deletes all data of a member from the database according to the member ID | tblMemberRecord | All | - |

1. "SELECT MAX([Event ID]) As [MAX] FROM tblEventRecord"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Event Record Form | 33 | Select parameter query | Obtains the largest event ID from the database | tblEventRecord | Event ID | If there is no event ID in database, then 1101 is used |

1. "INSERT INTO tblEventRecord([Event ID], [Event Name], [Event Venue], [Event Date], [Event Level], " & \_

"[Event Chief Advisor], [Event Advisor], [Remark]) " & \_

"VALUES (?, ?, ?, ?, ?, ?, ?, ?)"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Event Record Form | 332 | Insert parameter query | Inserts all data of an event inputted into the form into the database | tblEventRecord | All | - |

1. "UPDATE tblEventRecord " & \_

"SET [Event Name]='" & txtEventName.Text & "'," & \_

"[Event Venue]='" & txtEventVenue.Text & "'," & \_

"[Event Date]='" & dtpEventDate.Text & "'," & \_

"[Event Level]='" & txtEventLevel.Text & "'," & \_

"[Event Chief Advisor]='" & txtEventChiefAdvisor.Text & "'," & \_

"[Event Advisor]='" & txtEventAdvisor.Text & "'," & \_

"[Remark]='" & txtRemark.Text & "' " & \_

"WHERE [Event ID]=" & EventID

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Event Record Form | 367 | Update parameter query | Updates all data of an event into the database according to the event ID | tblEventRecord | All | - |

1. "SELECT \* FROM tblEventRecord " & \_

"WHERE [Event ID]=" & EventID

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Event Record Form | 399 | Select parameter query | Selects all data of an event into the database according to the event ID | tblEventRecord | All | - |

1. "SELECT tblEventRecord.[Event ID] FROM tblEventRecord INNER JOIN tblEventAttendance ON " & \_

"tblEventRecord.[Event ID] = tblEventAttendance.[Event ID]" & \_

"WHERE (((tblEventRecord.[Event ID])=" & EventID & "))"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Event Record Form | 430 | Delete parameter query | Deletes all data of an event from the database according to the event ID | tblEventRecord, tblEventAttendance | All | - |

1. "INSERT INTO tblEventAttendance([Event ID], [Member ID], [Attendance]) VALUES (?, ?, ?)"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Event Record Form | 552 | Insert parameter query | Inserts all data of an event attendance inputted into the form into the database | tblEventAttendance | All | - |

1. "SELECT tblEventAttendance.[Event ID], tblEventAttendance.[Member ID], tblEventAttendance.Attendance, " & \_

"tblMemberRecord.[Member Name] FROM tblMemberRecord INNER JOIN tblEventAttendance " & \_

"ON tblMemberRecord.[Member ID] = tblEventAttendance.[Member ID]" & \_

"WHERE (((tblEventAttendance.[Event ID])=" & EventID & \_

")) ORDER BY tblEventAttendance.[Member ID]"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Event Record Form | 604 | Delete parameter query | Deletes all data of an event from the database according to the event ID | tblEventRecord, tblEventAttendance | All | - |

1. "SELECT [Member Name] " & \_

"FROM tblMemberRecord " & \_

"WHERE (([Condition]='Active') AND ([Position]='Chairperson'))"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Committee Board Form | 284 | Select parameter query | Obtains the name of the chairperson of the society of a specific year from the database according to the position and condition | tblMemberRecord, tblCommitteeBoard | Member Name, Position, Condition | Position = ‘Chairperson’ and Condition = ‘Active’ |

1. "SELECT [Member Name] " & \_

"FROM tblMemberRecord " & \_

"WHERE (([Condition]='Active') AND ([Position]='Vice Chairperson'))"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Committee Board Form | 301 | Select parameter query | Obtains the name of the vice chairperson of the society of a specific year from the database according to the position and condition | tblMemberRecord, tblCommitteeBoard | Member Name, Position, Condition | Position = ‘Vice Chairperson’ and Condition = ‘Active’ |

1. "SELECT [Member Name] " & \_

"FROM tblMemberRecord " & \_

"WHERE (([Condition]='Active') AND ([Position]='Secretary'))"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Committee Board Form | 318 | Select parameter query | Obtains the name of the secretary of the society of a specific year from the database according to the position and condition | tblMemberRecord, tblCommitteeBoard | Member Name, Position, Condition | Position = ‘Secretary’ and Condition = ‘Active’ |

1. "SELECT [Member Name] " & \_

"FROM tblMemberRecord " & \_

"WHERE (([Condition]='Active') AND ([Position]='Treasurer'))"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Committee Board Form | 335 | Select parameter query | Obtains the name of the treasurer of the society of a specific year from the database according to the position and condition | tblMemberRecord, tblCommitteeBoard | Member Name, Position, Condition | Position = ‘Treasurer’ and Condition = ‘Active’ |

1. "INSERT INTO tblCommitteeBoard([Chief Advisor], [Advisor 1], [Advisor 2], [Advisor 3], [Year], " & \_

"[Chairperson], [Vice Chairperson], [Secretary], [Treasurer]) " & \_

"VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Committee Board Form | 353 | Insert parameter query | Inserts all data of a committee board into the database | tblCommitteeBoard | All | - |

1. "SELECT \* FROM tblCommitteeBoard " & \_

"WHERE [Year]=" & Year

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Committee Board Form | 394 | Select parameter query | Selects all data of a committee board from the database according to the year | tblCommitteeBoard | All | - |

1. "UPDATE tblCommitteeBoard " & \_

"SET [Chief Advisor]='" & txtChiefAdvisor.Text & "'," & \_

"[Advisor 1]='" & txtAdvisor1.Text & "'," & \_

"[Advisor 2]='" & txtAdvisor2.Text & "'," & \_

"[Advisor 3]='" & txtAdvisor3.Text & "'," & \_

"WHERE [Year]=" & Year

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Committee Board Form | 426 | Update parameter query | Updates all data of a committee board in the database according to the year | tblCommitteeBoard | All | - |

1. "SELECT tblEventAttendance.[Event ID], tblEventRecord.[Event Name], tblEventRecord.[Event Venue], " & \_

"tblEventRecord.[Event Date], tblEventAttendance.[Member ID], tblMemberRecord.[Member Name], " & \_

"tblEventAttendance.Attendance FROM tblMemberRecord INNER JOIN (tblEventRecord INNER JOIN " & \_

"tblEventAttendance ON tblEventRecord.[Event ID] = tblEventAttendance.[Event ID]) ON " & \_

"tblMemberRecord.[Member ID] = tblEventAttendance.[Member ID]" & \_

"WHERE (((tblEventAttendance.[Event ID])=" & EventID & ") AND ((tblEventAttendance.Attendance)='1')) " & \_

"ORDER BY tblEventAttendance.[Member ID]"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Present List Form | 38 | Select parameter query | Selects the name and ID of members who were present in a specific event from the database according to the attendance and event ID | tblEventAttendance, tblEventRecord, tblMemberRecord | Event ID, Event Name, Event Venue, Event Date, Member ID, Member Name | Attendance = ‘1’ |

1. "SELECT tblMemberRecord.[Highest English Grade], tblMemberRecord.[Potential Contestant], " & \_

"tblMemberRecord.[Member Name], tblMemberRecord.[Member ID] FROM tblMemberRecord " & \_

"WHERE (tblMemberRecord.[Potential Contestant]='Potential Contestant') " & \_

"ORDER BY tblMemberRecord.[Highest English Grade], tblMemberRecord.[Member ID]"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Form** | **Line** | **Query Type** | **Purpose** | **Table Accessed** | **Fields Accessed** | **Notes/ Criteria** |
| Potential Contestant List Form | 30 | Select parameter query | Selects the name and ID of members who are potential contestants currently active in the society from the database according to the condition and potential contestant status | tblMemberRecord | Highest English Grade, Potential Contestant, Member ID, Member Name | Potential Contestant = ‘Potential Contestant’, order according to Highest English Grade and Member ID |

1. **Programming**

**Code Listing**

Main Menu Form

Public Class frmMainMenu

Private Sub frmMainMenu\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Displays simple instructions when pointed

With ToolTip1

.SetToolTip(btnMemberRecord, "Opens member record form")

.SetToolTip(btnEventRecord, "Opens event record form")

.SetToolTip(btnCommitteeBoard, "Opens committee board form")

.SetToolTip(btnPresentList, "Lists out the present members in a specific event")

.SetToolTip(btnPotentialContestants, "Lists out the active members who have high english grades and are listed as potential contestants")

End With

End Sub

Private Sub btnMemberRecord\_Click(sender As Object, e As EventArgs) Handles btnMemberRecord.Click

'Closes the main menu and opens the member record form

Me.Hide()

frmMemberRecord.Show()

End Sub

Private Sub btnEventRecord\_Click(sender As Object, e As EventArgs) Handles btnEventRecord.Click

'Closes the main menu and opens the event record form

Me.Hide()

frmEventRecord.Show()

End Sub

Private Sub btnCommitteeBoard\_Click(sender As Object, e As EventArgs) Handles btnCommitteeBoard.Click

'Closes the main menu and opens committee board form

Me.Hide()

frmCommitteeBoard.Show()

End Sub

Private Sub btnPresentList\_Click(sender As Object, e As EventArgs) Handles btnPresentList.Click

'Closes the main menu and opens the present list form

Me.Hide()

frmPresentList.Show()

End Sub

Private Sub btnPotentialContestants\_Click(sender As Object, e As EventArgs) Handles btnPotentialContestants.Click

'Closes the main menu and opens the potential contestant list form

Me.Hide()

frmPotentialContestantList.Show()

End Sub

Private Sub lnkHelp\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkHelp.LinkClicked

'Opens the index of help page

System.Diagnostics.Process.Start("Help\Index.htm")

End Sub

End Class

Member Record Form

Imports System.Data

Imports System.Data.OleDb

Public Class frmMemberRecord

Dim provider As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source ="

Dim dataFile As String = "Beserah Primary School English Language Society Management System.accdb"

Dim connString As String = provider & dataFile 'Path of database file

Dim myConnection As New OleDbConnection 'Joins provider and path

'Data reader to read data from Access Database

Dim dr As OleDbDataReader

Dim validationFlag As Boolean = False

Dim searchFlag As Boolean = False

Sub StartConnection()

'Connects the programme with the database

myConnection.ConnectionString = connString

myConnection.Open()

End Sub

Sub CloseConnection()

'Closes the connection between the programme and the database

myConnection.Close()

End Sub

Private Sub txtMemberID\_TextChanged(sender As Object, e As EventArgs) Handles txtMemberID.TextChanged

ValidateMemberID()

End Sub

Private Sub txtMemberName\_TextChanged(sender As Object, e As EventArgs) Handles txtMemberName.TextChanged

ValidateMemberName()

End Sub

Private Sub txtGuardian\_TextChanged(sender As Object, e As EventArgs) Handles txtGuardian.TextChanged

ValidateGuardian()

End Sub

Private Sub txtContactNumber\_TextChanged(sender As Object, e As EventArgs) Handles txtContactNumber.TextChanged

ValidateContactNumber()

End Sub

Private Sub txtPotentialContestant\_TextChanged(sender As Object, e As EventArgs) Handles txtPotentialContestant.TextChanged

AutoSetPotentialContestant()

End Sub

Sub ValidateMemberID()

'Only 5 digits allowed for member ID

Dim length As Integer = txtMemberID.Text.Length

Dim i As Integer = 0

If length <> 5 Then

lblError.Text = "Member ID must contain 5 digits only!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

If IsNumeric(txtMemberID.Text) = False Then

lblError.Text = "Member ID can only contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

End Sub

Sub ValidateMemberName()

'Member name cannot be too long, numbers are not allowed

Dim length As Integer = txtMemberName.Text.Length

Dim i As Integer = 0

If length > 50 Then

lblError.Text = "The member name is too long! (>50 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtMemberName.Text.Substring(i, 1)) >= 48 And Asc(txtMemberName.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Member name cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub AutoSetPotentialContestant()

'Automatically set member as potential contestant if condition is active and highest english grade achieved by member is A or B

If cboCondition.Text = "Active" Then

If (cboHighestEnglishGrade.Text = "B") Or (cboHighestEnglishGrade.Text = "A") Then

txtPotentialContestant.Text = "Potential Contestant"

txtPotentialContestant.Visible = True

End If

Else

txtPotentialContestant.Text = ""

End If

End Sub

Sub ValidateGuardian()

'Guardian cannot be too long, numbers are not allowed

Dim length As Integer = txtGuardian.Text.Length

Dim i As Integer = 0

If length > 50 Then

lblError.Text = "The guardian is too long! (>50 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtGuardian.Text.Substring(i, 1)) >= 48 And Asc(txtGuardian.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Guardian cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub ValidateContactNumber()

'Guardian cannot be too long or too short, only numbers are allowed

Dim length As Integer = txtContactNumber.Text.Length

If length > 11 Then

lblError.Text = "The contact number is too long! (>11 numbers)"

lblError.Visible = True

validationFlag = False

Exit Sub

ElseIf length < 9 Then

lblError.Text = "The contact number is too short! (<9 numbers)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

If IsNumeric(txtContactNumber.Text) = False Then

lblError.Text = "Contact number can only contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

End Sub

Sub BlankText()

'To ensure certain fields are not blank

If txtMemberID.Text = "" Then

lblBlank.Text = "Member ID cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If cboCondition.Text = "" Then

lblBlank.Text = "Condition cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtMemberName.Text = "" Then

lblBlank.Text = "Member name cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If cboMemberClass.Text = "" Then

lblBlank.Text = "Member class cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If cboPosition.Text = "" Then

lblBlank.Text = "Position cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If cboHighestEnglishGrade.Text = "" Then

lblBlank.Text = "Highest english grade cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtGuardian.Text = "" Then

lblBlank.Text = "Guardian cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtContactNumber.Text = "" Then

lblBlank.Text = "Contact number cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

End Sub

Sub ClearFields()

'Clear all the fields

txtMemberID.Text = ""

cboCondition.Text = "Active"

txtMemberName.Text = ""

cboMemberClass.Text = "1A"

cboPosition.Text = "None"

cboHighestEnglishGrade.Text = "F"

txtPotentialContestant.Visible = False 'Makes the potential contestant text box invisible

txtPotentialContestant.Text = ""

txtGuardian.Text = ""

txtContactNumber.Text = ""

End Sub

Sub FocusCursor()

'Focus cursor at the member ID text box

txtMemberID.Focus()

End Sub

Private Sub frmMemberRecord\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'On loading, a connection is opened to connect VB and MS Access

'Displays simple instructions when pointed

With ToolTip1

.SetToolTip(txtMemberID, "Enter member ID (e.g. EL0903)")

.SetToolTip(cboCondition, "Current condition of the member, active if is a student of the school, inactive if graduated or transferred")

.SetToolTip(txtMemberName, "Enter the name of member (e.g. Gan Bee Hua)")

.SetToolTip(cboMemberClass, "Select the class of member")

.SetToolTip(cboPosition, "Select the position of member in this society")

.SetToolTip(cboHighestEnglishGrade, "Select the highest english grade achieved by the member")

.SetToolTip(lblPotentialContestant, "A member is a potential contestant only if he/ she achieved highest english grade of A or B")

.SetToolTip(txtGuardian, "Enter the name of the guardian of the member (e.g. Gan Woo Beng)")

.SetToolTip(txtContactNumber, "Enter the contact number of the guardian (e.g. 0199376428)")

.SetToolTip(btnClear, "Clears all the fields")

.SetToolTip(btnInsert, "Creates a new member record in the database")

.SetToolTip(btnUpdate, "Updates an existing member record in the database")

.SetToolTip(btnSearch, "Searches for a specific member in the database")

.SetToolTip(btnDelete, "Deletes a member record from database")

.SetToolTip(btnMainMenu, "Go to the main menu")

End With

ClearFields()

FocusCursor()

lblError.Visible = False 'Makes the error label invisible to the user when this form is loaded

lblBlank.Visible = False 'Makes the blank label invisible to the user when this form is loaded

End Sub

Sub InsertNewMember()

'Adds a new member record into database

StartConnection()

Dim sql As String = "INSERT INTO tblMemberRecord([Member ID], [Condition], [Member Name], [Member Class], [Position], [Highest English Grade], " & \_

"[Potential Contestant], [Guardian], [Contact Number]) " & \_

"VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)"

Dim cmd As New OleDbCommand(sql, myConnection)

With cmd.Parameters

.Add(New OleDbParameter("[Member ID]", CType(txtMemberID.Text, Integer)))

.Add(New OleDbParameter("[Condition]", CType(cboCondition.Text, String)))

.Add(New OleDbParameter("[Member Name]", CType(txtMemberName.Text, String)))

.Add(New OleDbParameter("[Member Class]", CType(cboMemberClass.Text, String)))

.Add(New OleDbParameter("[Position]", CType(cboPosition.Text, String)))

.Add(New OleDbParameter("[Highest English Grade]", CType(cboHighestEnglishGrade.Text, String)))

.Add(New OleDbParameter("[Potential Contestant]", CType(txtPotentialContestant.Text, String)))

.Add(New OleDbParameter("[Guardian]", CType(txtGuardian.Text, String)))

.Add(New OleDbParameter("[Contact Number]", CType(txtContactNumber.Text, String)))

End With

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CloseConnection()

ClearFields()

FocusCursor()

End Sub

Sub UpdateMember()

'Updates a specific member record in the database

StartConnection()

Dim MemberID As Integer = CInt(txtMemberID.Text)

Dim sql As String = "UPDATE tblMemberRecord " & \_

"SET [Member Name]='" & txtMemberName.Text & "'," & \_

"[Condition]='" & cboCondition.Text & "'," & \_

"[Member Class]='" & cboMemberClass.Text & "'," & \_

"[Position]='" & cboPosition.Text & "'," & \_

"[Highest English Grade]='" & cboHighestEnglishGrade.Text & "'," & \_

"[Potential Contestant]='" & txtPotentialContestant.Text & "'," & \_

"[Guardian]='" & txtGuardian.Text & "'," & \_

"[Contact Number]='" & txtContactNumber.Text & "' " & \_

"WHERE [Member ID]=" & MemberID

Dim cmd As New OleDbCommand(sql, myConnection)

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CloseConnection()

ClearFields()

FocusCursor()

End Sub

Sub SearchMember()

'Searches for a specific member in the database through its ID and then displays it

StartConnection()

'Pops out an input box to enter the specific member's ID

Dim MemberID As Integer = CInt(InputBox("Enter Member ID: ", "Search Member"))

Dim sql As String = "SELECT \* FROM tblMemberRecord " & \_

"WHERE [Member ID]=" & MemberID

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

'Fills up the information about the member

While dr.Read()

txtMemberID.Text = dr("Member ID").ToString

cboCondition.Text = dr("Condition").ToString

txtMemberName.Text = dr("Member Name").ToString

cboMemberClass.Text = dr("Member Class").ToString

cboPosition.Text = dr("Position").ToString

cboHighestEnglishGrade.Text = dr("Highest English Grade").ToString

txtPotentialContestant.Text = dr("Potential Contestant").ToString

txtGuardian.Text = dr("Guardian").ToString

txtContactNumber.Text = dr("Contact Number").ToString

searchFlag = True

End While

If txtPotentialContestant.Text = "Potential Contestant" Then

txtPotentialContestant.Visible = True

End If

If searchFlag = False Then

MsgBox("Member ID " & MemberID & " not found!", MsgBoxStyle.OkOnly + MsgBoxStyle.Exclamation, "Error")

End If

CloseConnection()

End Sub

Sub DeleteMember()

'Deletes a member record from the database

StartConnection()

Dim MemberID As Integer = CInt(txtMemberID.Text)

Dim sql As String = "DELETE FROM tblMemberRecord " & \_

"WHERE [Member ID]=" & MemberID

Dim cmd As New OleDbCommand(sql, myConnection)

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CloseConnection()

ClearFields()

FocusCursor()

End Sub

Private Sub btnInsert\_Click(sender As Object, e As EventArgs) Handles btnInsert.Click

'Saves member details into database after validation

BlankText()

AutoSetPotentialContestant()

If validationFlag = True Then

InsertNewMember()

MsgBox("Member record added successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

End If

End Sub

Private Sub btnUpdate\_Click(sender As Object, e As EventArgs) Handles btnUpdate.Click

'Updates member details in database after validation

BlankText()

AutoSetPotentialContestant()

If validationFlag = True Then

UpdateMember()

MsgBox("Member record updated successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

End If

End Sub

Private Sub btnSearch\_Click(sender As Object, e As EventArgs) Handles btnSearch.Click

SearchMember()

End Sub

Private Sub btnClear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click

ClearFields()

FocusCursor()

End Sub

Private Sub btnDelete\_Click(sender As Object, e As EventArgs) Handles btnDelete.Click

If txtMemberID.Text = "" Then

MsgBox("Please enter the member ID whose record is to be deleted!", MsgBoxStyle.OkOnly + MsgBoxStyle.Exclamation, "Error")

Exit Sub

End If

'Prompts out a message box to confirm deletion, deletes the specific member record if deletion is confirmed

If MsgBox("Are you sure you want to delete this record?", \_

MsgBoxStyle.YesNo + MsgBoxStyle.Critical, "Confirmation") = MsgBoxResult.Yes Then

DeleteMember()

MsgBox("Member record deleted successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

Else

Exit Sub

End If

End Sub

Private Sub btnMainMenu\_Click(sender As Object, e As EventArgs) Handles btnMainMenu.Click

'Closes the member record form and opens the main menu

Me.Hide()

frmMainMenu.Show()

End Sub

Private Sub lnkHelp\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkHelp.LinkClicked

'Opens the index of help page

System.Diagnostics.Process.Start("Help\Index.htm")

End Sub

End Class

Event Record and Event Attendance Form

Imports System.Data

Imports System.Data.OleDb

Public Class frmEventRecord

Dim provider As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source ="

Dim dataFile As String = "Beserah Primary School English Language Society Management System.accdb"

Dim connString As String = provider & dataFile 'Path of database file

Dim myConnection As New OleDbConnection 'Joins provider and path

Dim dr As OleDbDataReader 'Data reader to read data from Access Database

Dim validationFlag As Boolean = False

Dim searchFlag As Boolean = False

Sub StartConnection()

'Connects the programme with the database

myConnection.ConnectionString = connString

myConnection.Open()

End Sub

Sub CloseConnection()

'Closes the connection between the programme and the database

myConnection.Close()

End Sub

Sub GenerateEventID()

'The programme finds the maximum event ID from the database and adds 1 to it to generate the new event ID

StartConnection()

Dim sql As String = "SELECT MAX([Event ID]) As [MAX] FROM tblEventRecord"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

While dr.Read()

If dr("Max").ToString = "" Then

txtEventID.Text = "1101"

Else

txtEventID.Text = ((dr("MAX")) + 1).ToString

End If

End While

CloseConnection()

End Sub

Sub ClearFields()

'Clear all the fields

txtEventID.Text = ""

txtEventName.Text = ""

txtEventVenue.Text = ""

dtpEventDate.Text = Date.Now

txtEventLevel.Text = ""

txtEventChiefAdvisor.Text = ""

txtEventAdvisor.Text = ""

txtRemark.Text = ""

End Sub

Sub FocusCursor()

'Focus cursor at the event name text box

txtEventName.Focus()

End Sub

Private Sub txtEventName\_TextChanged(sender As Object, e As EventArgs) Handles txtEventName.TextChanged

ValidateEventName()

End Sub

Private Sub txtEventVenue\_TextChanged(sender As Object, e As EventArgs) Handles txtEventVenue.TextChanged

ValidateEventVenue()

End Sub

Private Sub txtEventLevel\_TextChanged(sender As Object, e As EventArgs) Handles txtEventLevel.TextChanged

ValidateEventLevel()

