Software Engineering Lab

Eng. Iman Haidar

******

Hadi Abdelmajid

201801273

***Lebanese School Management System***

Description:

***project overview:***

***LSMS*** Provides school employees (IT managers, accountants ...) with a platform that enables them to transcribe school related data such as (students financial or academical issues etc.., parents phone numbers, addresses etc.…, employees whether they’re full timers/part timers and their salaries...). In addition to that, ***LSMS*** supply the concerned parties with a schedule maker and financial analyzer in order to perform balance-sheets. ***LSMS*** has a Grading system that allows teachers and coordinators to examine student’s improvements through graphical and analytical aspects. It also validates employee’s appointment’s attendance.

***Purpose:*** Facilitates schools management systems.

***Scope of work:*** The environment of the application is divided among:

* *Finance department:* Student and employees financial data entry.
* *Teachers and coordinators:* Grades analysis and entry and schedule modification.
* *Academic department:* inputs family’s, students, employees and teacher’s personal information

***Product Scenario:*** The employee should log in with his/her username and password and according to his/her scope of work, he/she will import or export data.

***Stakeholders:*** User of the application

* *User name/category:* All the school employees all over Lebanon
* *User role:* Import, export and analyze data
* *Subject matter experience*: Short ***LSMS*** training workshop
* *Technological experience:* User should know how to use pc.

***Mandated constraints*:**

The software is big and requires high security according to the data it holds. It has many subparts and needs database. Moreover, data should be updated frequently and it should be well advertised in order to spread the system all over the schools in Lebanon. Weekly and monthly maintenance should be available for the user. Finally, the software should be available on a common server where all the employees are connected. ***Product use cases:*** *Diagram*

***LSMS***

IT employee

Financial manager

Teachers

And Coordinators

General Manager

* ***The financial manager***: is responsible to manage all financial issues for instance student’s tuition fees or employee’s salary, balance sheet, school’s cost and expenses.
* ***Teachers and coordinators***: are responsible for applying all grading measures. To add, coordinators are in charge of creating schedules for teachers and student.
* ***IT employee*** manages personal data entry for student’s background (family’s information, age, address, email, phone number, time to graduate etc...). Moreover, teacher’s personal information (Phone number, email, address type of work “full time, part time “).
* ***General manager*** can access on all 5 sections and to keep an eye on employees attending time.

***Dependability requirement’s***

Reliability and Availability requirements:

* This software shall be available for a minimum of 10 hours/day, 365 days / year.
* The product should be available for use from 07:00 am until 16:00 pm.
* The product shall achieve an up-time of 85%.
* The software should have a fail / safe requirement in which it may fail but it should do that safely (No data loss!)

***Adaptability requirements:***

1. This software should run under all versions of windows, IOS, LINUX.
2. The product should be designed to be worked in offices.

***Robustness or Fault Tolerance Requirements:***

* The product should continue to perform in local mode whenever it loses its link to the server.
* The software shall not be electricity dependent in which it shouldn’t shut down when the electricity turns off.

***Software Model:***

The software Reuse model is chosen for this software and that’s due to:

* Reduce the development and implementation time due to the size of the project.
* Benefit from the already implemented codes available to similar softwares due to the similarities that are common in all such softwares
* Have more time to implement new aspects
* Enable the software to evolve throughout the years due to business, customer and new technologies.
* We don’t have the availability of several developers
* Reduce software and maintenance costs.

Class Diagram

Design

|  |
| --- |
| Admin |
| Id:Int  Name:String  Pass:char |
| Functions:  +Login()  +logout()  +AddNewTeachers()  +ModifyTeachers()  +AddNewStudent()  +ModifyStudent()  +AddNewClass()  +ModifyClass()  +AddAdmissionOffice()  +AddAccoutant()  +AddHR |

|  |
| --- |
| Teacher |
| Id:Int  Name:String  Phno:int  Add:String |
| Functions:  +MarkAttendance()  +AddSchedule()  +ModifySchedule() |

|  |
| --- |
| Student |
| Id:Int  Name:String  Phno:int  Add:String |
| Functions:  +FillAddmissionForm()  +GetEnrolled() |

|  |
| --- |
| Class |
| Id:Int  Name:String |
| +DisplayClass() |

|  |
| --- |
| Accountant |
| Name:String  Pass:char  Phno:int  Add:String |
| Functions:  +DisplayBalanceSheet()  +ModifyBalanceSheet() |

|  |
| --- |
| Human resources |
| Name:String  Pass:char  Phno:int  Add:String |
| Functions:  +DisplayPayroll()  +ModifyPayroll() |

|  |
| --- |
| Admission’s Office |
| Id:Int  Name:String  Pass:char |
| +DataEntry()  +DataExport()  +ModifyData() |