End Sub

Private Sub txtEventChiefAdvisor\_TextChanged(sender As Object, e As EventArgs) Handles txtEventChiefAdvisor.TextChanged

ValidateEventChiefAdvisor()

End Sub

Private Sub txtEventAdvisor\_TextChanged(sender As Object, e As EventArgs) Handles txtEventAdvisor.TextChanged

ValidateEventAdvisor()

End Sub

Private Sub txtRemark\_TextChanged(sender As Object, e As EventArgs) Handles txtRemark.TextChanged

ValidateRemark()

End Sub

Sub ValidateEventName()

'Event name cannot be too long

Dim length As Integer = txtEventName.Text.Length

If length > 30 Then

lblError.Text = "The event name is too long! (>30 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

End Sub

Sub ValidateEventVenue()

'Event venue cannot be too long

Dim length As Integer = txtEventVenue.Text.Length

If length > 30 Then

lblError.Text = "The event venue is too long! (>30 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

End Sub

Sub ValidateEventLevel()

'Event level cannot be too long, numbers are not allowed

Dim length As Integer = txtEventLevel.Text.Length

Dim i As Integer = 0

If length > 20 Then

lblError.Text = "The event level is too long! (>20 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtEventLevel.Text.Substring(i, 1)) >= 48 And Asc(txtEventLevel.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Event level cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub ValidateEventChiefAdvisor()

'Event chief advisor cannot be too long, numbers are not allowed

Dim length As Integer = txtEventChiefAdvisor.Text.Length

Dim i As Integer = 0

If length > 50 Then

lblError.Text = "The event chief advisor is too long! (>50 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtEventChiefAdvisor.Text.Substring(i, 1)) >= 48 And Asc(txtEventChiefAdvisor.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Event chief advisor cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub ValidateEventAdvisor()

'Event advisor cannot be too long, numbers are not allowed

Dim length As Integer = txtEventAdvisor.Text.Length

Dim i As Integer = 0

If length > 50 Then

lblError.Text = "The event advisor is too long! (>50 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtEventAdvisor.Text.Substring(i, 1)) >= 48 And Asc(txtEventAdvisor.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Event advisor cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub ValidateRemark()

'Remark cannot be too long

Dim length As Integer = txtRemark.Text.Length

If length > 100 Then

lblError.Text = "The remark is too long! (>100 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

End Sub

Sub BlankField()

'To ensure certain fields are not blank

If txtEventID.Text = "" Then

lblBlank.Text = "Event ID cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtEventName.Text = "" Then

lblBlank.Text = "Event name cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtEventVenue.Text = "" Then

lblBlank.Text = "Event venue cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtEventLevel.Text = "" Then

lblBlank.Text = "Event level cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtEventChiefAdvisor.Text = "" Then

lblBlank.Text = "Event chief advisor cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtEventAdvisor.Text = "" Then

lblBlank.Text = "Event advisor cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

End Sub

Private Sub frmEventRecord\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'On loading, a connection is opened to connect VB and MS Access

'Displays simple instructions when pointed

With ToolTip1

.SetToolTip(txtEventID, "Enter the event ID (e.g. EV1029)")

.SetToolTip(txtEventName, "Enter the name of event (e.g. Mr Tack Tan)")

.SetToolTip(txtEventVenue, "Enter the venue of event (e.g. School hall)")

.SetToolTip(dtpEventDate, "Select the date of event (e.g. 10/10/2014)")

.SetToolTip(txtEventLevel, "Enter the level of event (e.g. International)")

.SetToolTip(txtEventChiefAdvisor, "Enter the name of chief advisor of the event (e.g. Mrs Amirah Haida bt Ahmad Azmi)")

.SetToolTip(txtEventAdvisor, "Enter the name of advisor of the event (e.g. Mrs Lee Yin Man)")

.SetToolTip(txtRemark, "Enter the remark of this event (e.g. time, duration and detail of the event)")

.SetToolTip(btnClear, "Clears all the fields under the event heading")

.SetToolTip(btnInsert, "Creates a new event record in the database")

.SetToolTip(btnInsert, "Updates an existing event record in the database")

.SetToolTip(btnSearch, "Searches for a specific event in the database")

.SetToolTip(btnDelete, "Deletes an event record from database")

.SetToolTip(btnMainMenu, "Go to the main menu")

.SetToolTip(lstMemberList, "List of members who were absent in an event")

.SetToolTip(lstPresentList, "List of members who were present in an event")

End With

ClearFields()

FocusCursor()

GenerateEventID() 'The event ID is loaded automatically

ClearListBoxes()

LoadMemberList() 'The member list is loaded automatically

lblError.Visible = False 'Makes the error label invisible to the user when this form is loaded

lblBlank.Visible = False 'Makes the blank label invisible to the user when this form is loaded

End Sub

Sub InsertNewEvent()

'Adds new event record into database

StartConnection()

Dim sql As String = "INSERT INTO tblEventRecord([Event ID], [Event Name], [Event Venue], [Event Date], [Event Level], " & \_

"[Event Chief Advisor], [Event Advisor], [Remark]) " & \_

"VALUES (?, ?, ?, ?, ?, ?, ?, ?)"

Dim cmd As New OleDbCommand(sql, myConnection)

With cmd.Parameters

.Add(New OleDbParameter("[Event ID]", CType(txtEventID.Text, Integer)))

.Add(New OleDbParameter("[Event Name]", CType(txtEventName.Text, String)))

.Add(New OleDbParameter("[Event Venue]", CType(txtEventVenue.Text, String)))

.Add(New OleDbParameter("[Event Date]", CType(dtpEventDate.Text, Date)))

.Add(New OleDbParameter("[Event Level]", CType(txtEventLevel.Text, String)))

.Add(New OleDbParameter("[Event Chief Advisor]", CType(txtEventChiefAdvisor.Text, String)))

.Add(New OleDbParameter("[Event Advisor]", CType(txtEventAdvisor.Text, String)))

.Add(New OleDbParameter("[Remark]", CType(txtRemark.Text, String)))

End With

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CloseConnection()

ClearFields()

FocusCursor()

GenerateEventID()

ClearListBoxes()

LoadMemberList()

End Sub

Sub UpdateEvent()

'Updates a specific event record in the database

StartConnection()

Dim EventID As Integer = CInt(txtEventID.Text)

Dim sql As String = "UPDATE tblEventRecord " & \_

"SET [Event Name]='" & txtEventName.Text & "'," & \_

"[Event Venue]='" & txtEventVenue.Text & "'," & \_

"[Event Date]='" & dtpEventDate.Text & "'," & \_

"[Event Level]='" & txtEventLevel.Text & "'," & \_

"[Event Chief Advisor]='" & txtEventChiefAdvisor.Text & "'," & \_

"[Event Advisor]='" & txtEventAdvisor.Text & "'," & \_

"[Remark]='" & txtRemark.Text & "' " & \_

"WHERE [Event ID]=" & EventID

Dim cmd As New OleDbCommand(sql, myConnection)

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CloseConnection()

ClearFields()

FocusCursor()

GenerateEventID()

ClearListBoxes()

LoadMemberList()

End Sub

Sub SearchEvent()

'Searches for a specific event in the database through its ID and displays its information

StartConnection()

'Pops out an input box to enter the specific event's ID

Dim EventID As Integer = CInt(InputBox("Enter Event ID: ", "Search Event"))

Dim sql As String = "SELECT \* FROM tblEventRecord " & \_

"WHERE [Event ID]=" & EventID

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

'Fills up the information about the event

While dr.Read()

txtEventID.Text = dr("Event ID").ToString

txtEventName.Text = dr("Event Name").ToString

txtEventVenue.Text = dr("Event Venue").ToString

dtpEventDate.Text = dr("Event Date").ToString

txtEventLevel.Text = dr("Event Level").ToString

txtEventChiefAdvisor.Text = dr("Event Chief Advisor").ToString

txtEventAdvisor.Text = dr("Event Advisor").ToString

txtRemark.Text = dr("Remark").ToString

searchFlag = True

End While

If searchFlag = False Then

MsgBox("Event ID " & EventID & " not found!", MsgBoxStyle.OkOnly + MsgBoxStyle.Exclamation, "Error")

End If

CloseConnection()

End Sub

Sub DeleteEvent()

'Deletes an event record from the database based on the event ID

StartConnection()

Dim EventID As Integer = CInt(txtEventID.Text)

Dim sql As String = "SELECT tblEventRecord.[Event ID] FROM tblEventRecord INNER JOIN tblEventAttendance ON " & \_

"tblEventRecord.[Event ID] = tblEventAttendance.[Event ID]" & \_

"WHERE (((tblEventRecord.[Event ID])=" & EventID & "))"

Dim cmd As New OleDbCommand(sql, myConnection)

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CloseConnection()

ClearFields()

FocusCursor()

GenerateEventID()

ClearListBoxes()

LoadMemberList()

End Sub

Private Sub btnInsert\_Click(sender As Object, e As EventArgs) Handles btnInsert.Click

'Saves event details into database after validation

BlankField()

If validationFlag = True Then

InsertNewEvent()

MsgBox("Event record added successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

End If

End Sub

Private Sub btnUpdate\_Click(sender As Object, e As EventArgs) Handles btnUpdate.Click

'Updates event details in database after validation

BlankField()

If validationFlag = True Then

UpdateEvent()

MsgBox("Event record updated successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

End If

End Sub

Private Sub btnSearch\_Click(sender As Object, e As EventArgs) Handles btnSearch.Click

SearchEvent()

'If the event exists in the database then searches for the attendance of that event

If searchFlag = True Then

SearchAttendance()

End If

End Sub

Private Sub btnClear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click

ClearFields()

FocusCursor()

GenerateEventID()

End Sub

Private Sub btnDelete\_Click(sender As Object, e As EventArgs) Handles btnDelete.Click

If txtEventID.Text = "" Then

MsgBox("Please enter the event ID whose record is to be deleted!", MsgBoxStyle.OkOnly + MsgBoxStyle.Exclamation, "Error")

Exit Sub

End If

'Prompts out a message box to confirm deletion, deletes the specific event record if deletion is confirmed

If MsgBox("Are you sure you want to delete this record?", \_

MsgBoxStyle.YesNo + MsgBoxStyle.Critical, "Confirmation") = MsgBoxResult.Yes Then

DeleteEvent()

MsgBox("Event record deleted successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

Else

Exit Sub

End If

End Sub

Private Sub btnMainMenu\_Click(sender As Object, e As EventArgs) Handles btnMainMenu.Click

'Closes the event record form and opens main menu

Me.Hide()

frmMainMenu.Show()

End Sub

Sub ClearListBoxes()

'clear member and present lists

lstMemberList.Items.Clear()

lstPresentList.Items.Clear()

End Sub

Sub LoadMemberList()

'Loads a list of member IDs and their names

'To allow attendance to be taken

StartConnection()

Dim sql As String = "SELECT tblMemberRecord.[Member ID], tblMemberRecord.[Member Name], tblMemberRecord.Condition " & \_

"FROM tblMemberRecord WHERE(((tblMemberRecord.Condition) = 'Active')) ORDER BY tblMemberRecord.[Member ID]"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

'Fills up list box with the names and IDs of the members

While dr.Read()

lstMemberList.Items.Add(dr("Member ID").ToString & " " & \_

dr("Member Name").ToString)

End While

CloseConnection()

End Sub

Sub InsertNewAttendance()

'Adds attendances of an event into database

StartConnection()

Dim sql As String

Dim i As Integer = 0

Dim PresentMemberID As Integer

Dim AbsentMemberID As Integer

Dim Present As String = "1" 'Sets the value for presence

Dim Absent As String = "" 'Sets the value for absence

For i = 0 To lstPresentList.Items.Count - 1

'Extract the member IDs from the list box

PresentMemberID = CInt(lstPresentList.Items.Item(i).ToString.Substring(0, 5))

sql = "INSERT INTO tblEventAttendance([Event ID], [Member ID], [Attendance]) VALUES (?, ?, ?)"

Dim cmd As New OleDbCommand(sql, myConnection)

With cmd.Parameters

.Add(New OleDbParameter("[Event ID]", CType(txtEventID.Text, Integer)))

.Add(New OleDbParameter("[Member ID]", CType(PresentMemberID, Integer)))

.Add(New OleDbParameter("[Attendance]", CType(Present, String)))

End With

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

Next

For i = 0 To lstMemberList.Items.Count - 1

'Extract the member IDs from the list box

AbsentMemberID = CInt(lstMemberList.Items.Item(i).ToString.Substring(0, 5))

sql = "INSERT INTO tblEventAttendance([Event ID], [Member ID], [Attendance]) VALUES (?, ?, ?)"

Dim cmd As New OleDbCommand(sql, myConnection)

With cmd.Parameters

.Add(New OleDbParameter("[Event ID]", CType(txtEventID.Text, Integer)))

.Add(New OleDbParameter("[Member ID]", CType(AbsentMemberID, Integer)))

.Add(New OleDbParameter("[Attendance]", CType(Absent, String)))

End With

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

Next

CloseConnection()

ClearFields()

FocusCursor()

GenerateEventID()

ClearListBoxes()

LoadMemberList()

End Sub

Sub SearchAttendance()

'Searches a specific event's attendance in the database through its ID and displays its information

ClearListBoxes()

LoadMemberList()

StartConnection()

Dim EventID As Integer = CInt(txtEventID.Text)

Dim sql As String = "SELECT tblEventAttendance.[Event ID], tblEventAttendance.[Member ID], tblEventAttendance.Attendance, " & \_

"tblMemberRecord.[Member Name] FROM tblMemberRecord INNER JOIN tblEventAttendance " & \_

"ON tblMemberRecord.[Member ID] = tblEventAttendance.[Member ID]" & \_

"WHERE (((tblEventAttendance.[Event ID])=" & EventID & \_

")) ORDER BY tblEventAttendance.[Member ID]"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

'If a member was present in the event, then move the member to the present list

While dr.Read()

If dr("Attendance").ToString = "1" Then

lstPresentList.Items.Add(dr("Member ID").ToString & " " & dr("Member Name").ToString)

lstMemberList.Items.Remove(dr("Member ID").ToString & " " & dr("Member Name").ToString)

End If

End While

CloseConnection()

End Sub

Private Sub btnMove\_Click(sender As Object, e As EventArgs) Handles btnMove.Click

'Moves selected item from one list box to another

If lstMemberList.SelectedIndex <> -1 Then

lstPresentList.Items.Add(lstMemberList.SelectedItem)

lstMemberList.Items.Remove(lstMemberList.SelectedItem)

End If

If lstPresentList.SelectedIndex <> -1 Then

lstMemberList.Items.Add(lstPresentList.SelectedItem)

lstPresentList.Items.Remove(lstPresentList.SelectedItem)

End If

End Sub

Private Sub btnSave\_Click(sender As Object, e As EventArgs) Handles btnSave.Click

'Prompts out a message box to confirm insertion of attendances into database, adds attendances of the event if confirmed

If MsgBox("Please check the data again. Are you sure you want to insert this attendance list?", \_

MsgBoxStyle.YesNo + MsgBoxStyle.Information, "Confirmation") = MsgBoxResult.Yes Then

InsertNewAttendance()

MsgBox("The attendance list has been added successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

Else

Exit Sub

End If

End Sub

Private Sub lnkHelp\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkHelp.LinkClicked

'Opens the index of help page

System.Diagnostics.Process.Start("Help\Index.htm")

End Sub

End Class

Committee Board Form

Imports System.Data

Imports System.Data.OleDb

Public Class frmCommitteeBoard

Dim provider As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source ="

Dim dataFile As String = "Beserah Primary School English Language Society Management System.accdb"

Dim connString As String = provider & dataFile 'Path of database file

Dim myConnection As New OleDbConnection 'Joins provider and path

Dim dr As OleDbDataReader 'Data reader to read data from Access Database

Dim validationFlag As Boolean = True

Dim searchFlag As Boolean = False

Sub StartConnection()

'Connects the programme with the database

myConnection.ConnectionString = connString

myConnection.Open()

End Sub

Sub CloseConnection()

'Closes the connection between the programme and the database

myConnection.Close()

End Sub

Sub GenerateYear()

'Automatically generates the year by taking the year of the input day

txtYear.Text = Date.Now.Year

End Sub

Sub ClearFieldsOtherThanYear()

'Clears all the fields except the year

txtChiefAdvisor.Text = ""

txtAdvisor1.Text = ""

txtAdvisor2.Text = ""

txtAdvisor3.Text = ""

txtChairperson.Text = ""

txtViceChairperson.Text = ""

txtSecretary.Text = ""

txtTreasurer.Text = ""

End Sub

Sub FocusCursor()

'Focus cursor at the chief advisor text box

txtChiefAdvisor.Focus()

End Sub

Private Sub txtChiefAdvisor\_TextChanged(sender As Object, e As EventArgs) Handles txtChiefAdvisor.TextChanged

ValidateChiefAdvisor()

End Sub

Private Sub txtAdvisor1\_TextChanged(sender As Object, e As EventArgs) Handles txtAdvisor1.TextChanged

ValidateAdvisor1()

End Sub

Private Sub txtAdvisor2\_TextChanged(sender As Object, e As EventArgs) Handles txtAdvisor2.TextChanged

ValidateAdvisor2()

End Sub

Private Sub txtAdvisor3\_TextChanged(sender As Object, e As EventArgs) Handles txtAdvisor3.TextChanged

ValidateAdvisor3()

End Sub

Private Sub txtYear\_TextChanged(sender As Object, e As EventArgs) Handles txtYear.TextChanged

ValidateYear()

End Sub

Sub ValidateChiefAdvisor()

'Chief advisor name cannot be too long, numbers are not allowed

Dim length As Integer = txtChiefAdvisor.Text.Length

Dim i As Integer = 0

If length - 1 > 50 Then

lblError.Text = "The chief advisor is too long! (>50 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtChiefAdvisor.Text.Substring(i, 1)) >= 48 And Asc(txtChiefAdvisor.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Chief advisor cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub ValidateAdvisor1()

'Advisor 1 name cannot be too long, numbers are not allowed

Dim length As Integer = txtAdvisor1.Text.Length

Dim i As Integer = 0

If length - 1 > 50 Then

lblError.Text = "The advisor 1 is too long! (>50 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtAdvisor1.Text.Substring(i, 1)) >= 48 And Asc(txtAdvisor1.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Advisor 1 cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub ValidateAdvisor2()

'Advisor 2 name cannot be too long, numbers are not allowed

Dim length As Integer = txtAdvisor2.Text.Length

Dim i As Integer = 0

If length - 1 > 50 Then

lblError.Text = "The advisor 2 is too long! (>50 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtAdvisor2.Text.Substring(i, 1)) >= 48 And Asc(txtAdvisor2.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Advisor 2 cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub ValidateAdvisor3()

'Advisor 3 name cannot be too long, numbers are not allowed

Dim length As Integer = txtAdvisor3.Text.Length

Dim i As Integer = 0

If length - 1 > 50 Then

lblError.Text = "The advisor 3 is too long! (>50 characters)"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

For i = 0 To length - 1

If Asc(txtAdvisor3.Text.Substring(i, 1)) >= 48 And Asc(txtAdvisor3.Text.Substring(i, 1)) <= 57 Then

lblError.Text = "Advisor 3 cannot contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

Next

End Sub

Sub ValidateYear()

'Must be a valid, 4 digit year

Dim length As Integer = txtYear.Text.Length

Dim i As Integer = 0

If (length <> 4) Then

lblError.Text = "Year can only contain 4 digits!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

If IsNumeric(txtYear.Text) = False Then

lblError.Text = "Year must contain numbers!"

lblError.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblError.Visible = False

End If

End Sub

Sub BlankField()

'To ensure certain fields are not blank

If txtYear.Text = "" Then

lblBlank.Text = "Year cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtChiefAdvisor.Text = "" Then

lblBlank.Text = "Chief advisor cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtAdvisor1.Text = "" Then

lblBlank.Text = "Advisor 1 cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtAdvisor2.Text = "" Then

lblBlank.Text = "Advisor 2 cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

If txtAdvisor3.Text = "" Then

lblBlank.Text = "Advisor 3 cannot be blank!"

lblBlank.Visible = True

validationFlag = False

Exit Sub

Else

validationFlag = True

lblBlank.Visible = False

End If

End Sub

Private Sub frmCommitteeMembers\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

GenerateYear()

ClearFieldsOtherThanYear()

FocusCursor()

'The text boxes and error label are hidden from view when this form is loaded

txtChairperson.Visible = False

txtViceChairperson.Visible = False

txtSecretary.Visible = False

txtTreasurer.Visible = False

lblError.Text = ""

lblBlank.Text = ""

End Sub

Sub Chairperson()

'Loads chairperson's name from the database

StartConnection()

Dim sql As String = "SELECT [Member Name] " & \_

"FROM tblMemberRecord " & \_

"WHERE (([Condition]='Active') AND ([Position]='Chairperson'))"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

While dr.Read()

txtChairperson.Text = dr("Member Name").ToString

End While

CloseConnection()

End Sub

Sub ViceChairperson()

'Loads vice chairperson's name from the database

StartConnection()

Dim sql As String = "SELECT [Member Name] " & \_

"FROM tblMemberRecord " & \_

"WHERE (([Condition]='Active') AND ([Position]='Vice Chairperson'))"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

While dr.Read()

txtViceChairperson.Text = dr("Member Name").ToString

End While

CloseConnection()

End Sub

Sub Secretary()

'Loads secretary's name from the database

StartConnection()

Dim sql As String = "SELECT [Member Name] " & \_

"FROM tblMemberRecord " & \_

"WHERE (([Condition]='Active') AND ([Position]='Secretary'))"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

While dr.Read()

txtSecretary.Text = dr("Member Name").ToString

End While

CloseConnection()

End Sub

Sub Treasurer()

'Loads treasurer's name from the database

StartConnection()

Dim sql As String = "SELECT [Member Name] " & \_

"FROM tblMemberRecord " & \_

"WHERE (([Condition]='Active') AND ([Position]='Treasurer'))"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

While dr.Read()

txtTreasurer.Text = dr("Member Name").ToString

End While

CloseConnection()

End Sub

Sub SaveCommitteeBoard()

'Adds committee members' records into database

StartConnection()

Dim sql As String = "INSERT INTO tblCommitteeBoard([Chief Advisor], [Advisor 1], [Advisor 2], [Advisor 3], [Year], " & \_

"[Chairperson], [Vice Chairperson], [Secretary], [Treasurer]) " & \_

"VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)"

Dim cmd As New OleDbCommand(sql, myConnection)

With cmd.Parameters

.Add(New OleDbParameter("[Chief Advisor]", CType(txtChiefAdvisor.Text, String)))

.Add(New OleDbParameter("[Advisor 1]", CType(txtAdvisor1.Text, String)))

.Add(New OleDbParameter("[Advisor 2]", CType(txtAdvisor2.Text, String)))

.Add(New OleDbParameter("[Advisor 3]", CType(txtAdvisor3.Text, String)))

.Add(New OleDbParameter("[Year]", CType(txtYear.Text, Integer)))

.Add(New OleDbParameter("[Chairperson]", CType(txtChairperson.Text, String)))

.Add(New OleDbParameter("[Vice Chairperson]", CType(txtViceChairperson.Text, String)))

.Add(New OleDbParameter("[Secretary]", CType(txtSecretary.Text, String)))

.Add(New OleDbParameter("[Treasurer]", CType(txtTreasurer.Text, String)))

End With

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CloseConnection()

ClearFieldsOtherThanYear()

FocusCursor()

txtYear.Text = ""

GenerateYear()

lblError.Text = ""

lblBlank.Text = ""

End Sub

Sub SearchCommitteeBoard()

'Searches a specific year's committee board in the database

ClearFieldsOtherThanYear()

txtYear.Text = ""