Sequence Diagram

Admission office

Teacher

Class

Student

Admin

**Add/ModifyStudent() DisplayClass() DataEntry()**

**Return**

**AddAdmissionOffice()**

**AddNewTeacher()**

**Return**

**Add/ModifyClass()**

**Return**

**Mark Attendance**

**Return** **Return**

**PrepareReportCard()**

**Return**

Human Reso

Accountant

**AddHR()**

**AddAccountant()** **DisplayBalanceSheet()** **DisplayPayroll()**

State Machine UML Diagram

**Data Entry**

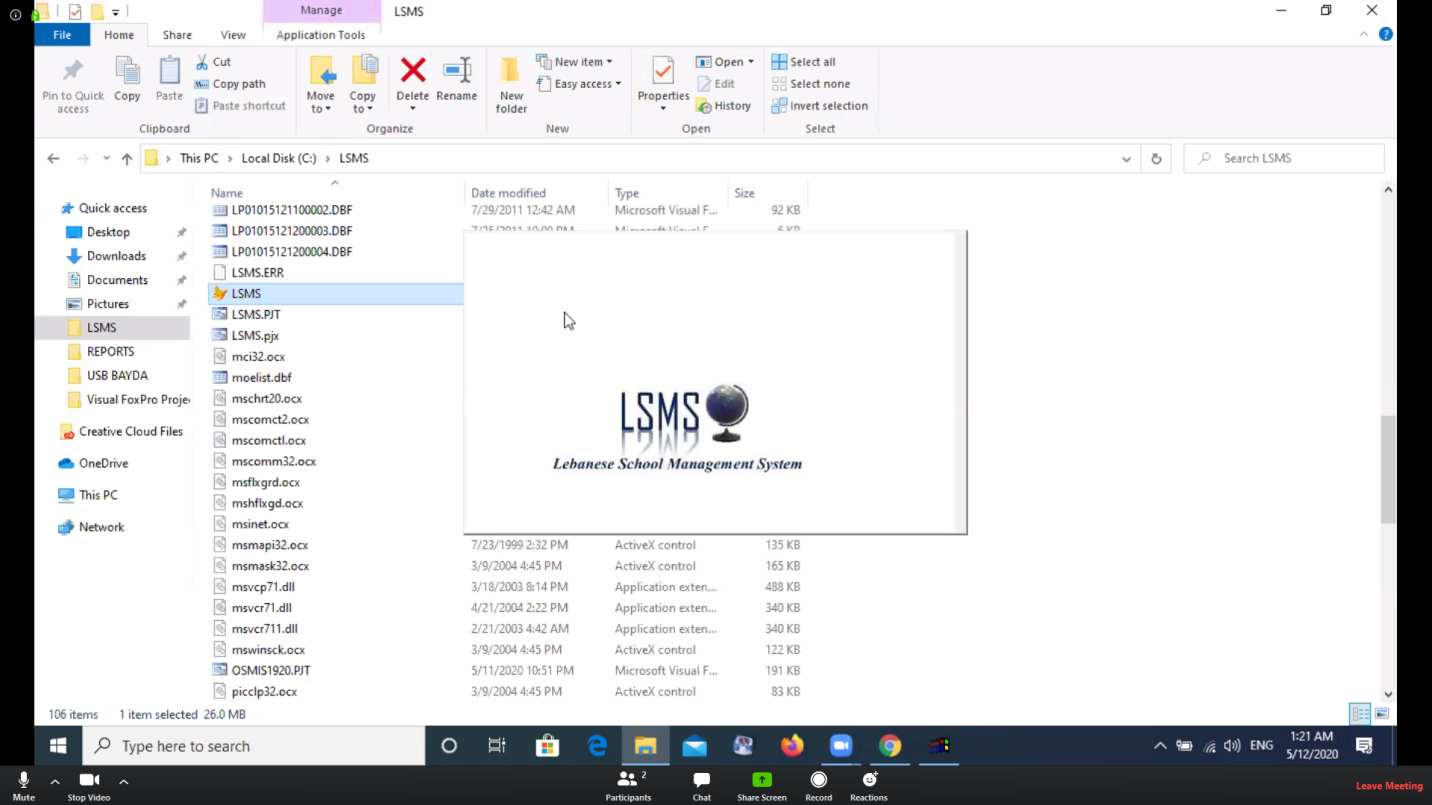
Data Export

|  |  |
| --- | --- |
| Login | Data type |
| Teacher | Grades/Schedule/attendance |
| Admission Office | Student/Family/Teacher (personal information) |
| Human resources (HR) | Payroll |
| Accountant | Balance Sheet |

Implementation:

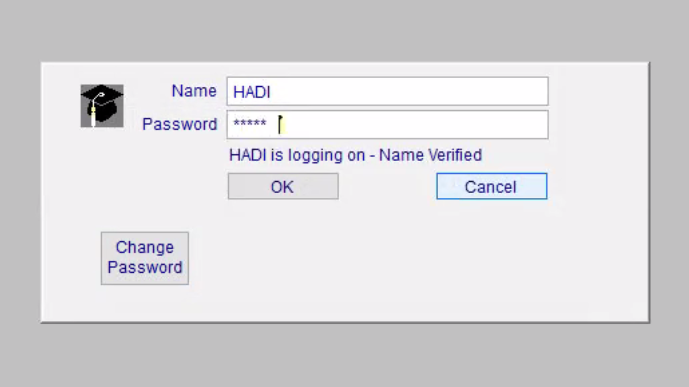
This software has been done using Microsoft Visual SoftPro.