StartConnection()

'Pops out an input box to enter the specific year

Dim Year As Integer = CInt(InputBox("Enter year: ", "Search Committee Board"))

Dim sql As String = "SELECT \* FROM tblCommitteeBoard " & \_

"WHERE [Year]=" & Year

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

'Fills up the information of the committee board

While dr.Read()

txtYear.Text = Year

txtChiefAdvisor.Text = dr("Chief Advisor").ToString

txtAdvisor1.Text = dr("Advisor 1").ToString

txtAdvisor2.Text = dr("Advisor 2").ToString

txtAdvisor3.Text = dr("Advisor 3").ToString

txtChairperson.Text = dr("Chairperson").ToString

txtViceChairperson.Text = dr("Vice Chairperson").ToString

txtSecretary.Text = dr("Secretary").ToString

txtTreasurer.Text = dr("Treasurer").ToString

searchFlag = True

End While

If searchFlag = False Then

MsgBox("Year " & Year & " not found!", MsgBoxStyle.OkOnly + MsgBoxStyle.Exclamation, "Error")

End If

CloseConnection()

End Sub

Sub UpdateCommitteeBoard()

'Updates a year's committee member record in the database

StartConnection()

Dim Year As Integer = CInt(txtYear.Text)

Dim sql As String = "UPDATE tblCommitteeBoard " & \_

"SET [Chief Advisor]='" & txtChiefAdvisor.Text & "'," & \_

"[Advisor 1]='" & txtAdvisor1.Text & "'," & \_

"[Advisor 2]='" & txtAdvisor2.Text & "'," & \_

"[Advisor 3]='" & txtAdvisor3.Text & "' " & \_

"WHERE [Year]=" & Year

Dim cmd As New OleDbCommand(sql, myConnection)

Try

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message)

End Try

CloseConnection()

ClearFieldsOtherThanYear()

FocusCursor()

txtYear.Text = ""

GenerateYear()

lblError.Text = ""

lblBlank.Text = ""

End Sub

Private Sub lnkPrint\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkPrint.LinkClicked

'Print preview with the link labels, buttons and error labels hidden

lnkPrint.Hide()

lnkHelp.Hide()

btnMainMenu.Hide()

btnSave.Hide()

btnSearch.Hide()

btnUpdate.Hide()

btnClear.Hide()

lblBlank.Hide()

lblError.Hide()

prtCommitteeBoard.PrintAction = Printing.PrintAction.PrintToPreview

'Print action

prtCommitteeBoard.Print()

'Allows the link labels, buttons and error labels to show again

lnkPrint.Show()

lnkHelp.Show()

btnMainMenu.Show()

btnSave.Show()

btnSearch.Show()

btnUpdate.Show()

btnClear.Show()

lblBlank.Show()

lblError.Show()

End Sub

Private Sub lnkHelp\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkHelp.LinkClicked

'Opens the index of help page

System.Diagnostics.Process.Start("Help\Index.htm")

End Sub

Private Sub btnSave\_Click(sender As Object, e As EventArgs) Handles btnSave.Click

'Saves committee board details into database after validation

BlankField()

If validationFlag = True Then

Chairperson()

ViceChairperson()

Secretary()

Treasurer()

SaveCommitteeBoard()

MsgBox("Committee board added successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

End If

End Sub

Private Sub btnSearch\_Click(sender As Object, e As EventArgs) Handles btnSearch.Click

SearchCommitteeBoard()

'Allows the contents of text boxes to be displayed

txtChairperson.Visible = True

txtViceChairperson.Visible = True

txtSecretary.Visible = True

txtTreasurer.Visible = True

lblError.Text = ""

lblBlank.Text = ""

End Sub

Private Sub btnMainMenu\_Click(sender As Object, e As EventArgs) Handles btnMainMenu.Click

'Closes the committee board form and opens the main menu

Me.Hide()

frmMainMenu.Show()

End Sub

Private Sub btnClear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click

ClearFieldsOtherThanYear()

FocusCursor()

txtYear.Text = ""

GenerateYear()

lblError.Text = ""

lblBlank.Text = ""

End Sub

Private Sub btnUpdate\_Click(sender As Object, e As EventArgs) Handles btnUpdate.Click

'Updates committee board details into database after validation

BlankField()

If validationFlag = True Then

Chairperson()

ViceChairperson()

Secretary()

Treasurer()

UpdateCommitteeBoard()

MsgBox("Committee board updated successfully!", MsgBoxStyle.OkOnly + MsgBoxStyle.Information, "Success")

End If

End Sub

End Class

List of Members Present Form

Imports System.Data

Imports System.Data.OleDb

Public Class frmPresentList

Dim provider As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source ="

Dim dataFile As String = "Beserah Primary School English Language Society Management System.accdb"

Dim connString As String = provider & dataFile 'Path of database file

Dim myConnection As New OleDbConnection 'Joins provider and path

Dim dr As OleDbDataReader 'Data reader to read data from Access Database

Dim validationFlag As Boolean = True

Dim searchFlag As Boolean = False

Dim EventID As Integer

Sub StartConnection()

'Connects the programme with the database

myConnection.ConnectionString = connString

myConnection.Open()

End Sub

Sub CloseConnection()

'Closes the connection between the programme and the database

myConnection.Close()

End Sub

Private Sub frmPresentList\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

SearchForEvent()

End Sub

Sub SearchForEvent()

'Searches for the present list of a specific event in the database through the event ID and displays its information

StartConnection()

EventID = CInt(InputBox("Event ID: ", "Search for present list of event"))

Dim sql As String = "SELECT tblEventAttendance.[Event ID], tblEventRecord.[Event Name], tblEventRecord.[Event Venue], " & \_

"tblEventRecord.[Event Date], tblEventAttendance.[Member ID], tblMemberRecord.[Member Name], " & \_

"tblEventAttendance.Attendance FROM tblMemberRecord INNER JOIN (tblEventRecord INNER JOIN " & \_

"tblEventAttendance ON tblEventRecord.[Event ID] = tblEventAttendance.[Event ID]) ON " & \_

"tblMemberRecord.[Member ID] = tblEventAttendance.[Member ID]" & \_

"WHERE (((tblEventAttendance.[Event ID])=" & EventID & ") AND ((tblEventAttendance.Attendance)='1')) " & \_

"ORDER BY tblEventAttendance.[Member ID]"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

While dr.Read()

lblEventNameAndID.Text = dr("Event Name").ToString & " ( ID: " & dr("Event ID").ToString & " )"

lblEventDate.Text = dr("Event Date").ToShortDateString

lblEventVenue.Text = dr("Event Venue").ToString

dgvPresentList.Rows.Add(dr("Member ID").ToString, dr("Member Name").ToString)

End While

CloseConnection()

End Sub

Private Sub lnkPrint\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkPrint.LinkClicked

'Print preview with the link labels and button hidden

lnkPrint.Hide()

lnkHelp.Hide()

btnMainMenu.Hide()

prtPresentList.PrintAction = Printing.PrintAction.PrintToPreview

'Print action

prtPresentList.Print()

'Allows the link labels and button to show again

lnkPrint.Show()

lnkHelp.Show()

btnMainMenu.Show()

End Sub

Private Sub lnkHelp\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkHelp.LinkClicked

'Opens the index of help page

System.Diagnostics.Process.Start("Help\Index.htm")

End Sub

Private Sub btnMainMenu\_Click(sender As Object, e As EventArgs) Handles btnMainMenu.Click

'Closes the present list form and opens the main menu

Me.Hide()

frmMainMenu.Show()

End Sub

End Class

Potential Contestant List

Imports System.Data

Imports System.Data.OleDb

Public Class frmPotentialContestantList

Dim provider As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source ="

Dim dataFile As String = "Beserah Primary School English Language Society Management System.accdb"

Dim connString As String = provider & dataFile 'Path of database file

Dim myConnection As New OleDbConnection 'Joins provider and path

Dim dr As OleDbDataReader 'Data reader to read data from Access Database

Sub StartConnection()

'Connects the programme with the database

myConnection.ConnectionString = connString

myConnection.Open()

End Sub

Sub CloseConnection()

'Closes the connection between the programme and the database

myConnection.Close()

End Sub

Private Sub frmPotentialContestant\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Searches a specific member in the database through its ID and then displays it

StartConnection()

Dim sql As String = "SELECT tblMemberRecord.[Highest English Grade], tblMemberRecord.[Potential Contestant], " & \_

"tblMemberRecord.[Member Name], tblMemberRecord.[Member ID] FROM tblMemberRecord " & \_

"WHERE (tblMemberRecord.[Potential Contestant]='Potential Contestant') " & \_

"ORDER BY tblMemberRecord.[Highest English Grade], tblMemberRecord.[Member ID]"

Dim cmd As New OleDbCommand(sql, myConnection)

dr = cmd.ExecuteReader

While dr.Read()

dgvPotentialContestantList.Rows.Add(dr("Highest English Grade").ToString, dr("Member ID").ToString, dr("Member Name").ToString)

End While

CloseConnection()

End Sub

Private Sub lnkPrint\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkPrint.LinkClicked

'Print preview with the link labels and button hidden

lnkPrint.Hide()

lnkHelp.Hide()

btnMainMenu.Hide()

prtPotentialContestant.PrintAction = Printing.PrintAction.PrintToPreview

'Print action

prtPotentialContestant.Print()

'Allows the link labels and button to show again

lnkPrint.Show()

lnkHelp.Show()

btnMainMenu.Show()

End Sub

Private Sub lnkHelp\_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles lnkHelp.LinkClicked

'Opens the index of help page

System.Diagnostics.Process.Start("Help\Index.htm")

End Sub

Private Sub btnMainMenu\_Click(sender As Object, e As EventArgs) Handles btnMainMenu.Click

'Closes the potential contestant list form and opens main menu

Me.Hide()

frmMainMenu.Show()

End Sub

End Class

1. **Testing**

**Testing Strategy**

Testing is essential in order to ensure that the software functions properly and produces desired outputs. Any problem that arises when the software fails to function as we wanted is caused by bugs. Therefore, testing is carried out in the development stage of the system to reduce, minimise and eliminate, if possible, the bugs that cause errors to exist in the software. Thus, the main functions of the software can be achieved.

This system is mainly used to store and search for data besides to produce certain outputs where data is filtered. Hence, the main functions to be tested would be the member and event registration part. All data are to be validated before they are inputted into the system in order to maintain the consistency of data of different fields. If any data is in the wrong format or length, validation texts will be generated in the error labels in order to guide the user to correct them to the approved form so that they can be processed smoothly by the system.

The testing method used is black-box testing where the expected results and the actual results are compared. If they happen to be different, then actions will be taken to correct the errors made in the programming part. This method is chosen because it is fast and easy to perform and its outcomes are accurate. Thus, the bugs that existed can be corrected and eliminated faster.

**Testing**

Member Record Form

Validation routines for member name and guardian are similar.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test No.** | **Field** | **Test Type** | **Test Data** | **Expected Result** | **Actual Result** | **Action** | **Evidence (page)** |
| 1 | Member ID | Valid | 11029 | 11029 | 11029 | - | 137 |
| 2 | Member ID | Boundary | 99999 | 99999 | 99999 | - | 137 |
| 3 | Member ID | Invalid | 2312 | Error | Error | - | 138 |
| 4 | Condition | Valid | Active | Active | Active | - | 138 |
| 5 | Member Name | Valid | Tan Jiyen | Tan Jiyen | Tan Jiyen | - | 139 |
| 6 | Member Name | Boundary | Mohammad Amirul Khan bin Uzza Nafiz Nabirul Baderi | Mohammad Amirul Khan bin Uzza Nafiz Nabirul Baderi | Error | Changed the ‘>=’ sign to ‘>’ sign | 139, 140 |
| 7 | Member Name | Invalid | Mohammad Ali Amirul Khan bin Uza Nafiz Nabirul Baderi | Error | Error | - | 141 |
| 8 | Member Class | Valid | 4A | 4A | 4A | - | 141 |
| 9 | Member Class | Invalid |  | Error | Error | - | 142 |
| 10 | Position | Valid | None | None | None | - | 142 |
| 11 | Position | Invalid |  | Error | Error |  | 143 |
| 12 | Highest English Grade | Valid | A | A | A | - | 143 |
| 13 | Highest English Grade | Invalid |  | Error | Error | - | 144 |
| 14 | Contact Number | Valid | 0129045745 | 0129045745 | 0129045745 | - | 144 |
| 15 | Contact Number | Boundary | 01119053677 | 01119053677 | 01119053677 | - | 145 |
| 16 | Contact Number | Invalid | 018-5546775 | Error | Error | - | 145 |

Event Record Form

Validation routines for event name and event venue, event chief advisor and event advisor are the same while that of event ID and member ID are the same as that in the event record form and the member record form respectively.

Event ID does not need to be validated as it is generated automatically when this form is loaded.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test No.** | **Field** | **Test Type** | **Test Data** | **Expected Result** | **Actual Result** | **Action** | **Evidence (page)** |
| 17 | Event Name | Valid | Happy English Talk 2014 | Happy English Talk 2014 | Error | Removed the character limit from the code | 146, 147 |
| 18 | Event Name | Boundary | Year 14/15 Happy English Talk | Year 14/15 Happy English Talk | Year 14/15 Happy English Talk | - | 147 |
| 19 | Event Name | Invalid | Year 2014/2015 Happy English Talk | Error | Error | - | 148 |
| 20 | Event Date | Valid | 6/6/2015 | 6/6/2015 | 6/6/2015 | - | 148 |
| 21 | Event Level | Valid | School | School | School | - | 149 |
| 22 | Event Level | Boundary | School Zone together | School Zone together | School Zone together | - | 149 |
| 23 | Event Level | Invalid | Zone 17/09 | Error | Error | - | 150 |
| 24 | Event Chief Advisor | Valid | Tan Eng Eng | Tan Eng Eng | Tan Eng Eng | - | 150 |
| 25 | Event Chief Advisor | Boundary | Siti Aisyah Nuraini Shah binti Wakhrul Zani Osman | Siti Aisyah Nuraini Shah binti Wakhrul Zani Osman | Siti Aisyah Nuraini Shah binti Wakhrul Zani Osman | - | 151 |
| 26 | Event Chief Advisor | Invalid | Ami Rusiah Fatimah binti Gural1 | Error | Error | - | 151 |
| 27 | Remark | Valid | Held from 9 to 10 a.m. | Held from 9 to 10 a.m. | Held from 9 to 10 a.m. | - | 152 |
| 28 | Remark | Boundary | The talk was conducted by an Australian Lecturer from Cambridge University from 12 p.m. to 1.30 p.m. | The talk was conducted by an Australian Lecturer from Cambridge University from 12 p.m. to 1.30 p.m. | The talk was conducted by an Australian Lecturer from Cambridge University from 12 p.m. to 1.30 p.m. | - | 152 |
| 29 | Remark | Invalid | Talk was conducted by an Australian Lecturer from Cambridge University. It was held from 12 p.m. to 1.30 p.m. and is compulsory for all the year 4, 5 and 6 students to attend. | Error | Error | - | 153 |
| 30 | Attendance | Functional | 1 | 1 | 1 | - | 153 |

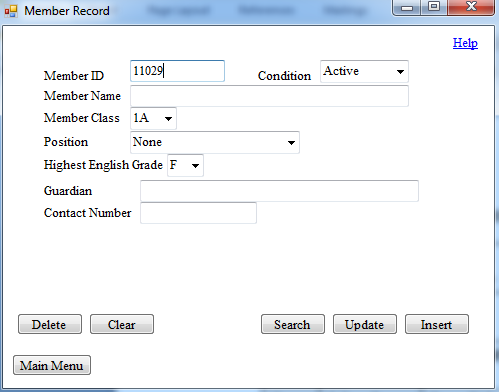
Committee Board Form

Validation routines of chief advisor, advisor 1, advisor 2 and advisor 3 are the same as those of the event chief advisor and event advisor in the event record form.

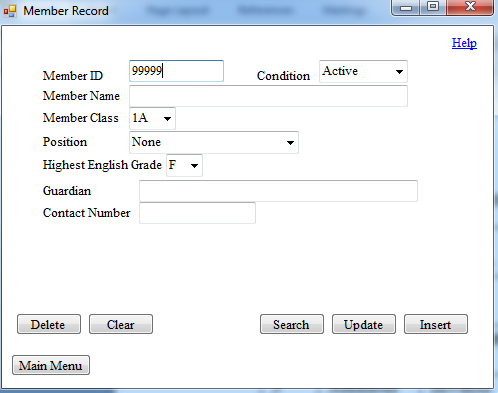
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test No.** | **Field** | **Test Type** | **Test Data** | **Expected Result** | **Actual Result** | **Action** | **Evidence (page)** |
| 31 | Year | Valid | 2014 | 2014 | 2014 | - | 154 |
| 32 | Year | Boundary | 1000 | 1000 | 1000 | - | 154 |
| 33 | Year | Invalid | 2ooo | Error | Error | - | 155 |

Evidence:

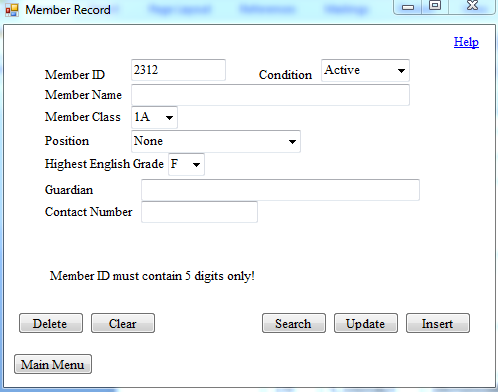
Member Record Form



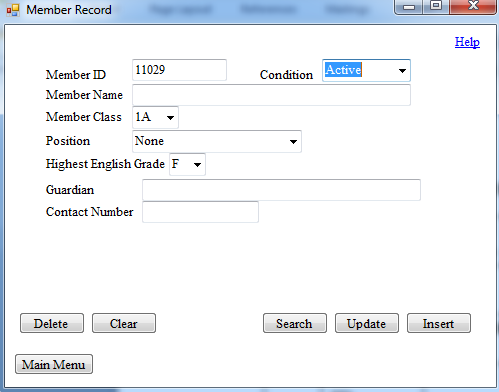
Test 1: 11029 did not produce any error message (Success)



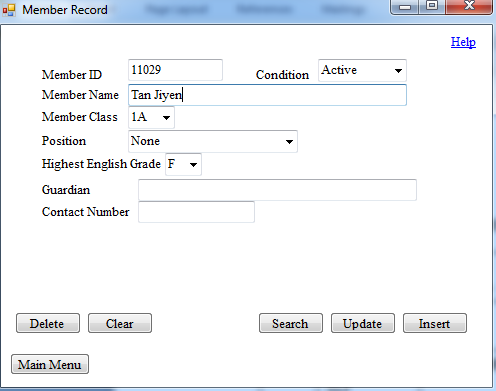
Test 2: 99999 did not produce any error message (Success)



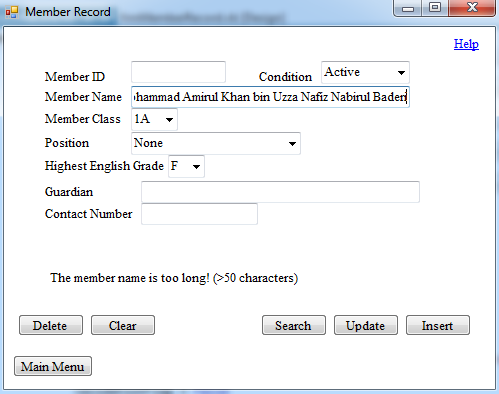
Test 3: 2312 produced an error as expected (Success)



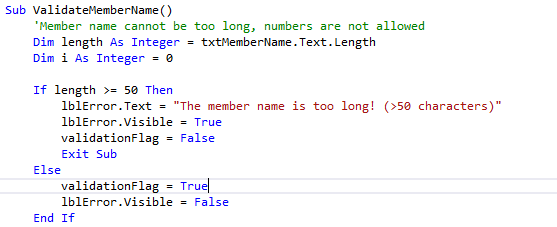
Test 4: ‘Active’ did not produce any error message (Success)



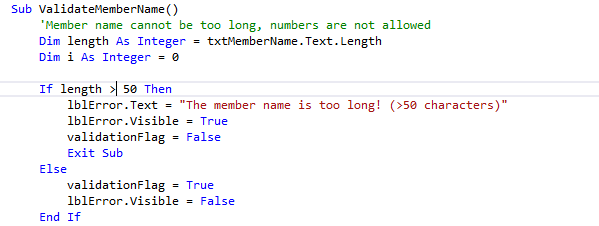
Test 5: ‘Tan Jiyen’ did not produce any error message (Success)



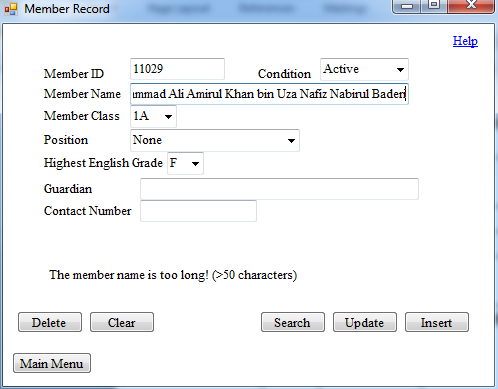
Test 6: ‘Mohammad Amirul Khan bin Uzza Nafiz Nabirul Baderi’ produced an error message (Failed)



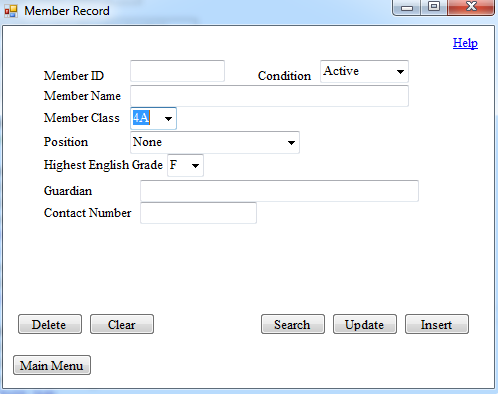
Test 6: This is the original code. This method is used in all forms to validate the length of names of people (Guardian, Event Chief Advisor, Event Advisor, Chief Advisor, Advisor 1, Advisor 2, Advisor 3)



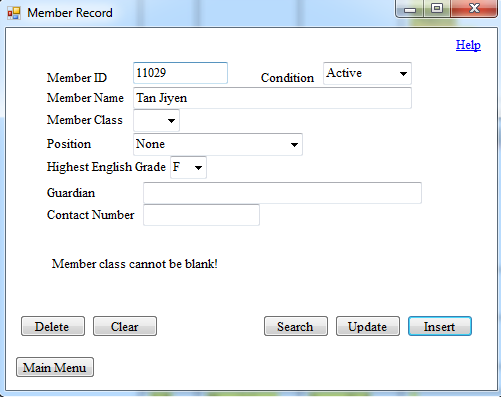
Test 6: This is the modified code which eliminates the logic error and produces error message only if the length of member name is more than 50 characters.



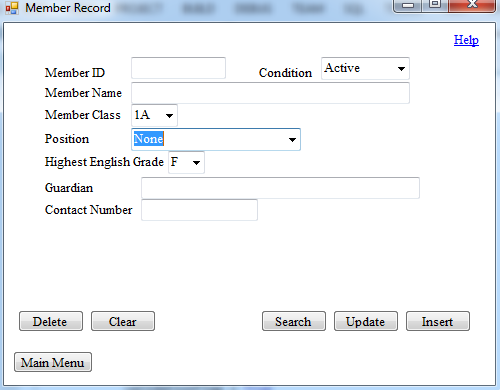
Test 7: ‘Mohammad Ali Amirul Khan bin Uza Nafiz Nabirul Baderi’ produced an error as expected (Success)



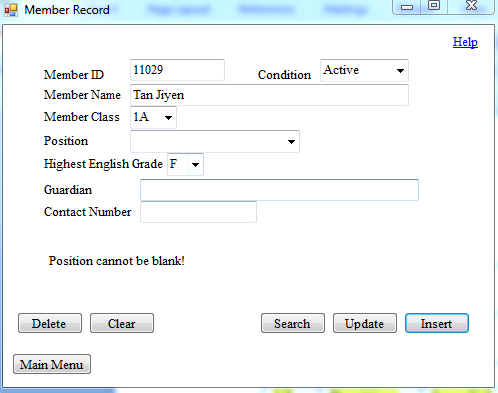
Test 8: ‘4A’ did not produce any error message (Success)



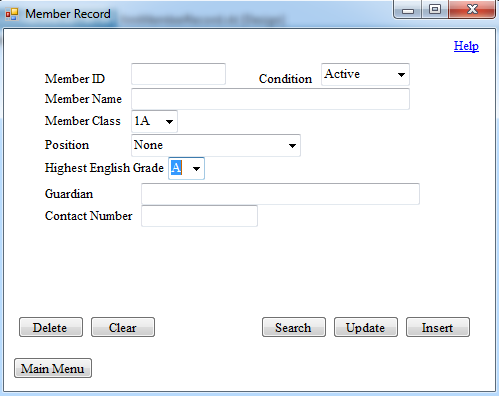
Test 9: Blanking ‘Condition’ field produced an error as expected (Success)



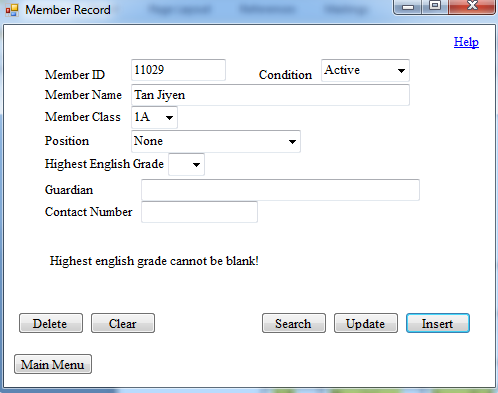
Test 10: ‘None’ did not produce any error message (Success)



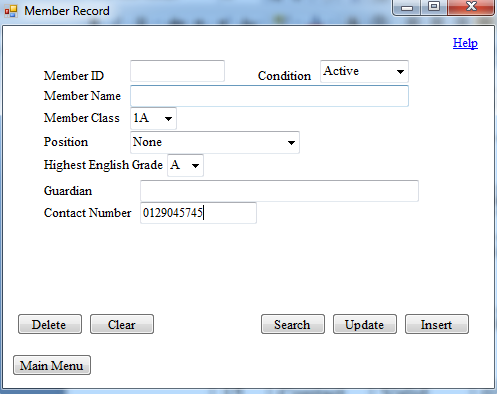
Test 11: Blanking ‘Position’ field produced an error as expected (Success)



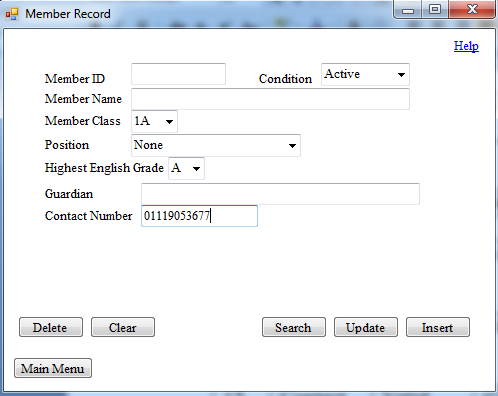
Test 12: ‘A’ did not produce any error message (Success)



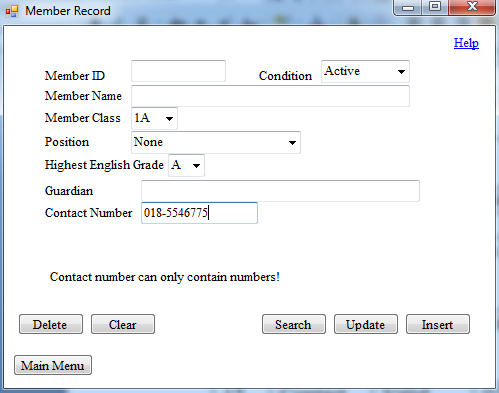
Test 13: Blanking ‘Highest English Grade’ field produced an error as expected (Success)

****

Test 14: ‘0129045745’ did not produce any error message (Success)

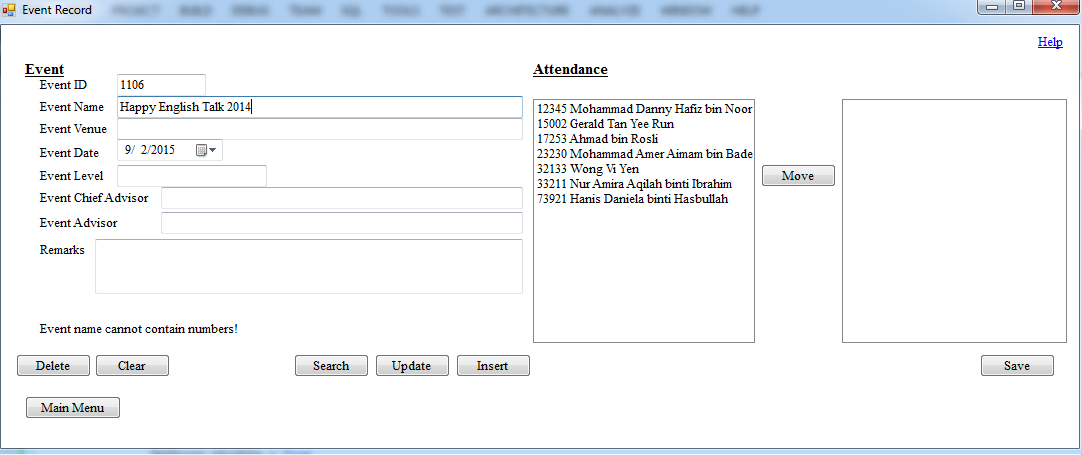
****

Test 15: ‘01119053677’ did not produce any error message (Success)

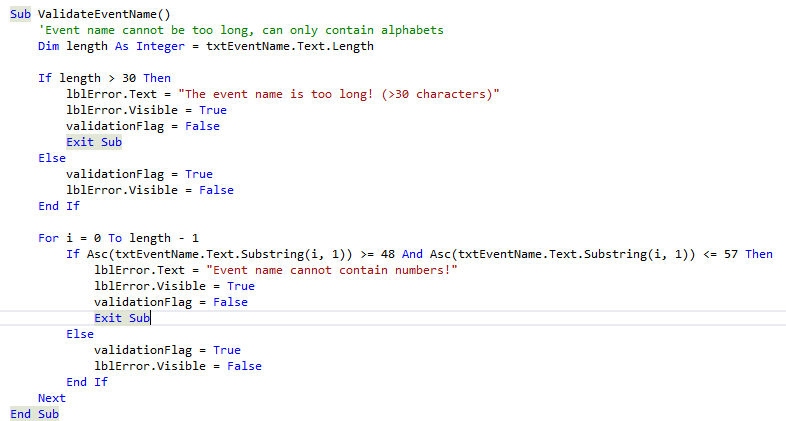


Test 16: ‘018-5546775’ produced an error as expected (Success)

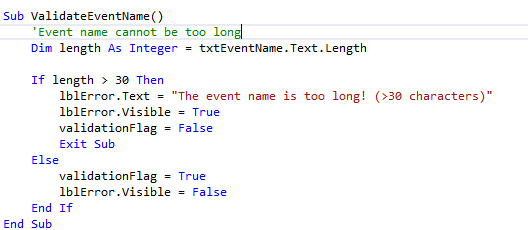
Event Record and Event Attendance Form



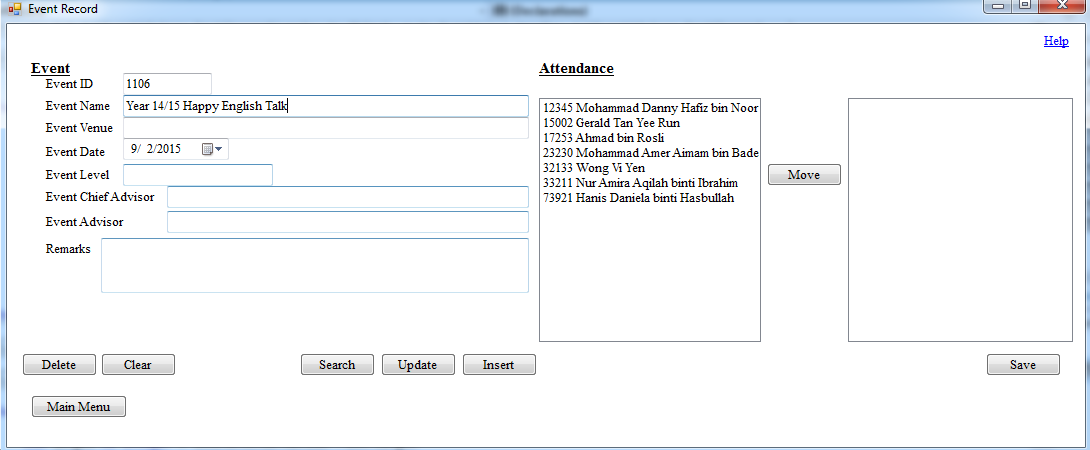
Test 17: ‘Happy English Talk 2014’ produced an error message (Failed)



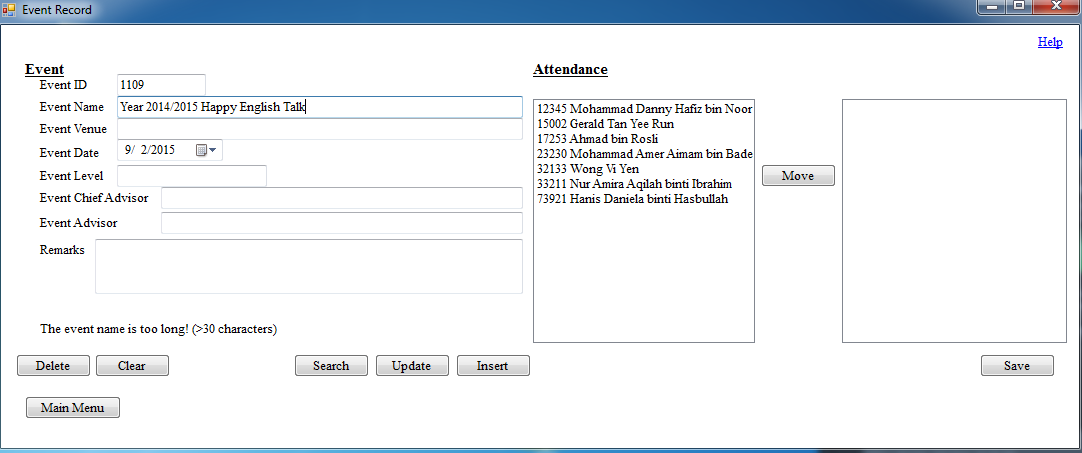
Test 17: This is the original code. This method is used in this form to validate the length of Event Name, Event Venue and Remark.



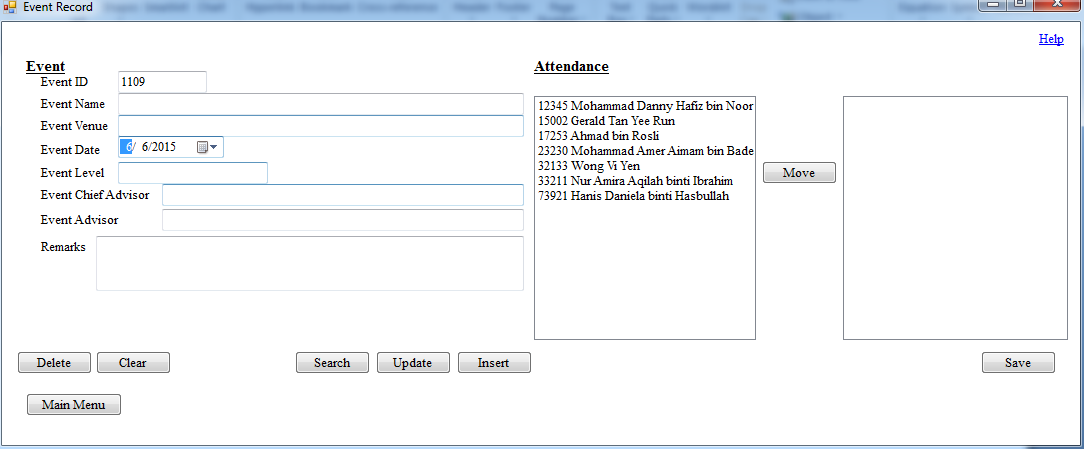
Test 17: This is the modified code which removes the character check and allows all characters (alphabets, numbers and symbols) to be entered.



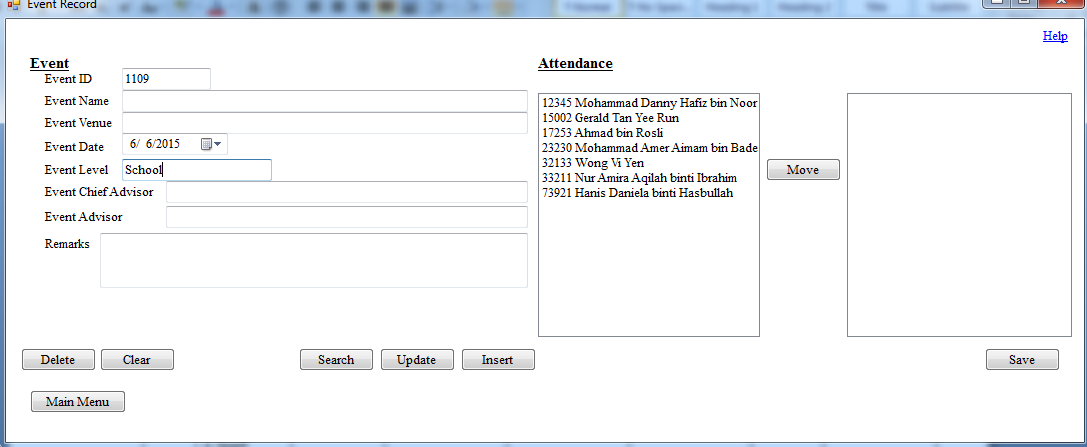
Test 18: ‘Year 14/15 Happy English Talk’ did not produce any error message (Success)



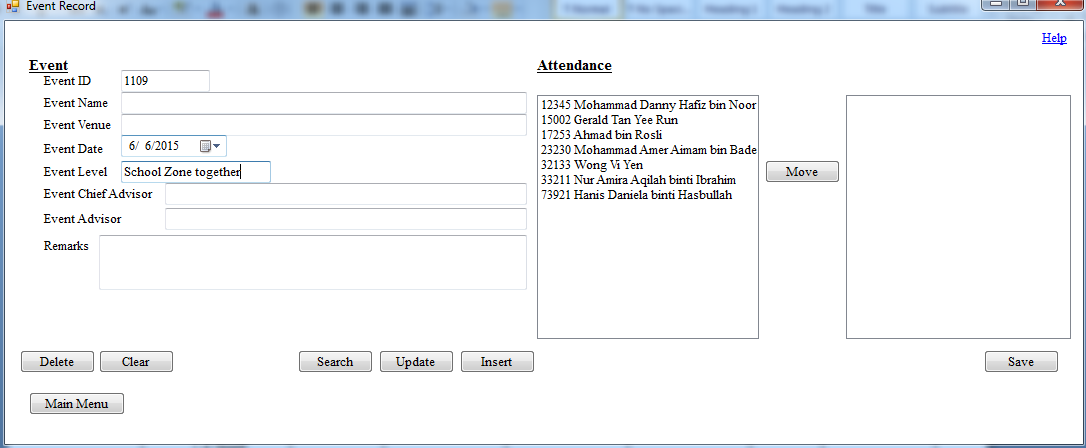
Test 19: ‘Year 2014/2015 Happy English Talk’ produced an error as expected (Success)



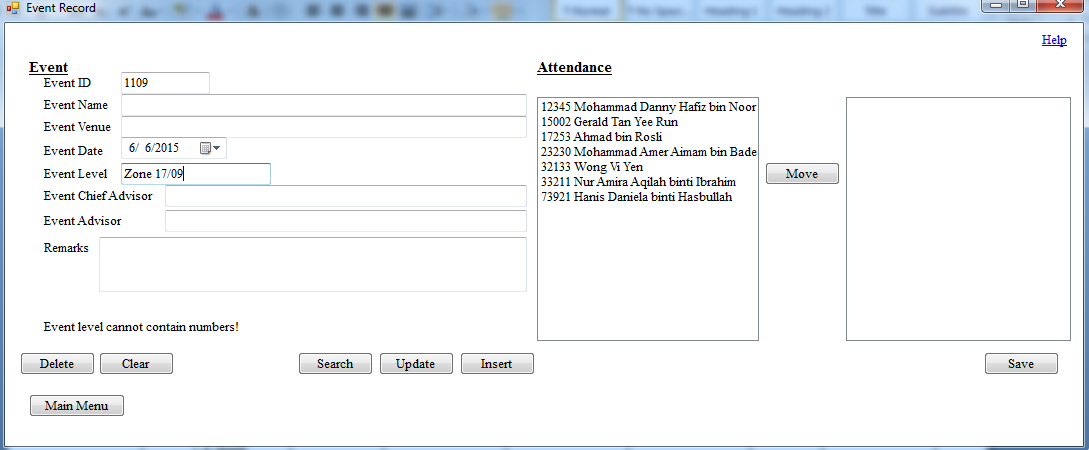
Test 20: ‘6/6/2015’ did not produce any error message (Success)



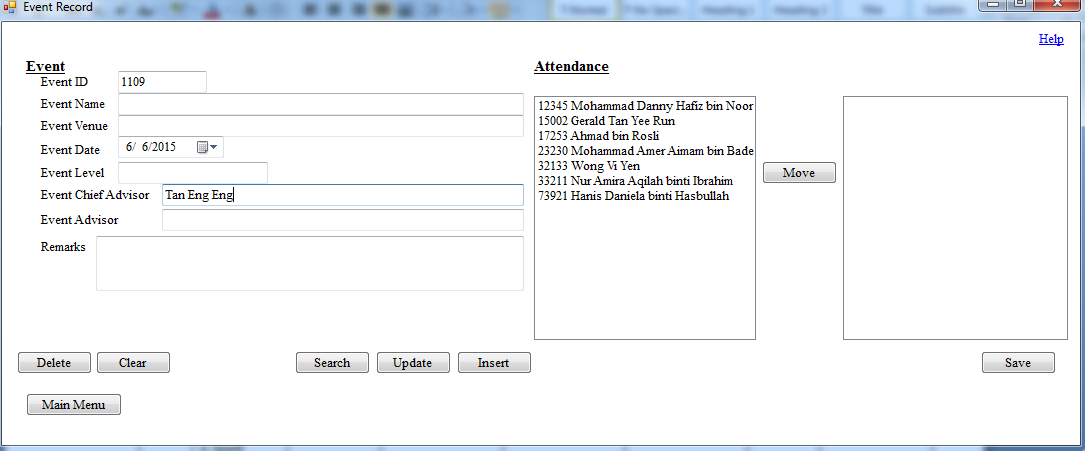
Test 21: ‘School’ did not produce any error message (Success)



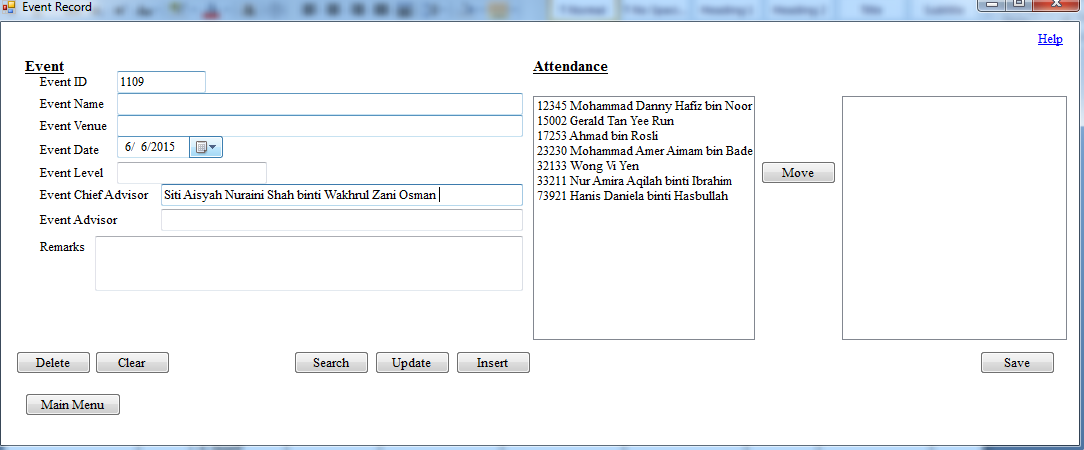
Test 22: ‘School Zone together’ did not produce any error message (Success)



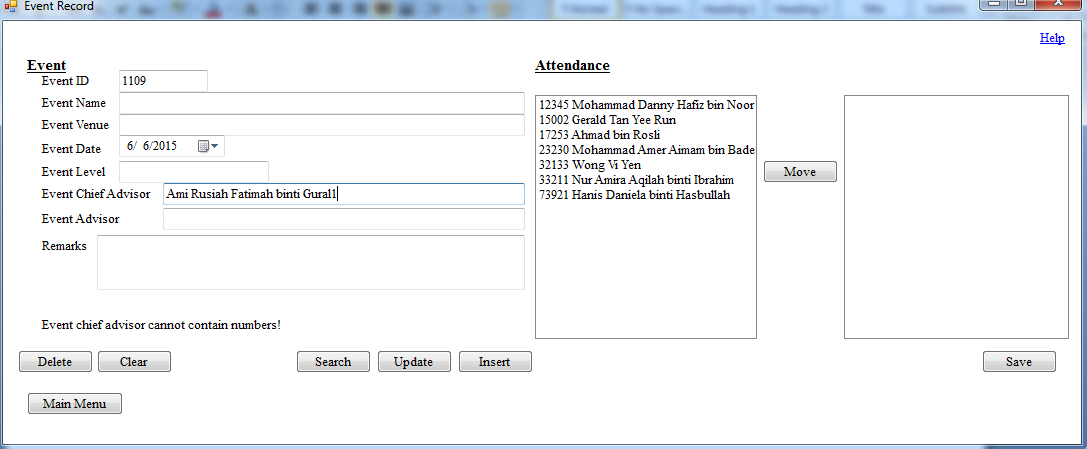
Test 23: ‘Zone 17/09’ produced an error as expected (Success)



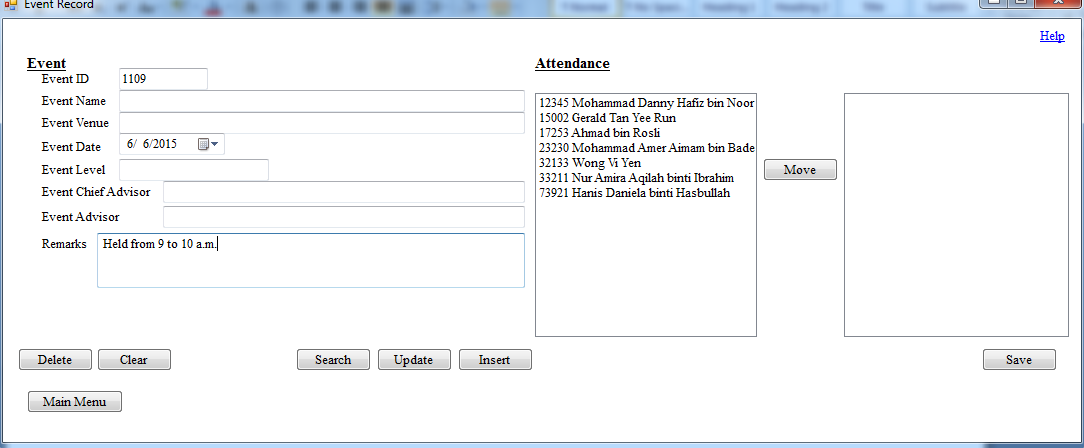
Test 24: ‘Tan Eng Eng’ did not produce any error message (Success)