**Executable version of LSMS:(All the data provided are fake data corresponding to a fake school ( The Future HighSchool) in order to facilitates and shows the software’s functionality.**

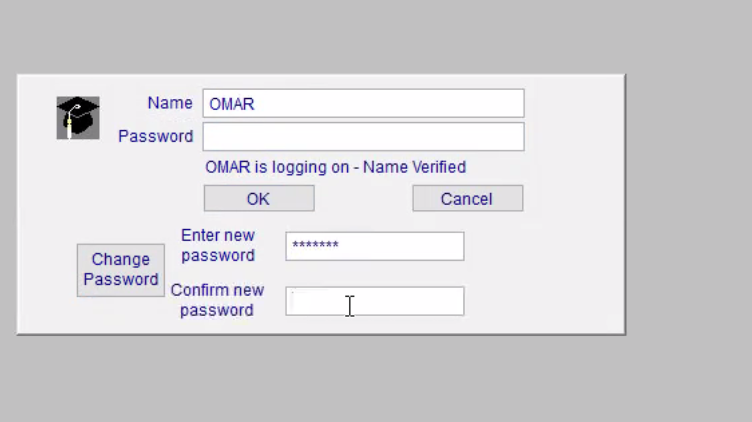


Loading Screen

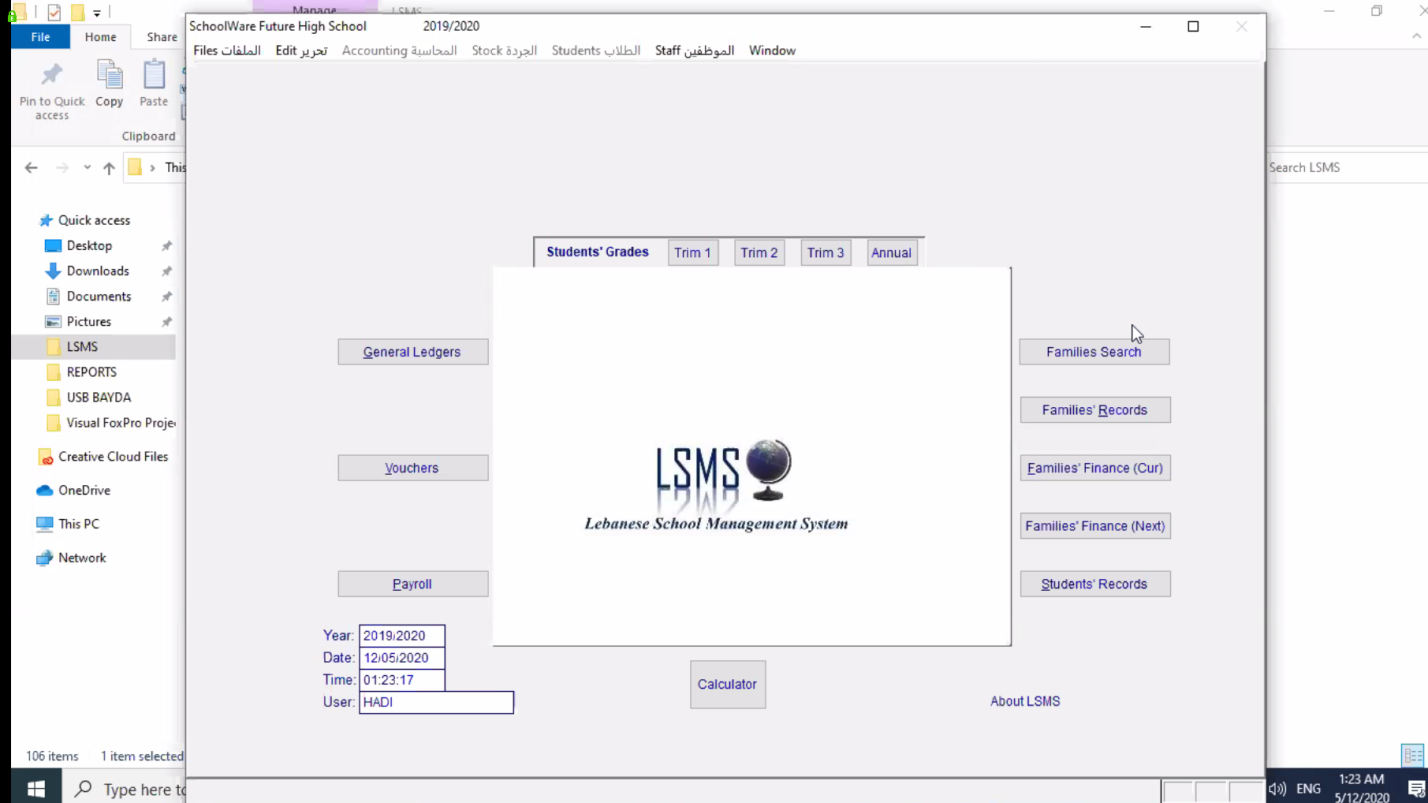
This capture will welcome the user once clicking on the software’s icon.



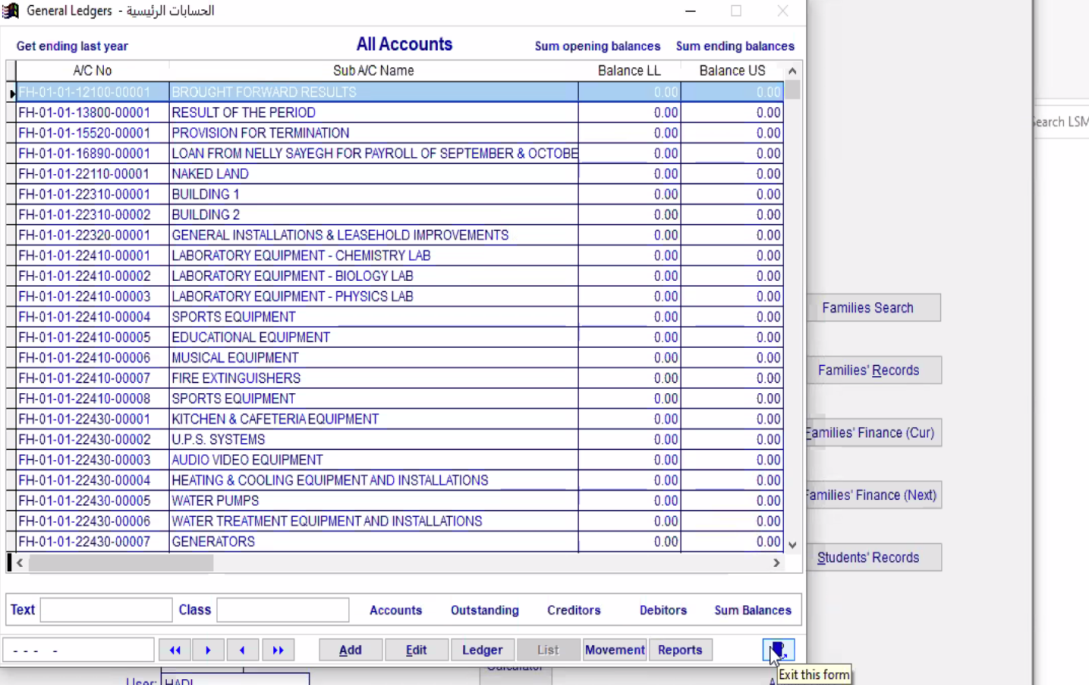
The Login phase : Each employee will have access to the software through his/her username and password



This command allows the user to change his/her password when needed and the new password will be updated in the database automatically.

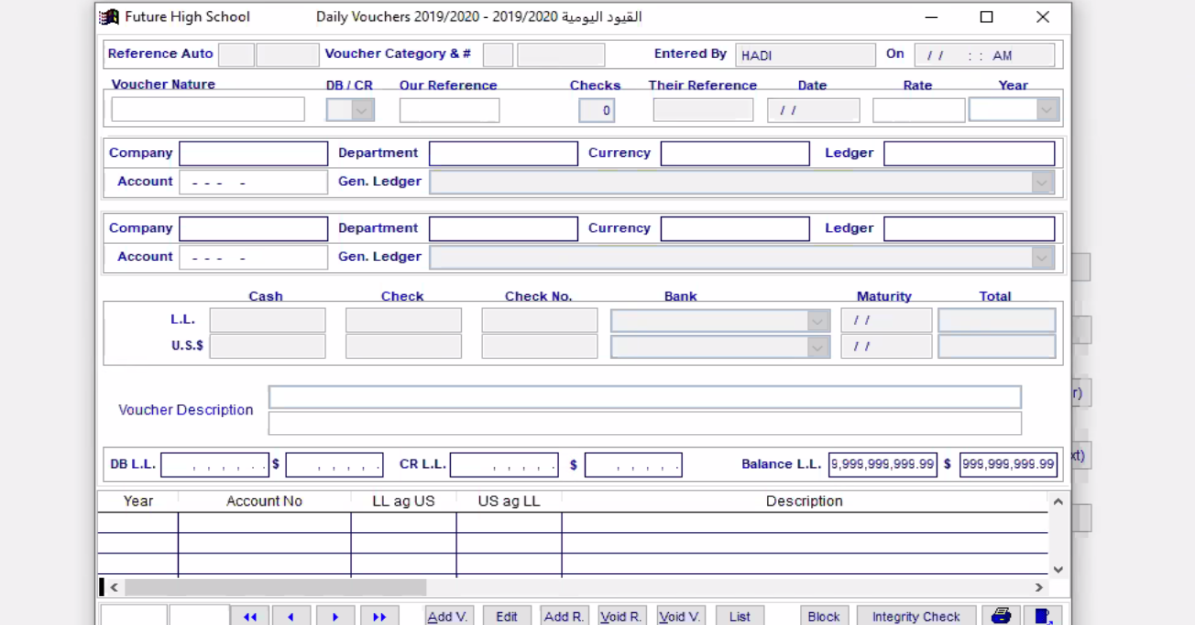


This is the main page of the software. That includes all the sections that serves all the employees Listed in the description section. On the left bottom side of the main page the software shows which user is currently logged in ,date&time.