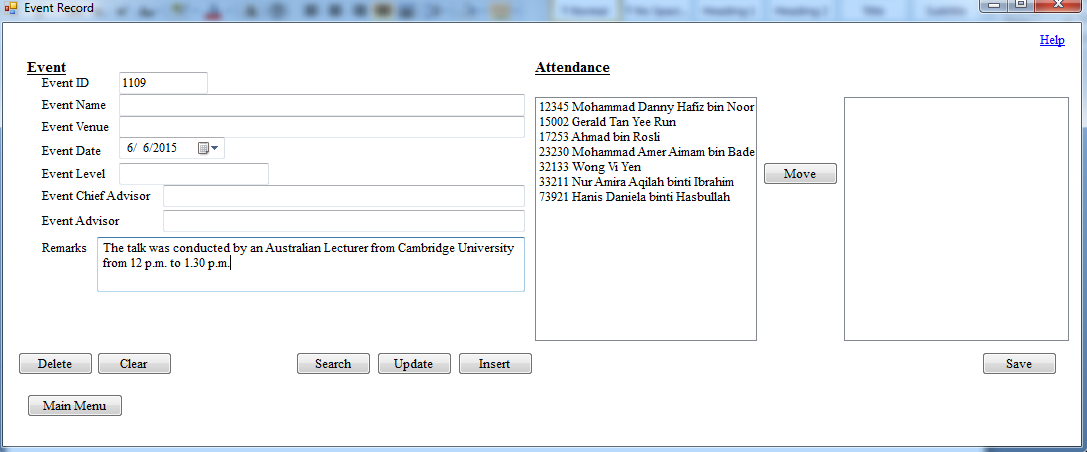
Test 25: ‘Siti Aisyah Nuraini Shah binti Wakhrul Zani Osman’ did not produce any error message (Success)



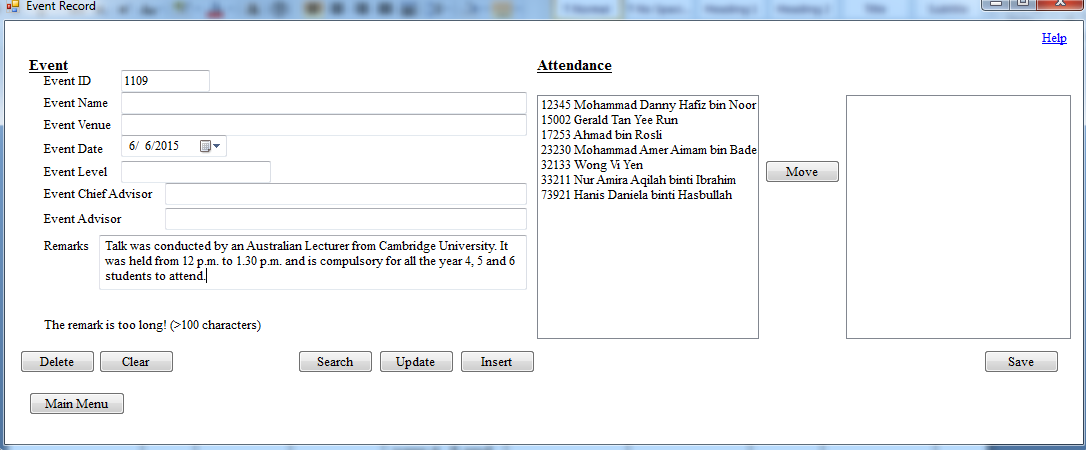
Test 26: ‘Ami Rusiah Fatimah binti Gural1’ produced an error as expected (Success)



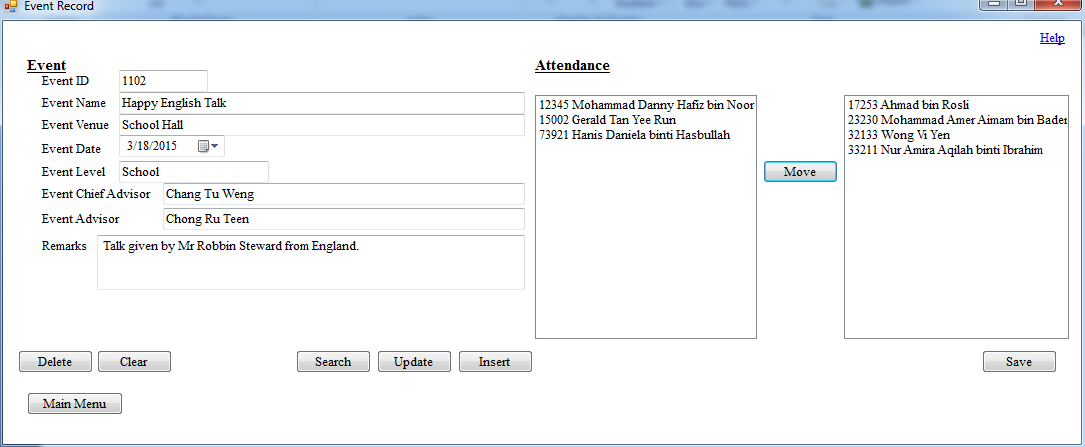
Test 27: ‘Siti Held from 9 to 10 a.m.’ did not produce any error message (Success)



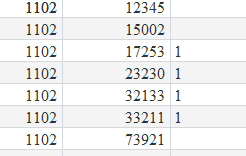
Test 28: ‘The talk was conducted by an Australian Lecturer from Cambridge University from 12 p.m. to 1.30 p.m.’ did not produce any error message (Success)



Test 29: ‘Talk was conducted by an Australian Lecturer from Cambridge University. It was held from 12 p.m. to 1.30 p.m. and is compulsory for all the year 4, 5 and 6 students to attend.’ produced an error as expected (Success)



Test 30: Moved the members who were present in event 1102 to the list box on the right.

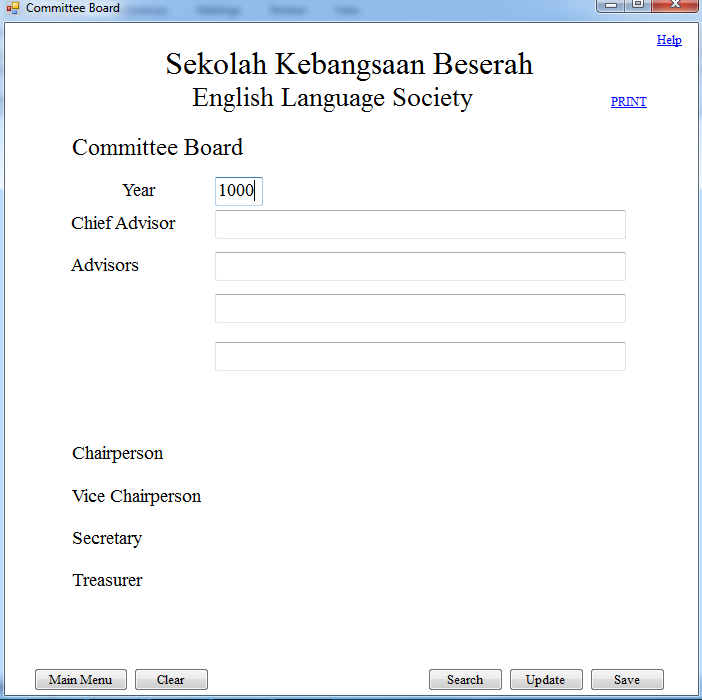


Test 30: Attendance marked as ‘1’ for members present, no error message produced(Success)

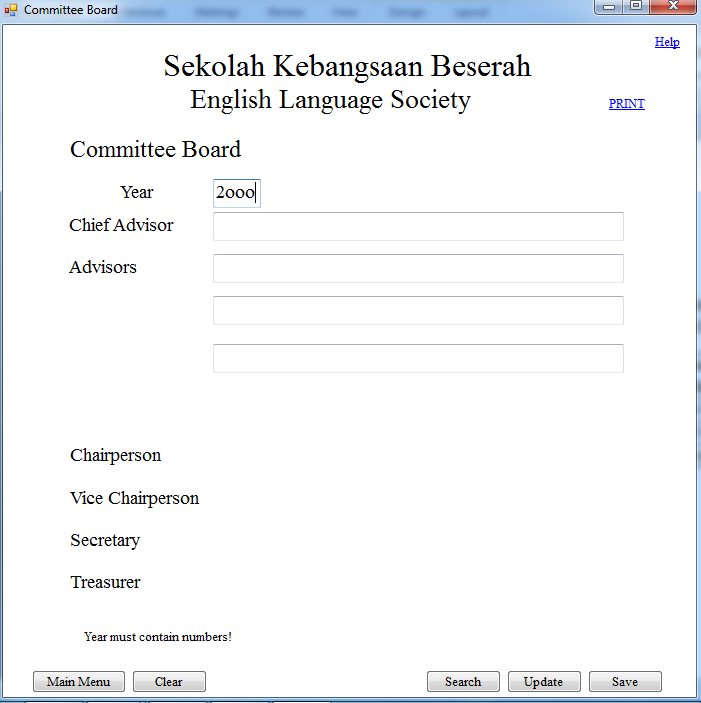
Committee Board Form



Test 31: ‘2014’ did not produce any error message (Success)



Test 32: ‘1000’ did not produce any error message (Success)



Test 33: ‘2ooo’ produced an error as expected (Success)

1. **Installation**

**Installation Plan**

After running several tests on the new system, it should be able to function with little or no problem at all. Therefore, the next stage can be carried out, which is the implementation of the new system. The implementation method that I have chosen is parallel running.

Data stored in this system has a high level of importance to the school, society and students. By parallel running the new system and the previous manual system, if the system corrupts, data will not be lost as the manual system is up to date.

Since the new system is a computerised tailored software which is completely new to the end users, staff training is required. Training will be carried out for two days after school hour on weekdays, depending on the end users’ schedule. The training period is short because the software was designed to have straightforward and clear instructions and functions with simple and basic interfaces which ease the learning and operating processes. Also, the user documentation can help to guide the users to operate the system as desired.

Data need to be converted from handwritten form to computerised form in order to allow the new system to operate. Thus, data need to be prepared, collected then inputted into the new system so that they can be stored in it.

**System Implementation**

After planning an installation plan, I decided to implement the new system on the 25th of August 2015, from 2 p.m. to 3 p.m. in the English Language Society room to allow the users to learn and experience using the system after learning that the users of the system will be free and available within that time period. I planned to guide and teach the users how to install the new system and use all the functions of the system throughout the day. If any question is proposed then I will also answer them on the spot.

The implementation of the new system was successful. No file conversion was necessary as the previous system implemented was a manual one. Only the .NET Framework v4.5 was required to be installed on the computer to enable the new system to operate. Since the laptop which the new system is going to be implemented in has a Windows 7 operating system, this problem is solved as Windows 7 operating system already consists of the component required.

**User Training**

A short and brief user training session was held on the 25th and 27th of August, from 3 p.m. to 5 p.m. in the English Language Society room. The main objectives are to show and teach my client and the end users the functions of the system and how to operate them properly.

Since all the end users are familiar and used to the operation of computers, only brief and simple demonstrations and explanations were needed for them to learn the operation of the new system. On-screen help was also available in the new system, which eases their learning process and can assist them in case they face problems. Furthermore, the new system was designed according to the requests posted by my client, Miss Narulhuda. Some of them already mastered the new system in the first day.



Miss Narulhuda learning to use the new system.

**User Testing**

A set of data were used to test the new system. This time, the tests were performed by one of the end users, Miss Narulhuda binti Yaacob. By testing the system, they became more familiar to the new system.

The tables below are filled up by Miss Narulhuda. The data that they used are shown below:

Member Record Form

Insertion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Member ID | *89374* | *89374* | *89374* |
| 2 | Member Name | *Yee Gan Long* | *Yee Gan Long* | *Yee Gan Long* |
| 3 | Member Class | *3A* | *3A* | *3A* |
| 4 | Condition | *Active* | *Active* | *Active* |
| 5 | Position | *None* | *None* | *None* |
| 6 | Highest English Grade | *A* | *A* | *A* |
| 7 | Potential Contestant | *Potential Contestant* | *Potential Contestant* | *Potential Contestant* |
| 8 | Guardian | *Yee Man Rong* | *Yee Man Rong* | *Yee Man Rong* |
| 9 | Contact Number | *0192838474* | *0192838474* | *0192838474* |

Updating:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Member ID | *89374* | *89374* | *89374* |
| 2 | Contact Number | *0124341531* | *0124341531* | *0124341531* |

Deleting:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Member ID | *89374* | *89374* | *89374* |

Event Record and Event Attendance Form

Insertion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Event ID | *1106* | *1106* | *1106* |
| 2 | Event Name | *Story Telling Competition 2015* | *Story Telling Competition 2015* | *Story Telling Competition 2015* |
| 3 | Event Venue | *SK Citarasa School Hall* | *SK Citarasa School Hall* | *SK Citarasa School Hall* |
| 4 | Event Date | *7/1/2015* | *7/1/2015* | *7/1/2015* |
| 5 | Event Level | *State* | *State* | *State* |
| 6 | Event Chief Advisor | *Tan Eng Hua* | *Tan Eng Hua* | *Tan Eng Hua* |
| 7 | Event Advisor | *Normala binti Hassan Haziz* | *Normala binti Hassan Haziz* | *Normala binti Hassan Haziz* |
| 8 | Remarks | *Held from 10 a.m. to 12 p.m.* | *Held from 10 a.m. to 12 p.m.* | *Held from 10 a.m. to 12 p.m.* |
| 9 | IDs of Members Present | *12345, 33211, 73921* | *12345, 33211, 73921* | *12345, 33211, 73921* |
| 10 | IDs of Members Absent | *15002, 17253, 23230, 32133, 89374* | *15002, 17253, 23230, 32133, 89374* | *15002, 17253, 23230, 32133, 89374* |

Updating:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Event ID | *1106* | *1106* | *1106* |
| 2 | Remarks | *Held from 10 a.m. to 12 p.m., Mohammad Danny Hafiz bin Noor Hazly won the third prize.* | *Held from 10 a.m. to 12 p.m., Mohammad Danny Hafiz bin Noor Hazly won the third prize.* | *Held from 10 a.m. to 12 p.m., Mohammad Danny Hafiz bin Noor Hazly won the third prize.* |

Deleting:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Event ID | *1106* | *1106* | *1106* |

Committee Member Form

Insertion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Year | *2015* | *2015* | *2015* |
| 2 | Chief Advisor | *Narulhuda binti Yaacob* | *Narulhuda binti Yaacob* | *Narulhuda binti Yaacob* |
| 3 | Advisor 1 | *Sharifah Nor Mala binti Hassan* | *Sharifah Nor Mala binti Hassan* | *Sharifah Nor Mala binti Hassan* |
| 4 | Advisor 2 | *Norhasima binti Jusoh* | *Norhasima binti Jusoh* | *Norhasima binti Jusoh* |
| 5 | Advisor 3 | *Shamila binti Syed Hussin* | *Shamila binti Syed Hussin* | *Shamila binti Syed Hussin* |
| 6 | Chairperson | *Mohammad Danny Hafiz bin Noor Hazly* | *Mohammad Danny Hafiz bin Noor Hazly* | *Mohammad Danny Hafiz bin Noor Hazly* |
| 7 | Vice Chairperson | *Mohammad Amer Aimam bin Baderol Hisham* | *Mohammad Amer Aimam bin Baderol Hisham* | *Mohammad Amer Aimam bin Baderol Hisham* |
| 8 | Secretary | *Nur Amira Aqilah binti Ibrahim* | *Nur Amira Aqilah binti Ibrahim* | *Nur Amira Aqilah binti Ibrahim* |
| 9 | Treasurer | *Hanis Daniela binti Hasbullah* | *Hanis Daniela binti Hasbullah* | *Hanis Daniela binti Hasbullah* |

Updating:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Year | *2015* | *2015* | *2015* |
| 2 | Advisor 2 | *Norhasimah binti Jusoh* | *Norhasimah binti Jusoh* | *Norhasimah binti Jusoh* |

Committee Member Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Event ID | *1106* | *1106* | *1106* |
| 2 | Event Name | *Story Telling Competition 2015* | *Story Telling Competition 2015* | *Story Telling Competition 2015* |
| 3 | Event Date | *7/1/2015* | *7/1/2015* | *7/1/2015* |
| 4 | Event Venue | *SK Citarasa School Hall* | *SK Citarasa School Hall* | *SK Citarasa School Hall* |
| 5 | IDs of Members Present | *12345, 33211, 73921* | *12345, 33211, 73921* | *12345, 33211, 73921* |
| 6 | Names of Members Present | *Mohammad Danny Hafiz bin Noor Hazly, Nur Amira Aqilah binti Ibrahim, Hanis Daniela binti Hasbullah* | *Mohammad Danny Hafiz bin Noor Hazly, Nur Amira Aqilah binti Ibrahim, Hanis Daniela binti Hasbullah* | *Mohammad Danny Hafiz bin Noor Hazly, Nur Amira Aqilah binti Ibrahim, Hanis Daniela binti Hasbullah* |

Potential Contestant List Form

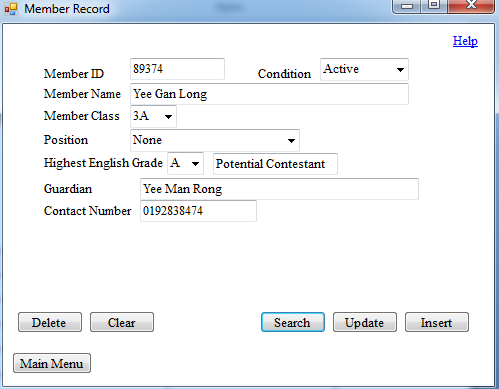
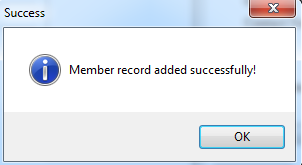
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Field** | **Test Data** | **Expected Results** | **Actual Results** |
| 1 | Highest English Grade | *A, B* | *A, B* | *A, B* |
| 2 | IDs of Members that Appear | *12345, 15002, 23230, 32133, 33211, 89374, 15068, 73921* | *12345, 15002, 23230, 32133, 33211, 89374, 15068, 73921* | *12345, 15002, 23230, 32133, 33211, 89374, 15068, 73921* |
| 3 | Names of Members that Appear | *Mohammad Danny Hafiz bin Noor Hazly, Gerald Tan Yee Run, Mohammad Amer Aimam bin Baderol Hisham, Wong Vi Yen, Nur Amira Aqilah binti Ibrahim, Yee Gan Long, Yumah binti Hussain, Hanis Daniela binti Hasbullah* | *Mohammad Danny Hafiz bin Noor Hazly, Gerald Tan Yee Run, Mohammad Amer Aimam bin Baderol Hisham, Wong Vi Yen, Nur Amira Aqilah binti Ibrahim, Yee Gan Long, Yumah binti Hussain, Hanis Daniela binti Hasbullah* | *Mohammad Danny Hafiz bin Noor Hazly, Gerald Tan Yee Run, Mohammad Amer Aimam bin Baderol Hisham, Wong Vi Yen, Nur Amira Aqilah binti Ibrahim, Yee Gan Long, Yumah binti Hussain, Hanis Daniela binti Hasbullah* |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( HOR SUI LYN ) ( NARULHUDA BINTI YAACOB )

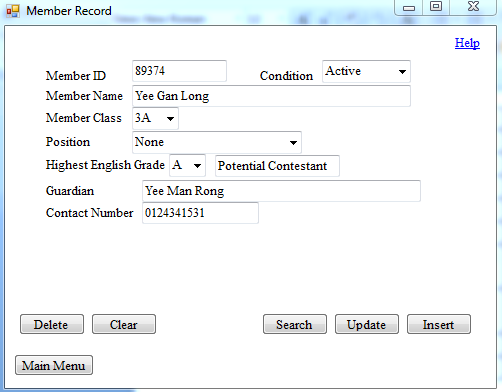
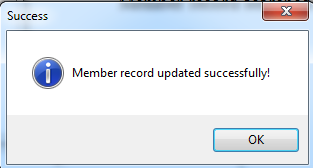
Evidence

All the data are added into the database then searched back to prove that they were added successfully.

Member Record Form:

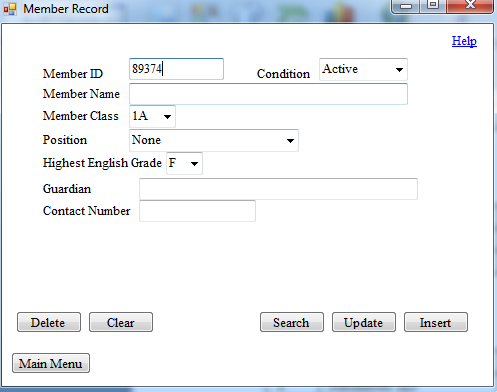
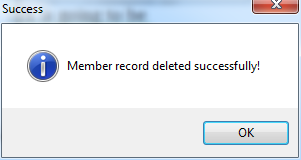
Member record added successfully.

Member record searched successfully.



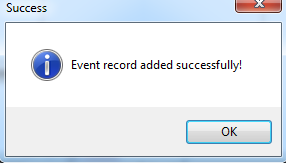
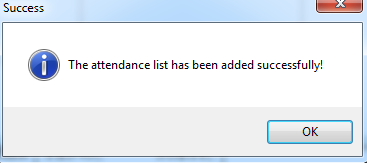
Member record updated successfully.

Member record which is going to be updated.



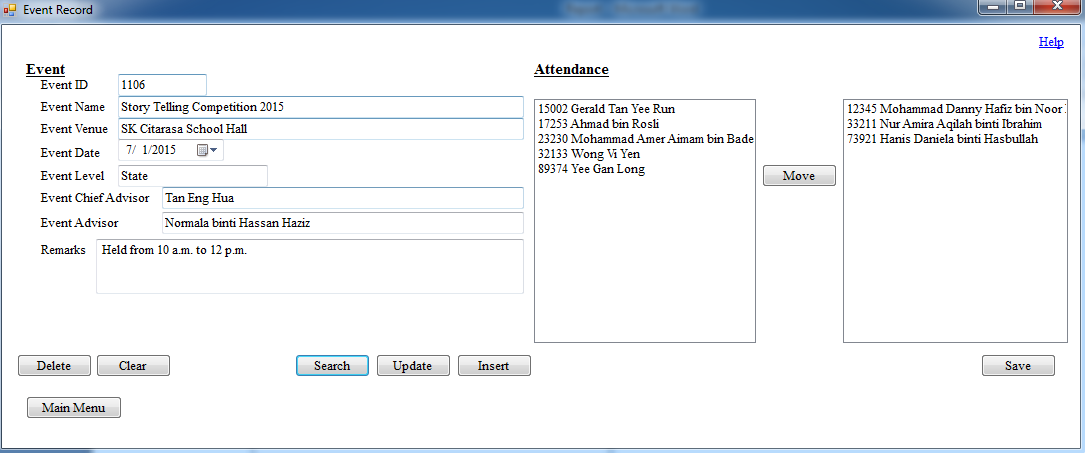
Member record deleted successfully.

Member record which is going to be deleted.

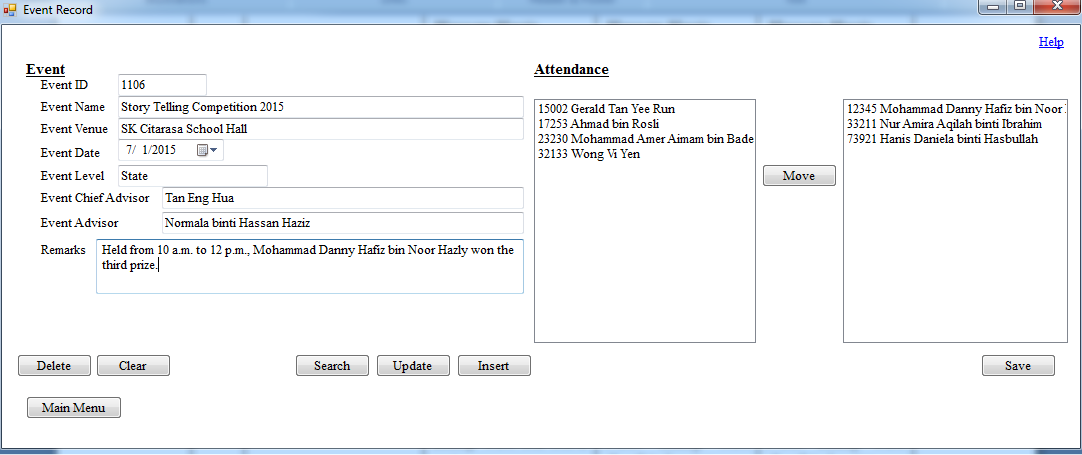
Event Record and Event Attendance Form:

Event attendance added successfully.

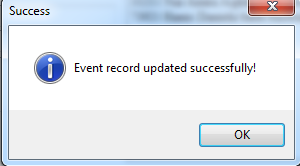
Event record added successfully.



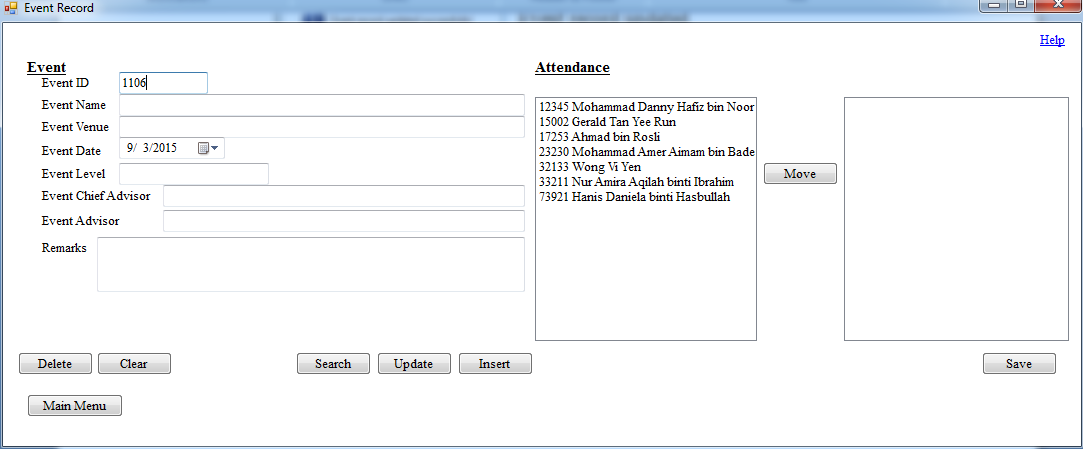
Event record and event attendance searched successfully.



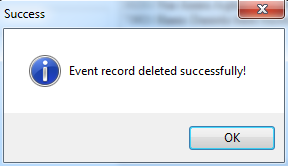
Event record which is going to be updated.



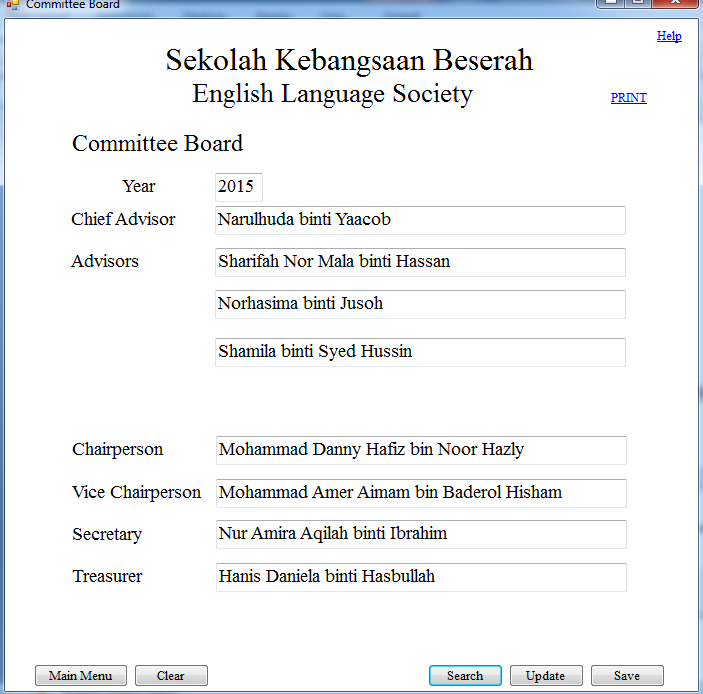
Event record updated successfully.

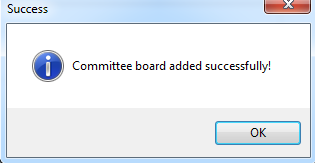


Event record which is going to be deleted.



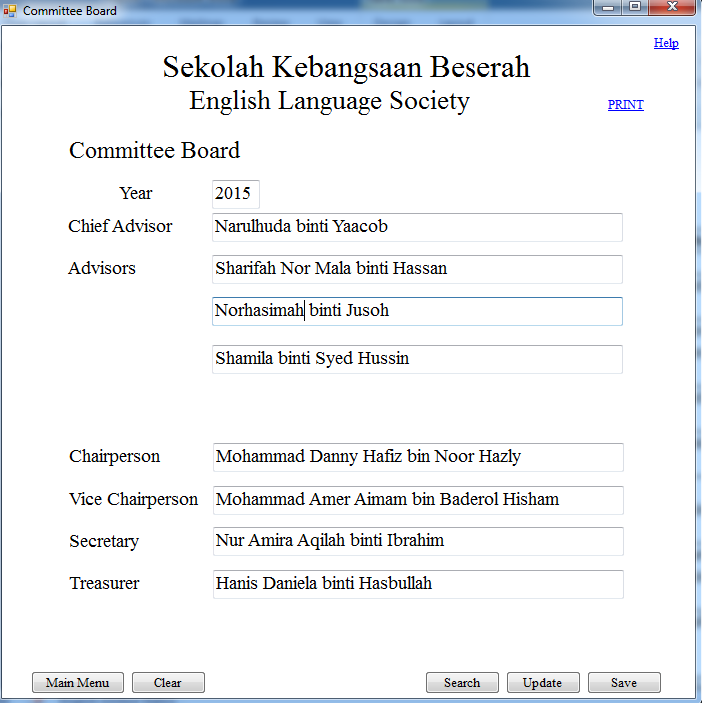
Event record deleted successfully.

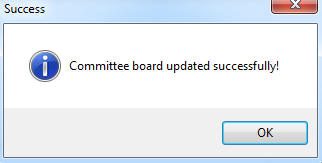
Committee Member Form:



Committee board added successfully.

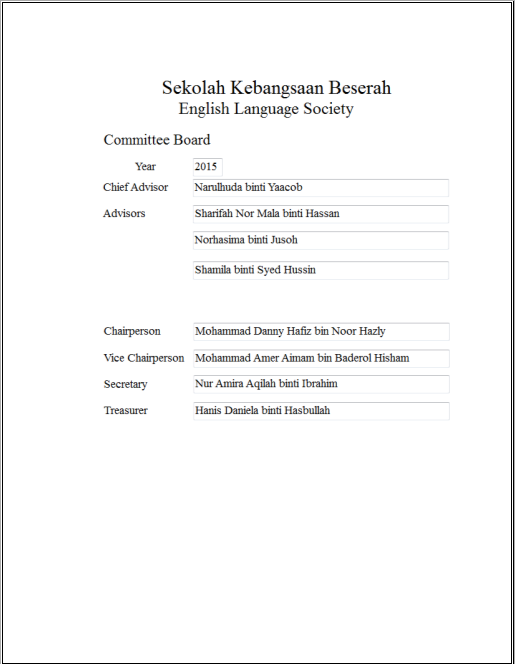
Committee board searched successfully.



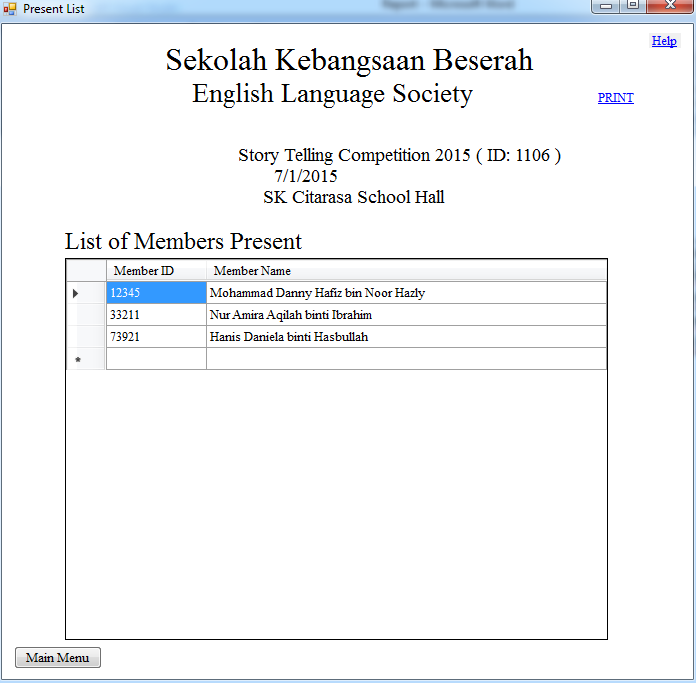


Committee board updated successfully.

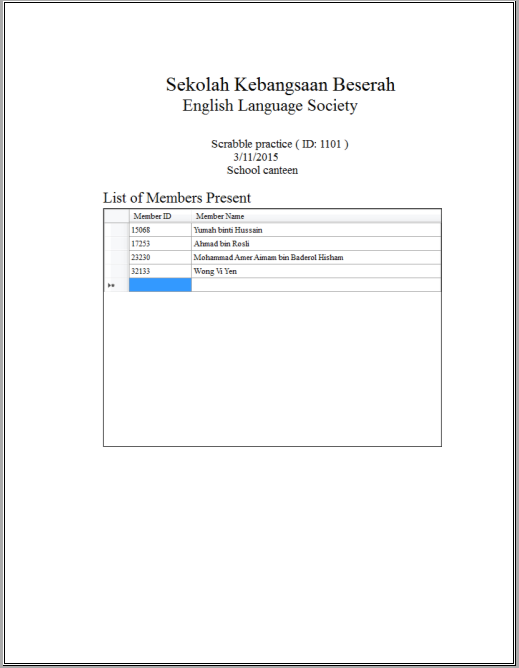
Committee board which is going to be updated.



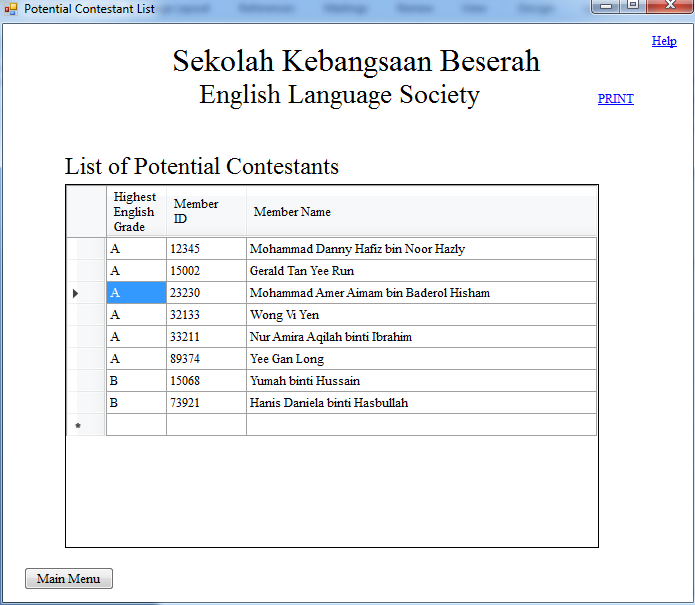
Committee board print previewed successfully and ready to be printed.

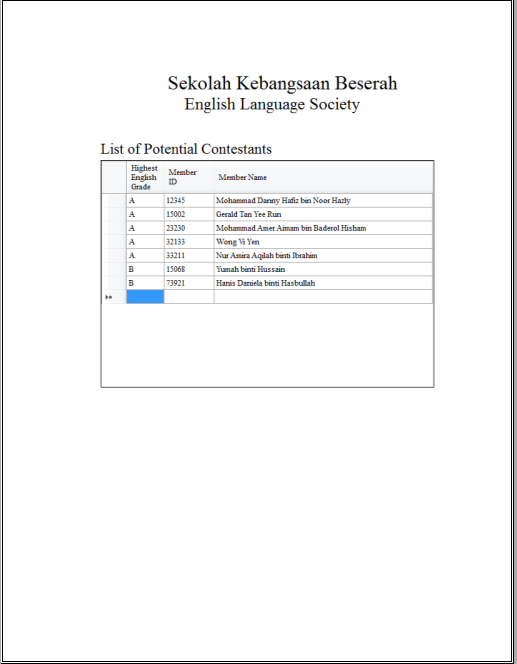
List of Members Present Form:

List of members present in event 1101 is generated successfully.

****

List of members present in event 1101 is print previewed successfully and ready to be printed.

Potential Contestant List Form:

****

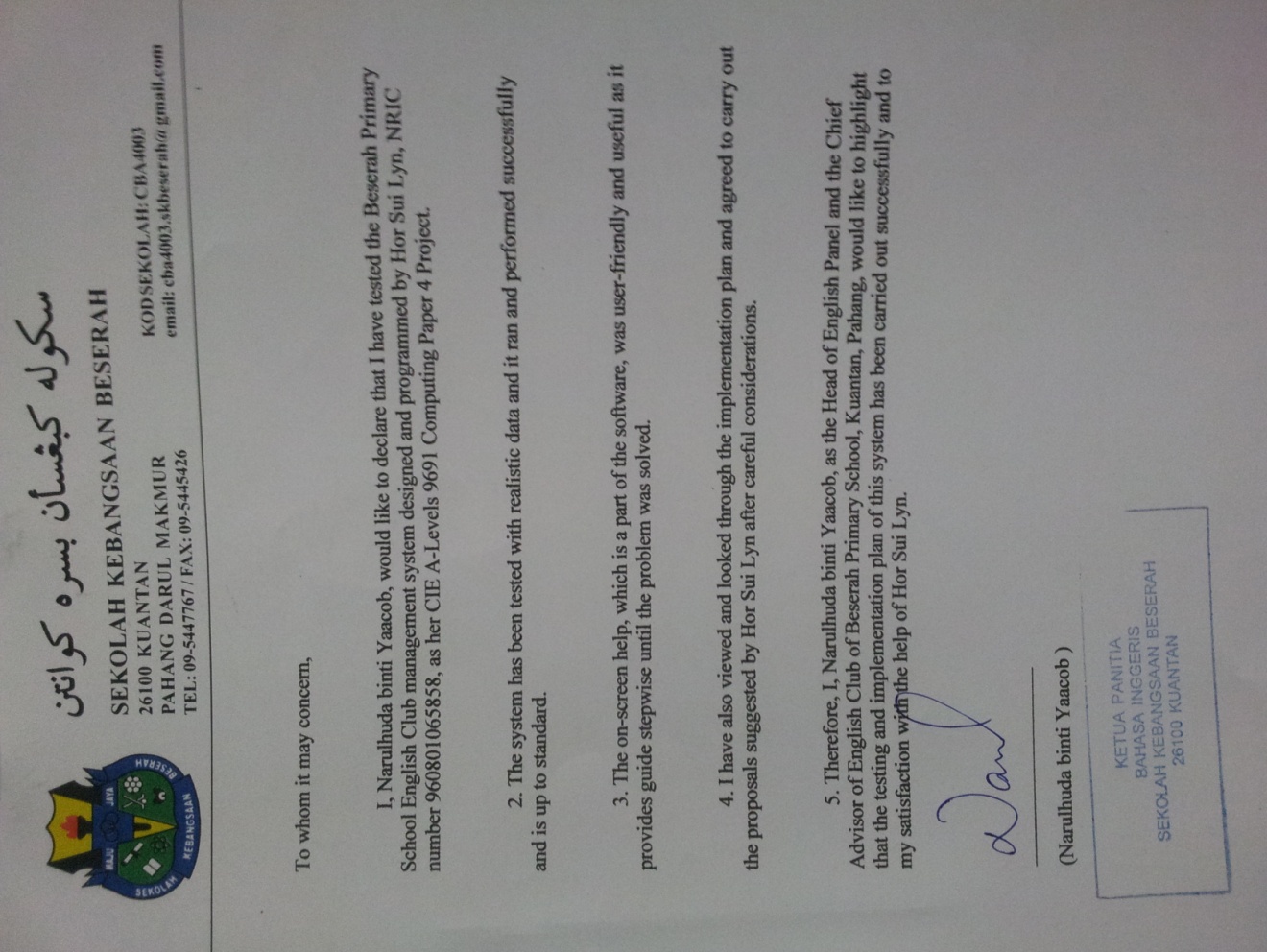
Potential contestant list is print previewed successfully and ready to be printed.

List of members present in event 1101 is generated successfully.

**User Acceptance**

Miss Narulhuda binti Yaacob had prepared a user acceptance letter as a proof of accepting the system designed for the English Language Society.

Below is the letter:



**(E) Documentation**

1. **System Maintenance Documentation**

**Forms**

|  |  |
| --- | --- |
| **Form Name** | **Use** |
| frmMainMenu | Displays a list of operations for the user to choose from |
| frmMemberRecord | Allows saving, updating, searching and deletion of member record |
| frmEventRecord | Allows saving, updating, searching and deletion of event record besides saving and searching of event attendance |
| frmCommitteeBoard | Allows saving, updating and searching of committee board besides allowing committee board to be printed |
| frmPresentList | Allows present list of event to be shown on-screen and to be printed |
| frmPotentialContestantList | Allows potential contestant list to be shown on-screen and to be printed |

**frmMainMenu Objects**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Type** | **Use** |
| btnMemberRecord | Button | Directs to the member record form |
| btnEventRecord | Button | Directs to the event record and event attendance form |
| btnCommitteeBoard | Button | Directs to the committee board form |
| btnPresentList | Button | Directs to the list of members present form |
| btnPotentialContestantList | Button | Directs to the potential contestant list form |
| lnkHelp | Link label | Opens the help page |

**frmMemberRecord Objects**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Type** | **Use** |
| lblMemberID | Label | Indicates that the textbox on the right is to be filled with the ID of the member |
| lblMemberName | Label | Indicates that the textbox on the right is to be filled with the name of the member |
| lblCondition | Label | Indicates that the combo box on the right is to be filled with the current condition of the member in the society |
| lblMemberClass | Label | Indicates that the combo box on the right is to be filled with the class of the member |
| lblPosition | Label | Indicates that the combo box on the right is to be filled with the position of the member in the society |
| lblHighestEnglishGrade | Label | Indicates that the combo box on the right is to be filled with the highest English grade achieved by the member |
| lblGuardian | Label | Indicates that the textbox on the right is to be filled with the name of the guardian of the member |
| lblContactNumber | Label | Indicates that the textbox on the right is to be filled with the reachable contact number of the guardian of the member |
| lblBlank | Label | To display validation texts when important fields are left blank. |
| lblError | Label | To display validation texts when data inputted is invalid. |
| txtMemberID | Textbox | Space to fill in the ID of the member |
| txtMemberName | Textbox | Space to fill in the name of the member |
| cboCondition | Combo box | List of choices to choose the current condition of the member in the society |
| cboMemberClass | Combo box | List of choices to choose the class of the member |
| cboPosition | Combo box | List of choices to choose the position of the member in the society |
| cboHighestEnglishGrade | Combo box | List of choices to choose the highest English grade achieved by the member |
| txtPotentialContestant | Textbox | Space to display whether the member is a potential contestant |
| txtGuardian | Textbox | Space to fill in the name of the guardian of the member |
| txtContactNumber | Textbox | Space to fill in the reachable contact number of the guardian of the member |
| btnDelete | Button | Allows deletion of member record to be performed |
| btnClear | Button | Allows all the textboxes and combo boxes to be cleared and reset |
| btnSearch | Button | Allows searching of member record to be performed |
| btnUpdate | Button | Allows update of member record to be performed |
| btnInsert | Button | Allows insertion of member record to be performed |
| btnMainMenu | Button | Directs to the main menu form |
| lnkHelp | Link label | Opens the help page |

**frmEventRecord Objects**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Type** | **Use** |
| lblEvent | Label | Indicates that the left side of the form is for details of the event |
| lblAttendance | Label | Indicates that the right side of the form is for details of the attendance |
| lblEventID | Label | Indicates that the textbox on the right is to be filled with the ID of the event |
| lblEventName | Label | Indicates that the textbox on the right is to be filled with the name of the event |
| lblEventVenue | Label | Indicates that the combo box on the right is to be filled with the venue where the event was held |
| lblEventDate | Label | Indicates that the combo box on the right is to be filled with the date when the event was held |
| lblEventLevel | Label | Indicates that the combo box on the right is to be filled with the level of the event |
| lblEventChiefAdvisor | Label | Indicates that the combo box on the right is to be filled with the name of the chief advisor of the event |
| lblEventAdvisor | Label | Indicates that the textbox on the right is to be filled with the name of the advisor who assisted the chief advisor of the event |
| lblRemark | Label | Indicates that the textbox on the right is to be filled with additional details of the event |
| lblBlank | Label | To display validation texts when important fields are left blank. |
| lblError | Label | To display validation texts when data inputted is invalid. |
| txtEventID | Textbox | Space to fill in the ID of the event |
| txtEventName | Textbox | Space to fill in the name of the event |
| txtEventVenue | Textbox | Space to fill in the venue where the event was held |
| dtpEventDate | Date time picker | Mini calendar to choose the date when the event was held |
| txtEventLevel | Textbox | Space to fill in the level of the event |
| txtEventChiefAdvisor | Textbox | Space to fill in the name of the chief advisor of the event |
| txtEventAdvisor | Textbox | Space to fill in the name of the advisor who assisted the chief advisor of the event |
| txtRemark | Textbox | Space to fill in the additional details of the event |
| lstMemberList | List box | List box to place or show the IDs and names of members who were absent in an event |
| lstPresentList | List box | List box to place or show the IDs and names of members who were present in an event |
| btnDelete | Button | Allows deletion of event record to be performed |
| btnClear | Button | Allows all the textboxes, the date time picker and the list boxes to be cleared and reset |
| btnSearch | Button | Allows searching of event record to be performed |
| btnUpdate | Button | Allows update of event record to be performed |
| btnInsert | Button | Allows insertion of event record to be performed |
| btnMove | Button | Allows the ID and name of members to be shifted between the two list boxes |
| btnSave | Button | Allows insertion of event attendance to be performed |
| btnMainMenu | Button | Directs to the main menu form |
| lnkHelp | Link label | Opens the help page |