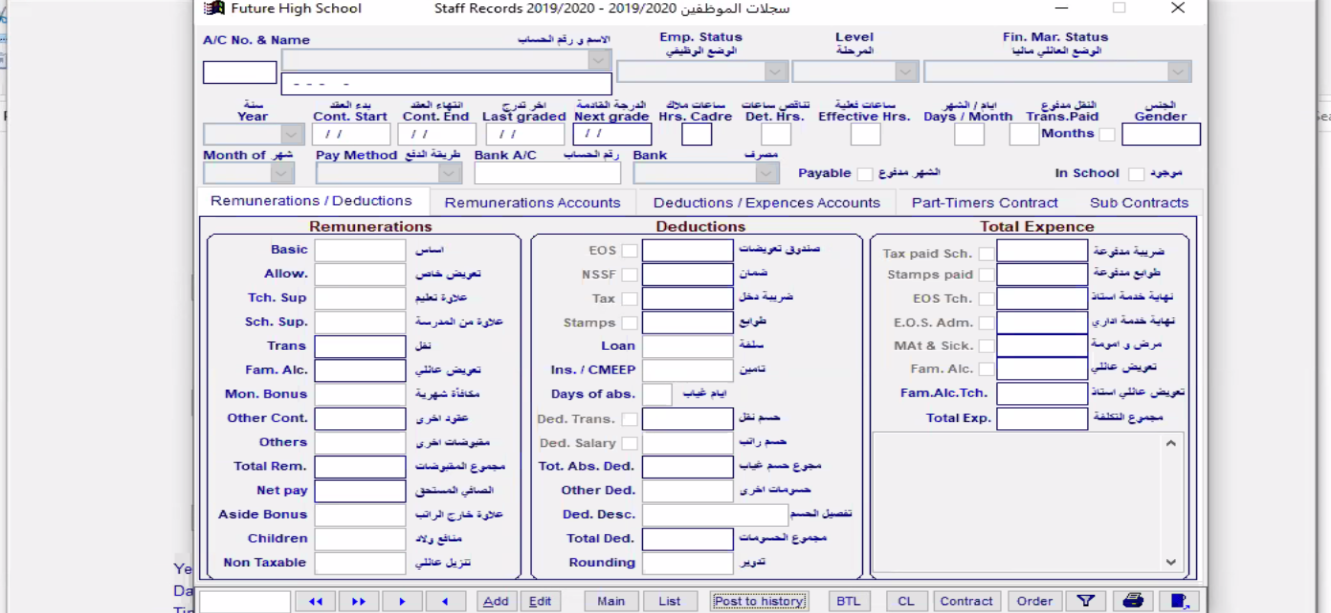


This tab is included in all the sections and used to exit them

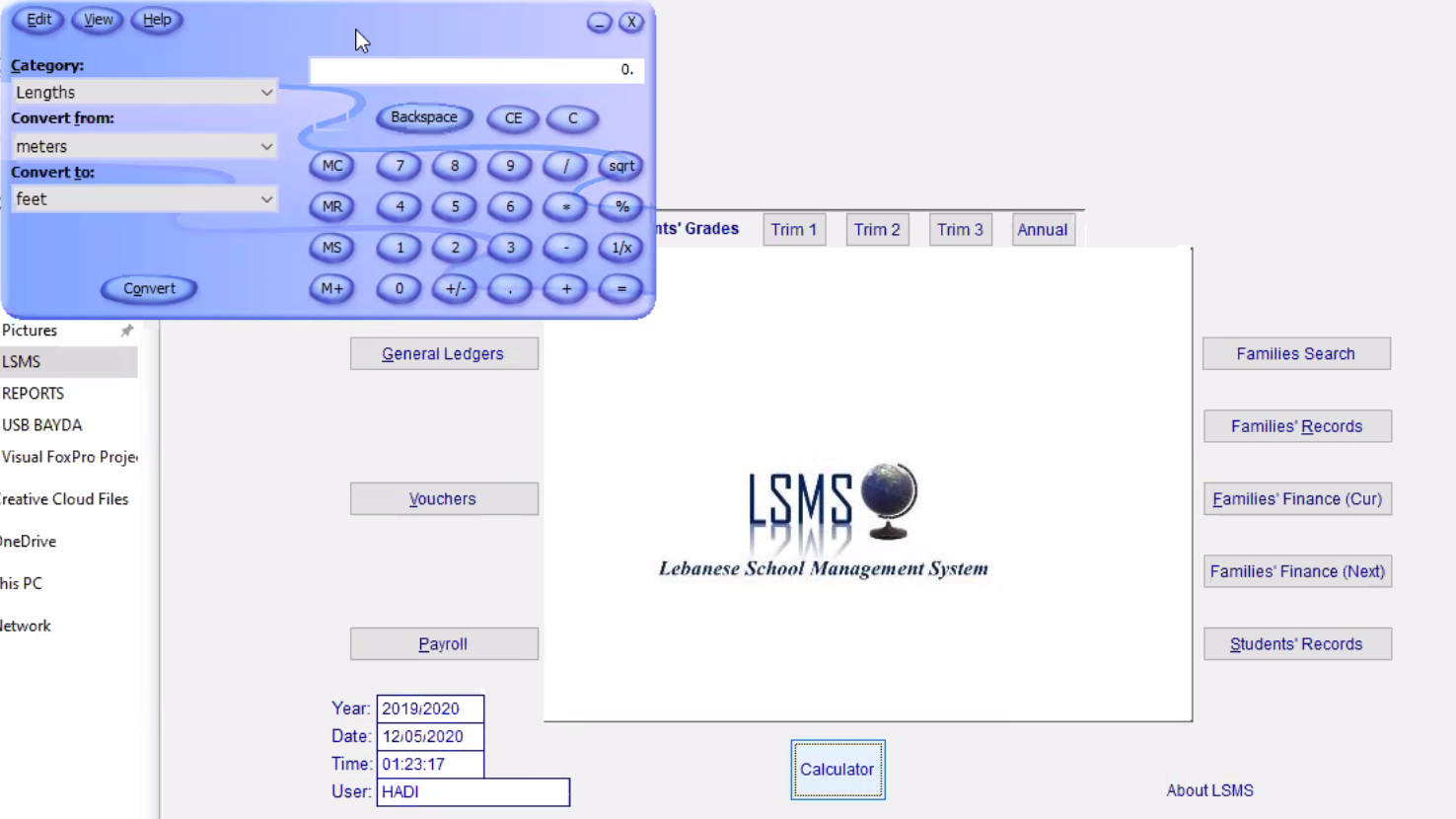