**frmCommitteeBoard Objects**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Type** | **Use** |
| lblSchoolName | Label | Indicates the name of the school |
| lblSocietyName | Label | Indicates the name of the society |
| lblTitle | Label | Indicates the title of the report |
| lblYear | Label | Indicates that the textbox on the right is to be filled with the year in which the committee board is effective |
| lblChiefAdvisor | Label | Indicates that the textbox on the right is to be filled with the name of the chief advisor of the society |
| lblAdvisors | Label | Indicates that the textboxes on the right is to be filled with the name of the advisors who assisted the chief advisor of the society |
| lblChairperson | Label | Indicates that the textbox on the right is filled with the name of the member who held the position of the chairperson of the society |
| lblViceChairperson | Label | Indicates that the textbox on the right is filled with the name of the member who held the position of the vice chairperson of the society |
| lblSecretary | Label | Indicates that the textbox on the right is filled with the name of the member who held the position of the secretary of the society |
| lblTreasurer | Label | Indicates that the textbox on the right is filled with the name of the member who held the position of the treasurer of the society |
| lblBlank | Label | To display validation texts when important fields are left blank. |
| lblError | Label | To display validation texts when data inputted is invalid. |
| txtYear | Textbox | Space to fill in the year in which the committee board is effective |
| txtChiefAdvisor | Textbox | Space to fill in the name of the chief advisor of the society |
| txtAdvisor1 | Textbox | Space to fill in the name of one of the advisors who assisted the chief advisor of the society |
| txtAdvisor2 | Textbox | Space to fill in the name of one of the advisors who assisted the chief advisor of the society |
| txtAdvisor3 | Textbox | Space to fill in the name of one of the advisors who assisted the chief advisor of the society |
| txtChairperson | Textbox | Space to fill in the name of the member who held the position of chairperson of the society |
| txtViceChairperson | Textbox | Space to fill in the name of the member who held the position of vice chairperson of the society |
| txtSecretary | Textbox | Space to fill in the name of the member who held the position of secretary of the society |
| txtTreasurer | Textbox | Space to fill in the name of the member who held the position of treasurer of the society |
| btnMainMenu | Button | Directs to the main menu form |
| btnClear | Button | Allows all the textboxes to be cleared and reset |
| btnSearch | Button | Allows searching of committee board to be performed |
| btnUpdate | Button | Allows update of committee board to be performed |
| btnSave | Button | Allows insertion of committee board to be performed |
| lnkHelp | Link label | Opens the help page |
| lnkPrint | Link label | Allows print preview and print action to be performed |

**frmPresentList Objects**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Type** | **Use** |
| lblSchoolName | Label | Indicates the name of the school |
| lblSocietyName | Label | Indicates the name of the society |
| lblTitle | Label | Indicates the title of the report |
| lblEventNameAndID | Label | Generates the name of the event and its ID in bracket |
| lblEventDate | Label | Generates the date when the event was held |
| lblEventVenue | Label | Generates the venue where the event was held |
| dgvPresentList | Data grid view | Generates the IDs and names of members who were present in the event in which its ID was inputted in the search input box |
| btnMainMenu | Button | Directs to the main menu form |
| lnkHelp | Link label | Opens the help page |
| lnkPrint | Link label | Allows print preview and print action to be performed |

**frmPotentialContestantList Objects**

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Type** | **Use** |
| lblSchoolName | Label | Indicates the name of the school |
| lblSocietyName | Label | Indicates the name of the society |
| lblTitle | Label | Indicates the title of the report |
| dgvPotentialContestantList | Data grid view | Generates the highest English grades achieved, IDs and names of members who are potential contestants |
| btnMainMenu | Button | Directs to the main menu form |
| lnkHelp | Link label | Opens the help page |
| lnkPrint | Link label | Allows print preview and print action to be performed |

**Tables**

|  |  |
| --- | --- |
| **Table Name** | **Use** |
| tblMemberRecord | To store details of members |
| tblEventRecord | To store details of events |
| tblEventAttendance | To link the member record table and the event record table and to allow attendance of members for events to be stored |
| tblCommitteeBoard | To store committee members every year |

**tblMemberRecord**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Use** |
| Member ID | Integer | Allows member records to be differed and allows the event attendance table to be linked through this unique key |
| Member Name | String | Allows the member to be recognised |
| Member Class | String | Allows the member to be found in special cases |
| Condition | String | Stores the current condition of the member in the society |
| Position | String | Allows the committee board to be generated |
| Highest English Grade | String | Allows the potential contestant list to be generated |
| Potential Contestant | String | Allows the potential contestant list to be generated |
| Guardian | String | Allows the name of the guardian of the member to be specified |
| Contact Number | String | Allows the guardian of the member to be contacted in emergency cases |

**tblEventRecord**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Use** |
| Event ID | Integer | Allows event records to be differed and allows the event attendance table to be linked through this unique key |
| Event Name | String | Allows the events to be recognised |
| Event Venue | String | Stores the venue where the event was held |
| Event Date | String | Stores the date when the event was held |
| Event Level | String | Stores the level of the event |
| Event Chief Advisor | String | Allows the chief advisor of the event to be identified in case problems or issues regarding the event arise |
| Event Advisor | String | Allows the advisor of the event to be identified in case problems or issues regarding the event arise |
| Remark | String | Stores additional details of the event |

**tblEventAttendance**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Use** |
| Event ID | Integer | Allows event records to be differed and allows the event record table to be linked through this unique key |
| Member ID | Integer | Allows member records to be differed and allows the member record table to be linked through this unique key |
| Attendance | String | Allows the attendance of members for events to be taken and allows lists of members present in events to be generated |

**tblCommitteeBoard**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Use** |
| Year | Integer | Allows committee boards to be differed through this unique key |
| Chief Advisor | String | Stores the name of the chief advisor of the society in a certain year |
| Advisor 1 | String | Stores the name of one of the advisors of the society in a certain year |
| Advisor 2 | String | Stores the name of one of the advisors of the society in a certain year |
| Advisor 3 | String | Stores the name of one of the advisors of the society in a certain year |
| Chairperson | String | Stores the name of the member who held the position of chairperson of the society in a certain year |
| Vice Chairperson | String | Stores the name of the member who held the position of vice chairperson of the society in a certain year |
| Secretary | String | Stores the name of the member who held the position of secretary of the society in a certain year |
| Treasurer | String | Stores the name of the member who held the position of treasurer of the society in a certain year |

**Section Reference**

Certain sections in the System Maintenance Documentation has been included in the previous pages. The table below shows the sections and their pages:

|  |  |
| --- | --- |
| **Section Reference** | **Page reference** |
| Data Dictionary | 31-34 |
| Hardware Requirements | 28 |
| Software Requirements | 27 |
| Input Design | 35-46 |
| Output Design | 47-52 |
| Database Design | 69-74 |
| Process Model | 58-63 |
| Annotated Program Listings | 92-131 |
| Structured Query Language (SQL) | 84-91 |

**Possible Adaptive Maintenance**

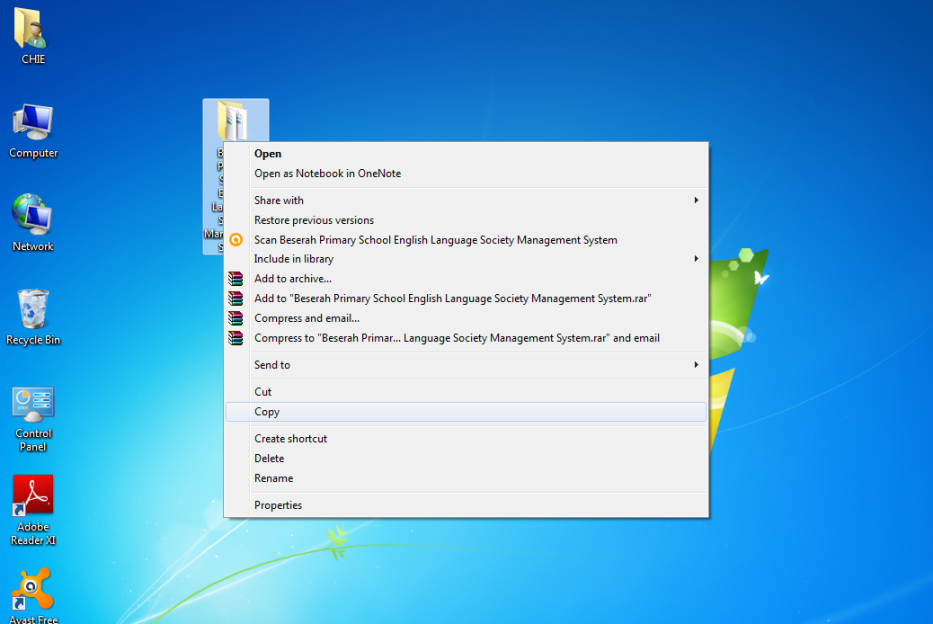
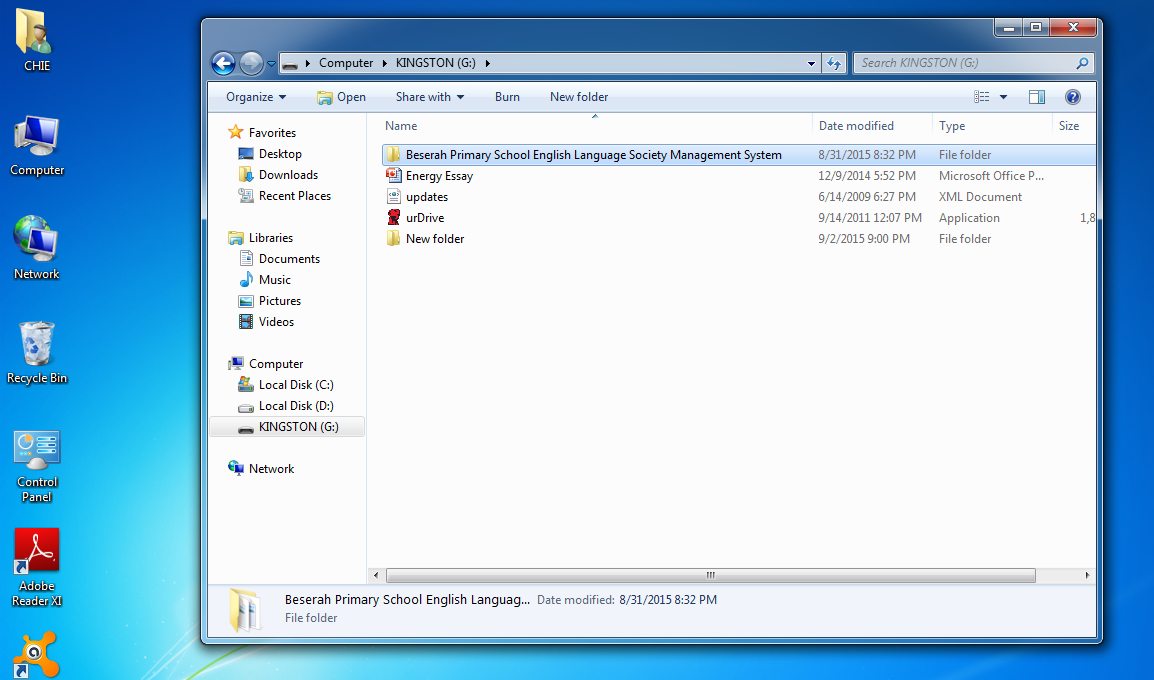
This is a guide for the system analyst to perform adaptive maintenance on this Beserah Primary School English Language Society Management System if any changes occur in the future.

Firstly, the classes of the members available in the school may change. Thus, the values in the combo box in the programme will need to be changed and updated.

Next, the committee member positions held by the members may increase if more members join the society. The positions of Vice Secretary, Vice Treasurer, Year Representatives may be added. Hence, the committee board form and its output will need to be modified. The values of the combo box for position in the member record form also need to be updated.

Furthermore, if the operating system in use is changed, the version Microsoft Office installed needs to be checked its compatibility with the version used to create the database. The .NET Framework v4.5 must also be installed if not included in the operating system in order to operate the programme.

**Backup Procedures**

1. Copy the Beserah Primary School English Language Society Management System on the Desktop.
2. Go to Computer, then to the pendrive’s folder, and paste it.
3. Keep the external backing storage (preferably flash drives) away from the laptop. It is recommended to back up the system and database every week or more frequent.

**Storage Media Use**

It is recommended for the user to have at least two pendrives of at least 4 GB storage space to backup the system and database every week or more frequent. This is to prevent the loss of precious data which may cause the breakdown of the system.

1. **User Guide**

**Introduction to the System**

Beserah Primary School English Language Society Management System is a co-curricular society management system designed by Hor Sui Lyn for Beserah Primary School, Kuantan, Pahang. This system allows the registration, update, search and deletion of details of the members and events, attendance taking of members to an event, search of attendance of an event, input and search of committee board, production of outputs of list of members present in an event, list of potential contestants and list of committee members of a particular year.

For further information, please contact Hor Sui Lyn at [riyuu-ga-nai@hotmail.com](mailto:riyuu-ga-nai@hotmail.com).

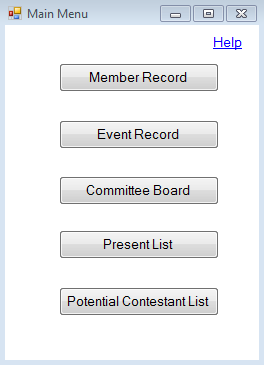
**Installing of System and Starting the Programme**

1. Copy the Beserah Primary School English Language Society Management System folder from the USB drive.
2. Paste the folder on the desktop.
3. Enter the English Language Society folder on the desktop, right click the English Language Society Management System application, point to Send To and select Desktop(create shortcut).
4. This shortcut can be used to access the programme faster. Left click it twice to start running the programme. You may rename the shortcut if you want to.

**Using the System**

Basic functions of forms:

Main Menu

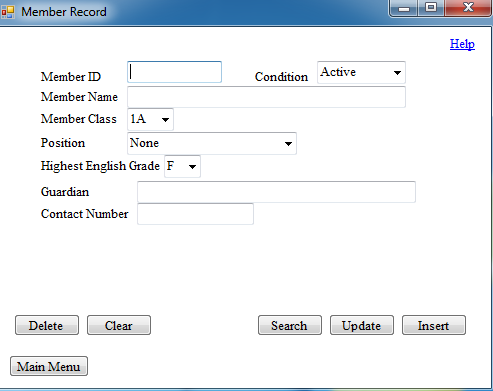


When the programme loads, the Main Menu appears. You can choose what to do by left clicking on the button once.

|  |  |
| --- | --- |
| **Button** | **Function** |
| Member Record | To manage details of a member. |
| Event Record | To manage details of an event and attendance of members in a certain event. |
| Committee Board | To manage a committee board effective in a certain year. |
| Present List | To generate a list of members present in a specific event. |
| Potential Contestant List | To generate a list of potential contestants who are currently active. |

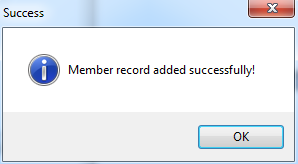
Member Record Form

You are redirected here if the ‘Member Record’ button was clicked on the main menu.



Registration of member:

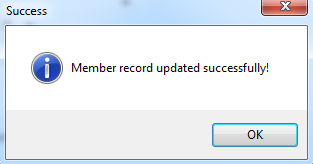
1. Fill up the form.
2. The ‘Member ID’ field should be filled up with the student ID.
3. All the fields in this form are mandatory to be filled up.
4. If words appear in the space between the buttons and the textboxes, edit or change the data mentioned in the words accordingly.
5. If no words appear after filling up the form, click on the ‘Insert’ button to save the new member’s particulars.
6. If the saving of data was successful, a message box will pop out to inform that the operation is successful.



Message box to inform that saving of member record is successful.

Editing of member details:

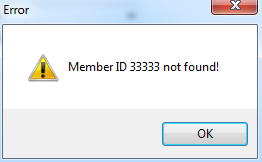
1. Fill up the form or change the details to be updated.
2. All the fields in this form are mandatory to be filled up.
3. If words appear in the space between the buttons and the textboxes, edit or change the data mentioned in the words accordingly.
4. If no words appear after filling up the form, click on the ‘Update’ button to update the member’s particulars.
5. If the updating of data was successful, a message box will pop out to inform that the operation is successful.

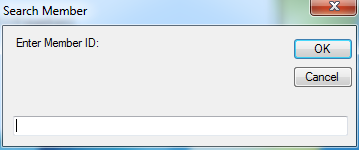


Message box to inform that updating of member record is successful.

Search of member details:

1. Click the ‘Search’ button.
2. An input box will pop out requesting you to enter the ID of the member to be searched.
3. After confirming that the member ID is entered correctly, click on the ‘Ok’ button.
4. If that member ID is not found in the database, an error message will pop out.
5. If that member ID is found in the database, that member’s details will be displayed in the form.



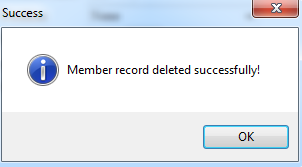
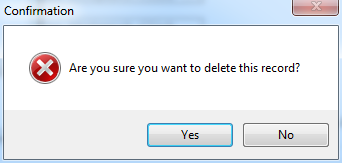


Input box to enter the Error message to inform

member ID to be searched. that the member ID was not found.

Deletion of a member:

1. Enter the ID of the member to be deleted from the database in the ‘Member ID’ field.
2. Click the ‘Delete’ button.
3. A message box will pop out to confirm the deletion.
4. If the ‘Yes’ button is clicked, the delete operation will be performed. Else, it will be canceled.
5. If the member ID is found and the deletion is successful, a message box will pop out to inform the successful deletion.



Message box to confirm deletion. Message box to inform that deletion

of member record is successful.

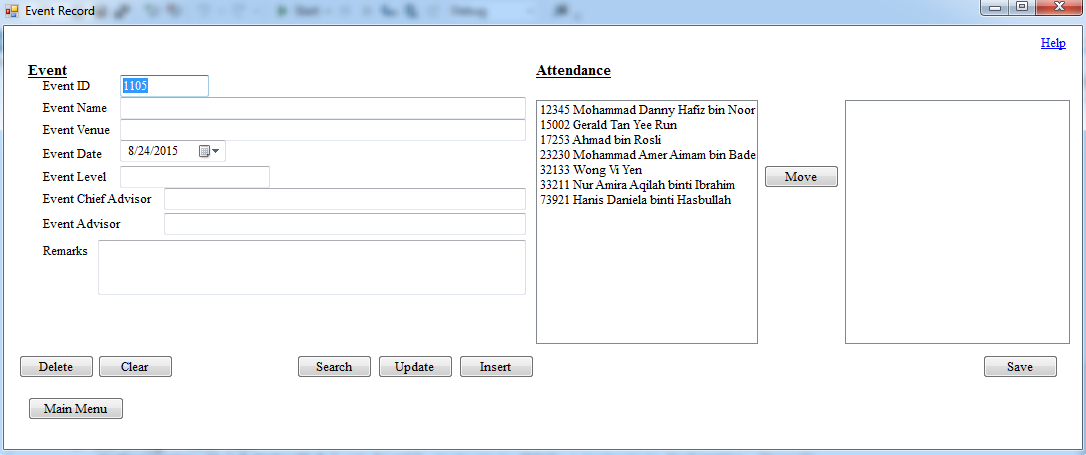
Clearing and resetting the form:

1. Click on the ‘Clear’ button.
2. All the textboxes will be reset to blank.

Return to main menu:

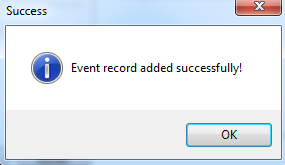
1. Click on the ‘Main Menu’ button.

Event Record and Event Attendance Form

You are redirected here if the ‘Event Record’ button was clicked on the main menu.

Registration of event:

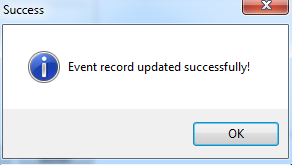
1. Fill up the form.
2. The ‘Event ID’ field will be automatically generated when the form is loaded.
3. All the fields in this form, except the ‘Remark’ field, are mandatory to be filled up.
4. If words appear in the space between the buttons and the textboxes, edit or change the data mentioned in the words accordingly.
5. If no words appear after filling up the form, click on the ‘Insert’ button to save the new event’s details.
6. If the saving of data was successful, a message box will pop out to inform that the operation is successful.



Message box to inform that saving of member record is successful.

Editing of event details:

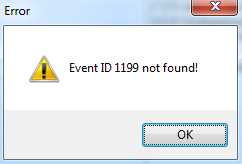
1. Fill up the form or change the details to be updated.
2. All the fields in this form, except the ‘Remark’ field, are mandatory to be filled up.
3. If words appear in the space between the buttons and the textboxes, edit or change the data mentioned in the words accordingly.
4. If no words appear after filling up the form, click on the ‘Update’ button to update the event’s details.
5. If the updating of data was successful, a message box will pop out to inform that the operation is successful.

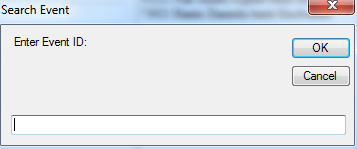


Message box to inform that updating of member record is successful.

Search of event details and its attendance:

1. Click the ‘Search’ button.
2. An input box will pop out requesting you to enter the ID of the event to be searched.
3. After confirming that the event ID is entered correctly, click on the ‘Ok’ button.
4. If that event ID is not found in the database, an error message will pop out.
5. If that event ID is found in the database, that event’s details and its attendance will be displayed in the form.



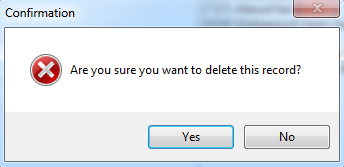
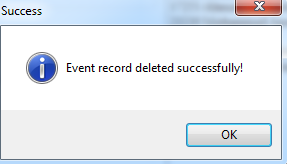


Input box to enter the Error message to inform

event ID to be searched. that the event ID was not found.

Deletion of an event:

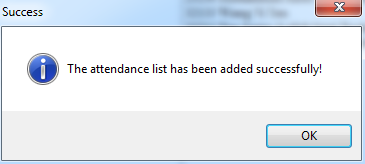
1. Enter the ID of the event to be deleted from the database in the ‘Event ID’ field.
2. Click the ‘Delete’ button.
3. A message box will pop out to confirm the deletion.
4. If the ‘Yes’ button is clicked, the delete operation will be performed. Else, it will be canceled.
5. If the event ID is found and the deletion is successful, a message box will pop out to inform the successful deletion.

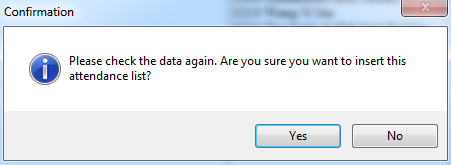


Message box to confirm deletion. Message box to inform that deletion

of event record is successful.

Taking attendance of members to an event:

1. Enter the event ID or search for the event from the database if it already exists.
2. Select the member and click on the ‘Move’ button to move the members between to the list box on the right if they were present in the event.
3. The members can also be moved back to the list box on the left if the wrong members are moved, by selecting them from the list box on the right then clicking the ‘Move’ button.
4. Once done, click on the ‘Save’ button.
5. A message box will pop out to confirm your action. If you are sure that the attendance is taken correctly, click ‘Yes’.
6. A message box will pop out to inform the successful saving of attendance if the operation is successful.



Message box to confirm saving of attendance list. Message box to inform that

saving of attendance list is successful.

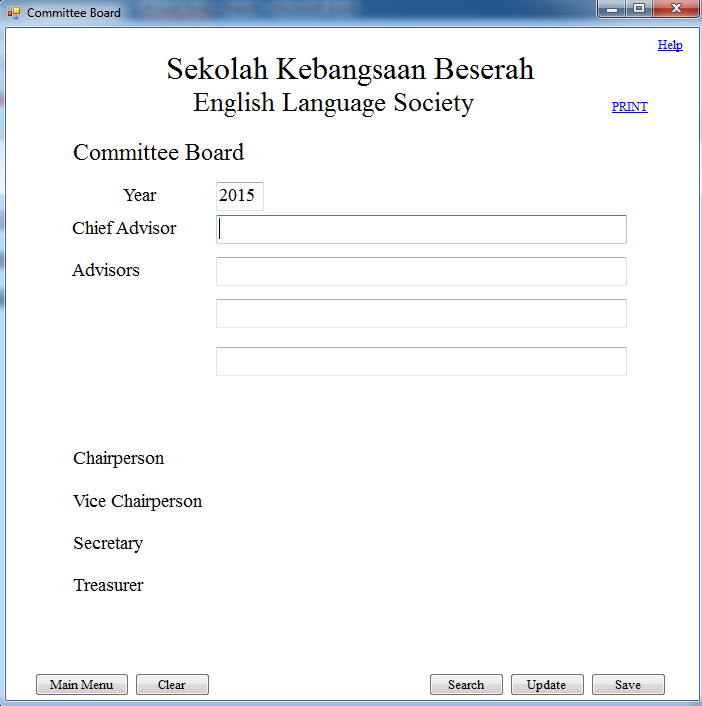
Clearing and resetting the form:

1. Click on the ‘Clear’ button.
2. All the textboxes will be reset to blank.

Return to main menu:

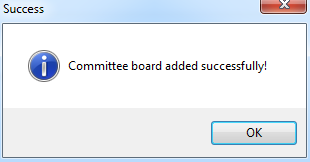
1. Click on the ‘Main Menu’ button.

Committee Board Form:

You are redirected here if the ‘Committee Board’ button was clicked on the main menu.

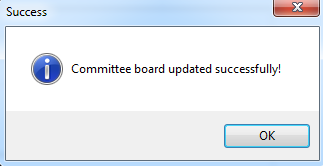
Saving new committee board:

1. Fill up the form.
2. The ‘Year’ field will be automatically filled with the current year when the form is loaded.
3. All the fields with textboxes shown in this form are mandatory to be filled up.
4. If words appear in the space between the buttons and the textboxes, edit or change the data mentioned in the words accordingly.
5. If no words appear after filling up the form, click on the ‘Save’ button to save the new committee board’s details.
6. If the saving of data was successful, a message box will pop out to inform that the operation is successful.
7. The members holding the ‘Chairperson’, ‘Vice Chairperson’, ‘Secretary’ and ‘Treasurer’ positions will be searched from the database and saved in the committee board table in the database.



Message box to inform that saving of committee board is successful.

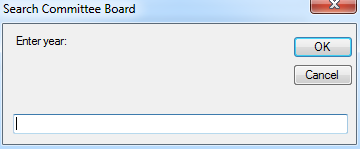
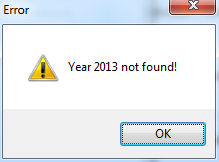
Updating existing committee board:

1. Fill up the form.
2. All the fields with textboxes shown in this form are mandatory to be filled up.
3. If words appear in the space between the buttons and the textboxes, edit or change the data mentioned in the words accordingly.
4. If no words appear after filling up the form, click on the ‘Update’ button to update the committee board of the year filled up in the ‘Year’ field.
5. If the updating of data was successful, a message box will pop out to inform that the operation is successful.

Message box to inform that updating of committee board is successful.

Searching existing committee board:

1. Click on the ‘Search’ button.
2. An input box will pop out requesting you to enter the year in which the committee board was effective.
3. After confirming that the year is entered correctly, click on the ‘Ok’ button.
4. If that year is not found in the database, an error message will pop out.
5. If that year is found in the database, the committee board effective on that year will be displayed in the form.



Input box to enter the Error message to inform

year to be searched. that the year was not found.

Clearing and resetting the form:

1. Click on the ‘Clear’ button.
2. All the textboxes shown on the form will be reset to blank.

Printing:

1. Click on the ‘Print’ button.
2. A print preview page will be generated.
3. Click on the print icon if the preview is fine.



The ‘Print’ icon.

The print preview.

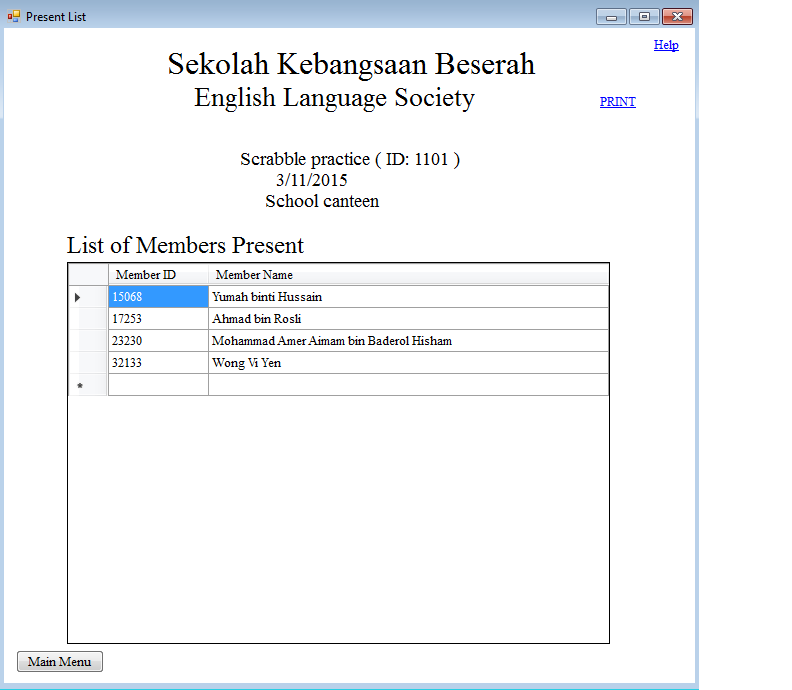
Return to main menu:

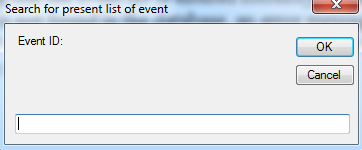
1. Click on the ‘Main Menu’ button.

Present List Form:

You are redirected here if the ‘Present List’ button was clicked on the main menu.

Generate present list of an event:

1. An input box will pop out requesting you to enter the ID of the event in which the present list is to be generated.
2. After confirming that the event ID is entered correctly, click on the ‘Ok’ button.
3. If that event ID is not found in the database, an error message will pop out.
4. If that event ID is found in the database, the present list of that event will be displayed on the form.



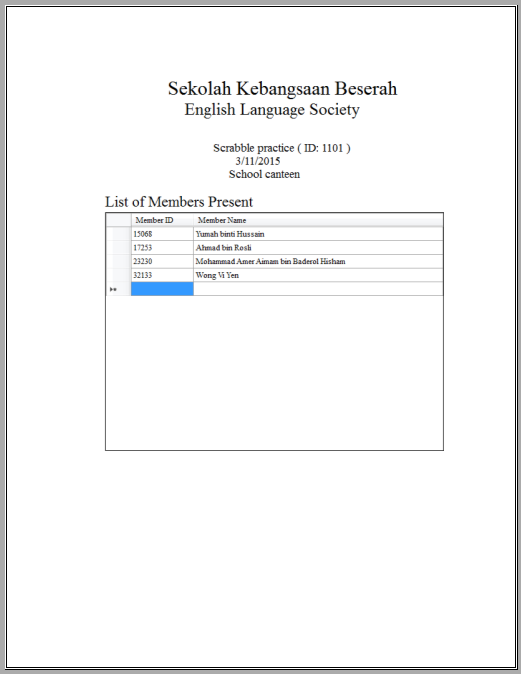
Input box to enter the ID of the event which its present list is to be generated.

List of members present and details

in the event in which the ID is entered.

Printing:

1. Click on the ‘Print’ button.
2. A print preview page will be generated.
3. Click on the print icon if the preview is fine.



The ‘Print’ icon.

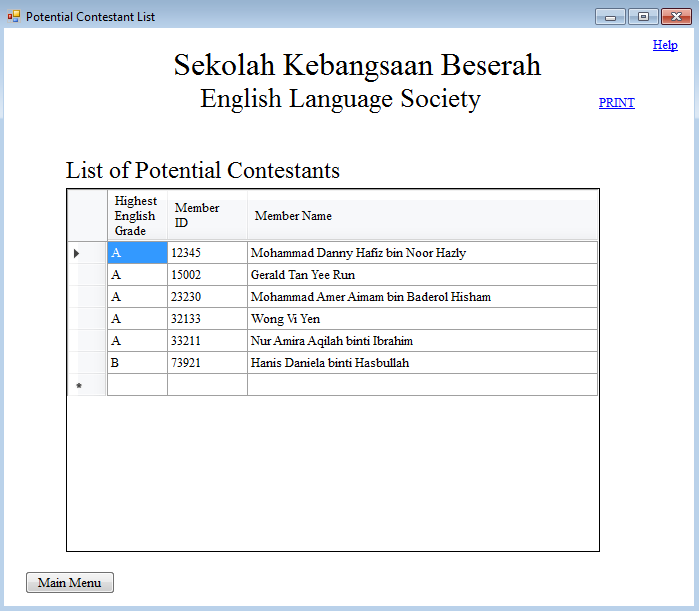
The print preview.

Return to main menu:

1. Click on the ‘Main Menu’ button.

Potential Contestant List Form:

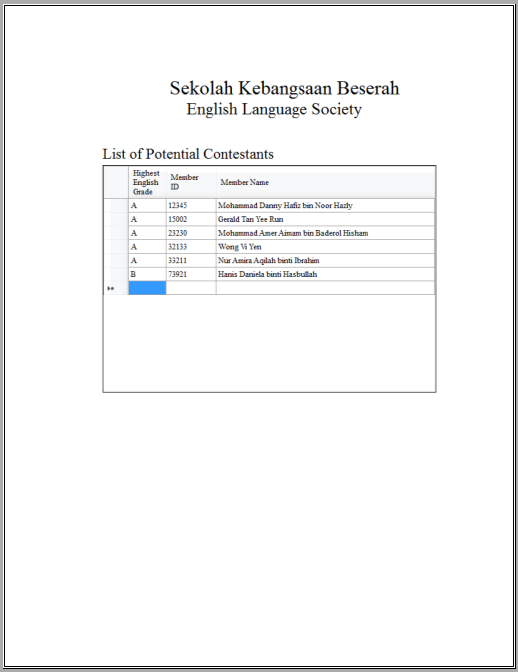
You are redirected here if the ‘Potential Contestant List’ button was clicked on the main menu.



The list of active members who are potential contestants, who achieved a highest English grade of A or B, will be generated once the form is loaded.

Printing:

1. Click on the ‘Print’ button.
2. A print preview page will be generated.
3. Click on the print icon if the preview is fine.



The ‘Print’ icon.

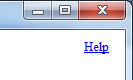
The print preview.

Return to main menu:

1. Click on the ‘Main Menu’ button.

On Screen Help

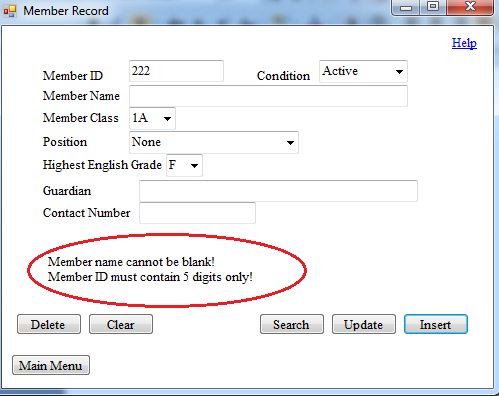
1. Click on the ‘Help’ link label on any form.
2. A help page will be generated on a web page.
3. Click on the title or section that you would like to get assistance on to direct to that page.



The ‘Help’ link label.

Common Error Guide:

Errors in Member Record Form



The image above shows the position of errors in this form.

First line of the error message is for errors that arise when certain fields are left blank.

id blank.PNG

Member ID:

id blank.PNGid blank.PNGMember Name:

id blank.PNGCondition:

id blank.PNGMember Class:

id blank.PNGPosition:

id blank.PNGHighest English Grade:

Guardian:

id blank.PNGContact Number:

Second line of the error message is for errors that arise when data filled in certain fields are wrong or not following the specifications.

id blank.PNGid blank.PNGMember ID:

id blank.PNGMember Name:

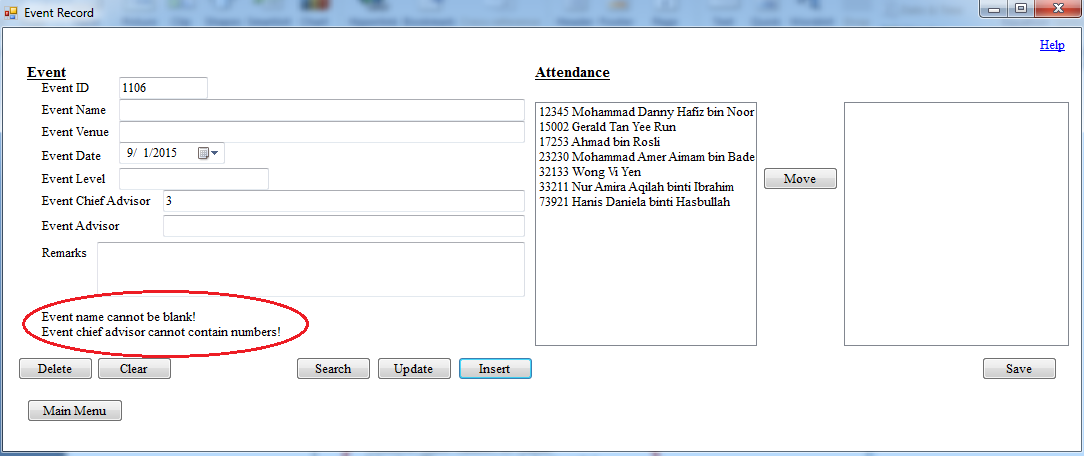
id blank.PNG

id blank.PNGGuardian:

id blank.PNG

id blank.PNGid blank.PNGContact Number:

id blank.PNG

Errors in Event Record and Event Attendance Form

The image above shows the position of errors in this form.

First line of the error message is for errors that arise when certain fields are left blank.

id blank.PNG

id blank.PNGEvent ID:

Event Name:

id blank.PNGid blank.PNGEvent Venue:

id blank.PNGEvent Level:

id blank.PNGEvent Chief Advisor:

Event Advisor:

Second line of the error message is for errors that arise when data filled in certain fields are wrong or not following the specifications.

id blank.PNGEvent Name:

id blank.PNGEvent Venue:

id blank.PNGid blank.PNGEvent Level:

id blank.PNGid blank.PNGEvent Chief Advisor:

id blank.PNG

id blank.PNGEvent Advisor:

id blank.PNG

Remark:

Errors in Event Record and Event Attendance Form

The image above shows the position of errors in this form.

First line of the error message is for errors that arise when certain fields are left blank.

id blank.PNG

Year:

id blank.PNGChief Advisor:

id blank.PNGid blank.PNGAdvisor 1:

id blank.PNGAdvisor 2:

Advisor 3:

Second line of the error message is for errors that arise when data filled in certain fields are wrong or not following the specifications.

id blank.PNGYear:

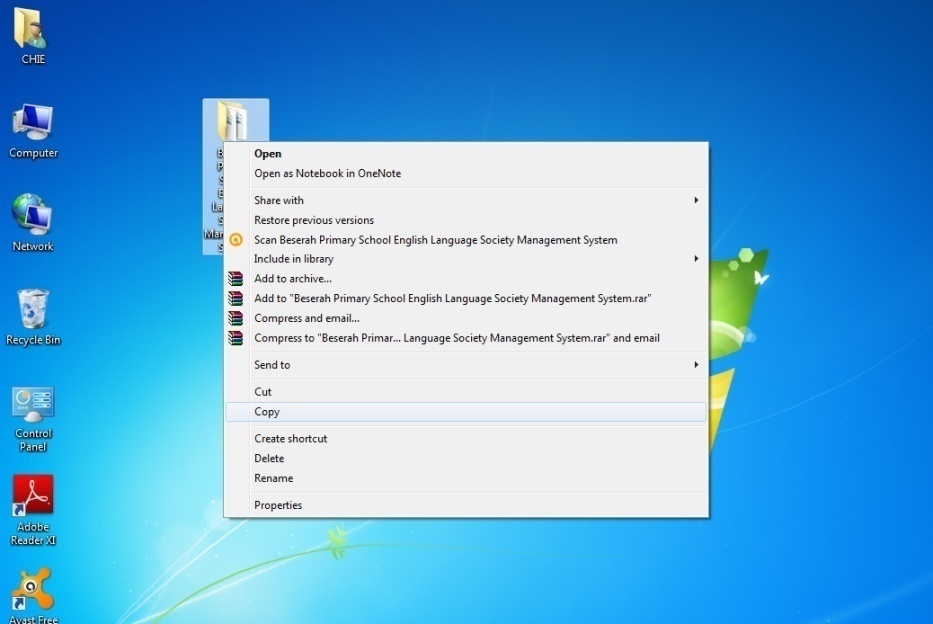
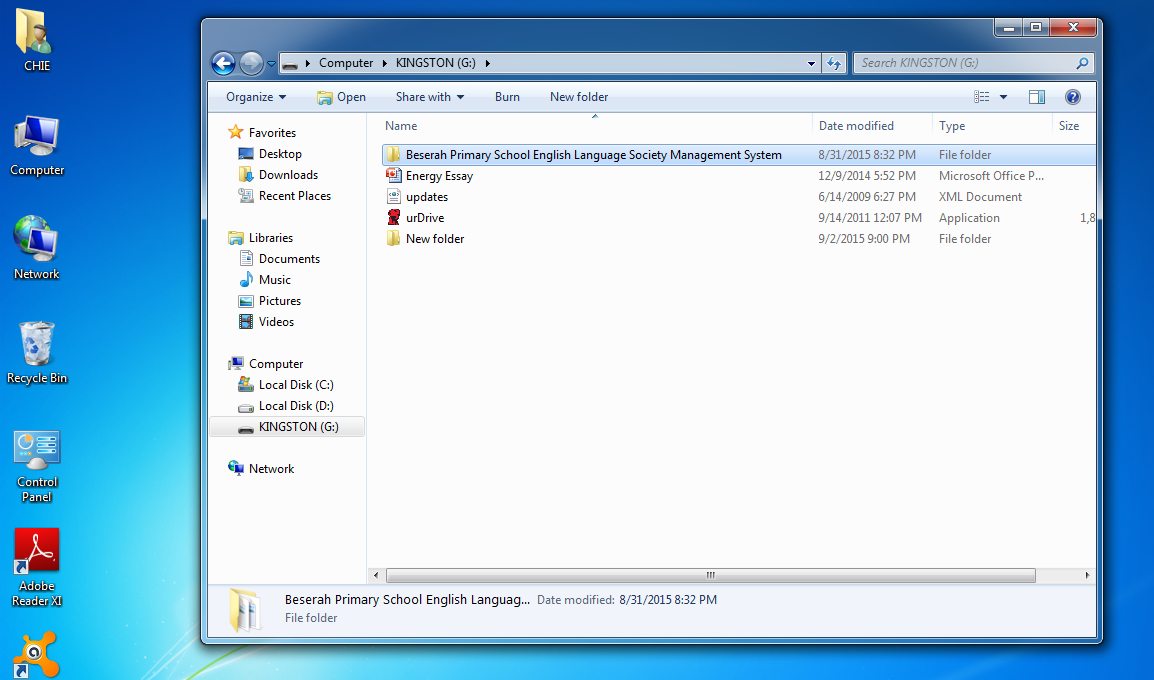
id blank.PNGid blank.PNGChief Advisor:

id blank.PNGid blank.PNGAdvisor 1:

id blank.PNGid blank.PNGAdvisor 2:

id blank.PNGid blank.PNGAdvisor 3:

**Backup Routine**

1. Copy the Beserah Primary School English Language Society Management System folder on the Desktop.
2. Go to Computer, then to the CD/ DVD/ external hard disk/ pendrive’s folder, and paste it.
3. Keep the external backing storage (preferably flash drives/ pendrives) away from the laptop. It is recommended to back up the system and database every week or more frequent.

To restore the backup file from the backing storage, refer to the Installing of System and Starting the Programme section from page 185.

**Glossary**

The following table contains several terms which should be known when reading the user guide:

|  |  |  |
| --- | --- | --- |
| **Number** | **Term** | **Definition** |
| 1 | Textbox | The space to input data. |
| 2 | Form | The screen to input data. |
| 3 | Record | A group of related fields, regarding a particular object or person. |
| 4 | Shortcut | A file that allows another file or device or service to be linked and opened faster and easier. |
| 5 | Backup | A copy of the database and software in a place or device other than the main device. |
| 6 | Message box | A box which pops out to inform status of operations. |

**Index**

Introduction to the System 185

Installing of System and Starting the Programme 185

Using the System

Basic functions of forms: Main Menu 186

Member Record Form 187

Event Record and Event Attendance Form 190

Committee Board Form 194

Present List Form 197

Potential Contestant List Form 199

On Screen Help 201

Common Error Guide: Member Record Form 202

Event Record and Event Attendance Form 204

Committee Board Form 206

Backup Routine 208

Glossary 209

**(F) Evaluation**

1. **Discussion of the Degree of Success in Meeting the Original Objectives**

In order to determine the level or degree of success of the Beserah Primary School English Language Society designed, I compared the final product against the requirements agreed upon in the Requirement Specifications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Objective** | **Evaluation** | **Results** | **Evidence (page)** |
| a | Able to register or insert, update, delete and search for the member details | The system has been designed to be able to register or insert, update, delete and search for the member details | Success | 163, 164 |
| b | Able to register or insert, update, delete and search for the event details | The system has been designed to be able to register or insert, update, delete and search for the event details | Success | 165, 166 |
| c | Able to take attendance of members for a certain event | The system has been designed to be able to take attendance of members for a certain event | Success | 165 |
| d | Able to search for the attendance list of members for a certain event | The attendance list of members for a certain event was searched and generated on-screen successfully | Success | 165 |
| e | Able to insert, update and search for and generate the committee board of a certain year and print it | The system has been designed to be able to insert, update and search for and generate the committee board of a certain year and print it | Success | 167, 168 |
| f | Able to generate a list of members present for a certain event and print it | The system has been designed to be able to generate a list of members present for a certain event and print it | Success | 169 |
| g | Able to generate a potential contestant list and print it | The system has been designed to be able to generate a potential contestant list and print it | Success | 170 |

1. **Discussion of the Degree of Success in Meeting the Original Objectives**

**Client’s Response**

Miss Narulhuda had prepared a letter to state her thoughts and feelings regarding the new system designed.



**Analysis of Client’s Response**

Based on the letter written by Miss Narulhuda, the client is satisfied with the system produced which is capable of performing all the tasks listed as the objectives to be fulfilled. It also fulfills all the requirements stated previously.

Miss Narulhuda also commented on the new system when I contacted her to ask about the condition of their society which had implemented the new system. She stated that:

* The objectives agreed upon were achieved and the system functioned smoothly without any problem.
* The designed input and output interfaces are simple and easy to use which is convenient for them to learn and operate the system.
* The system has yet to suffer from lag so far.
* The on-screen help was easy to use and has clear instructions for them to follow.

Also, she gave her thoughts on what should be improved in the new system:

* Interfaces for the same purpose, such as input or output, should have the same colour so that any mistake can be detected easily.