General ledgers for the accounting department.



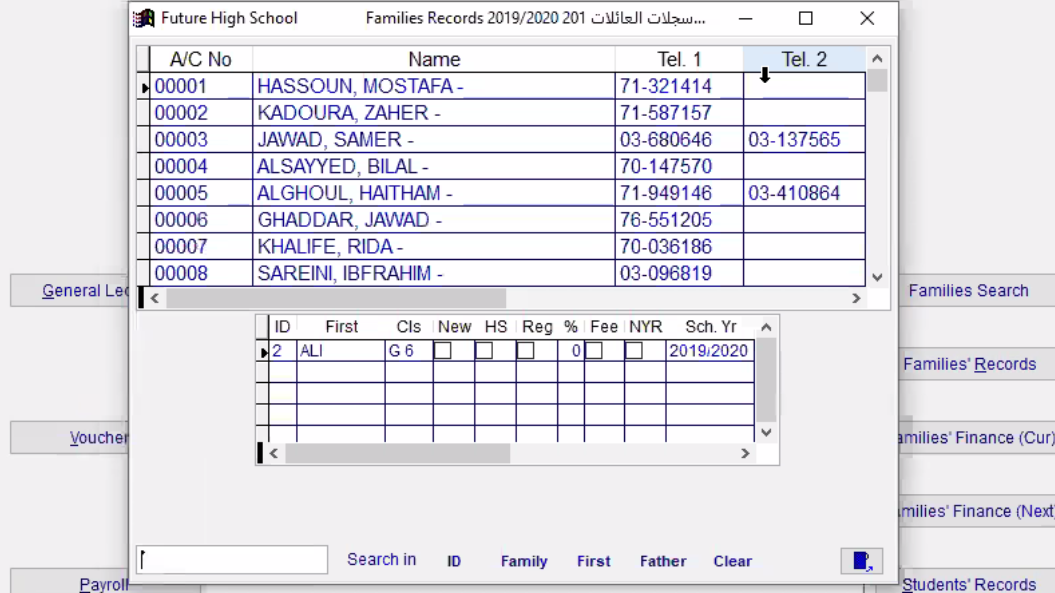
Vouchers is used for receipts and any payment method.



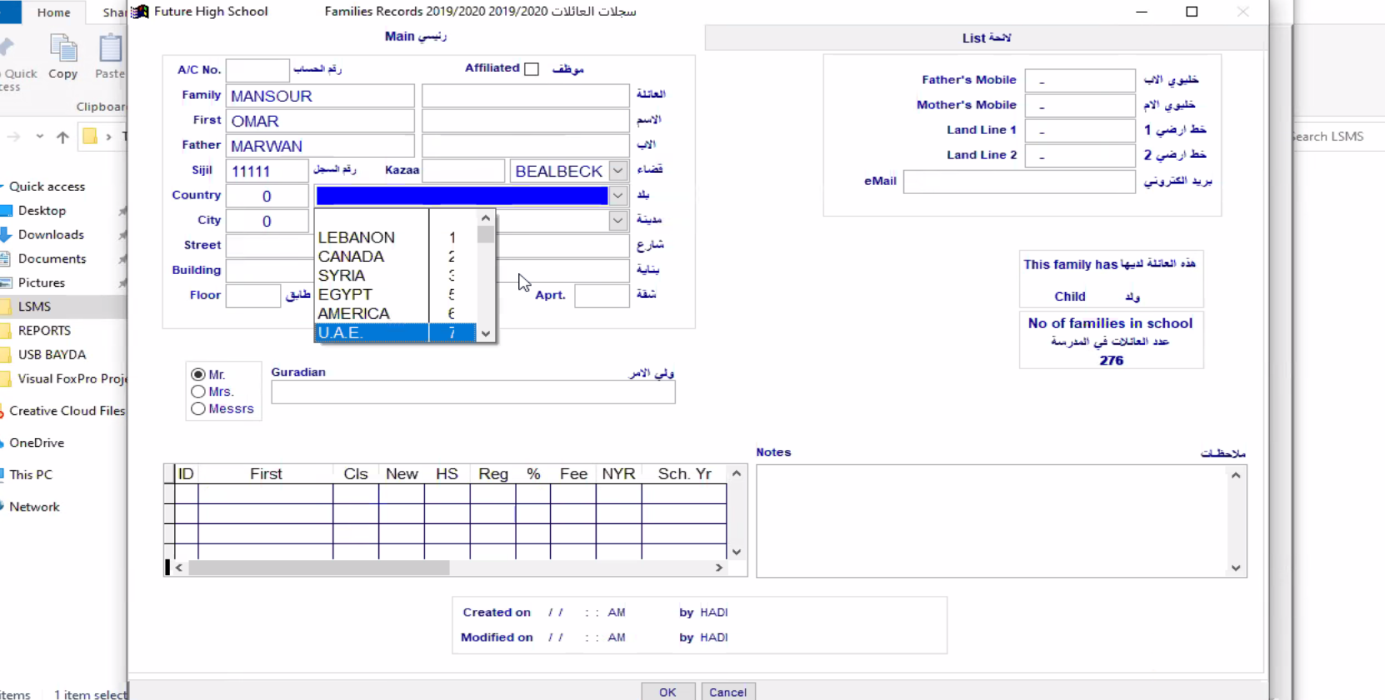
School payroll , including all staff and employees’ salaries



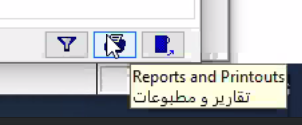
Clicking on calculator box, will open a calculator tab for fast calculations.



Families search: Through this search engine you can look for a certain family information through their first,last names and ID.



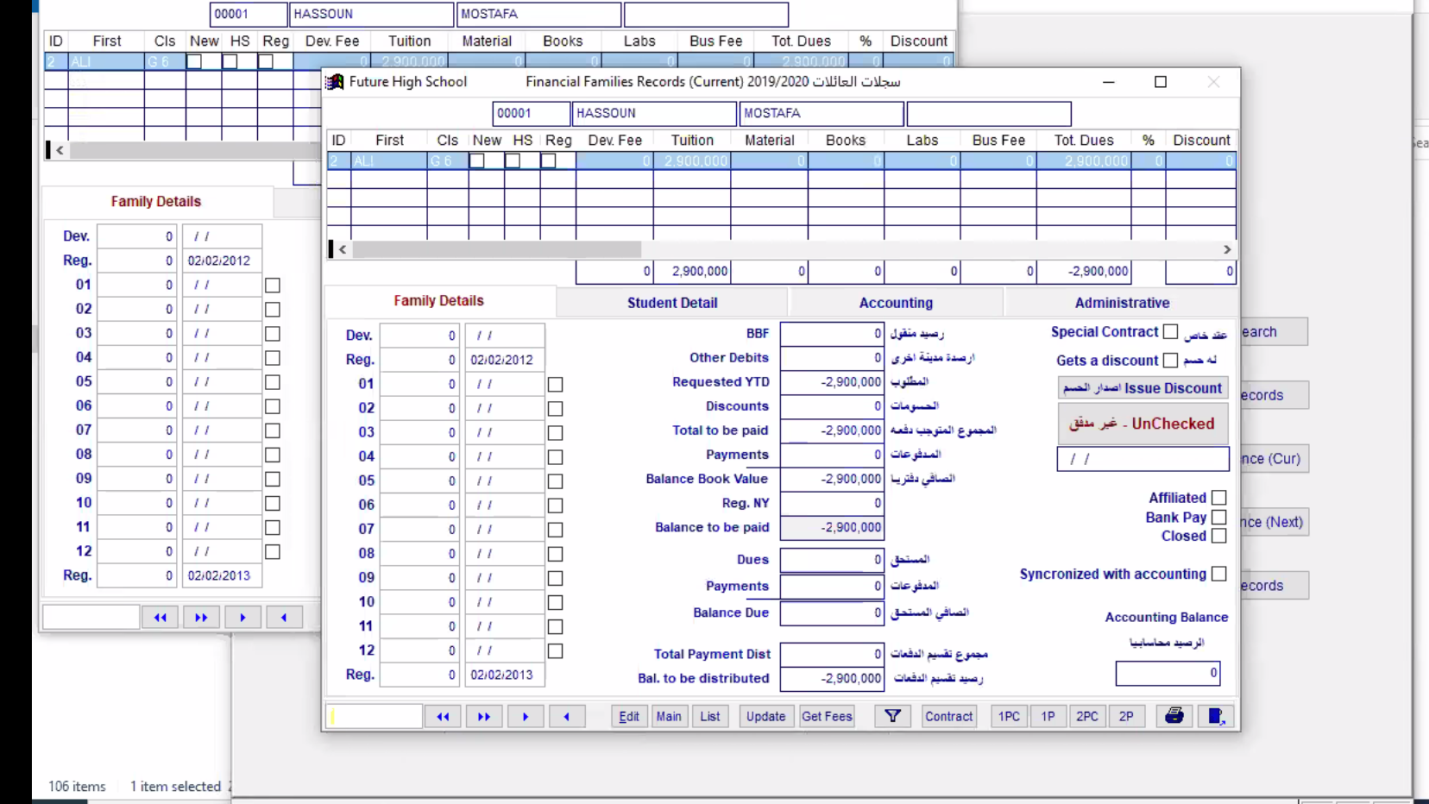
Family records: Using this button we can add new family records with all information needed.



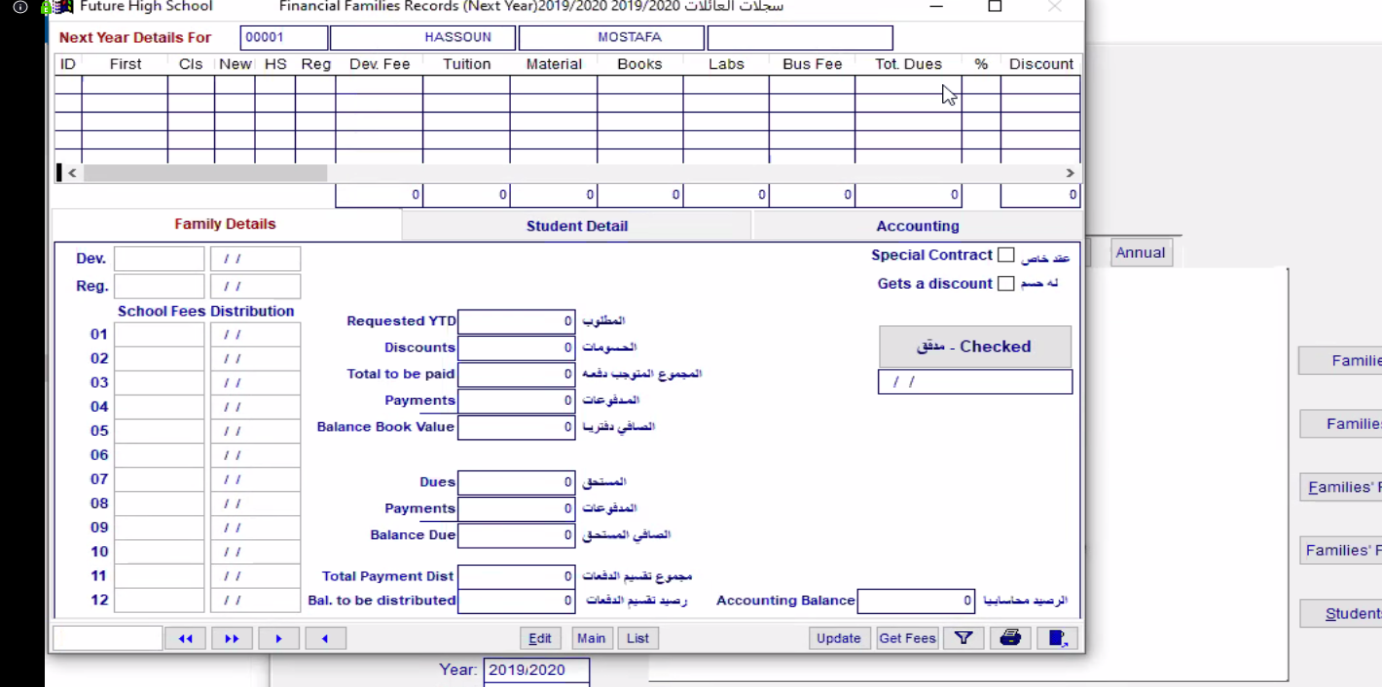
Filter is used to filter family records by their names ,family names ..

Reports and printouts is used to print out the opened file

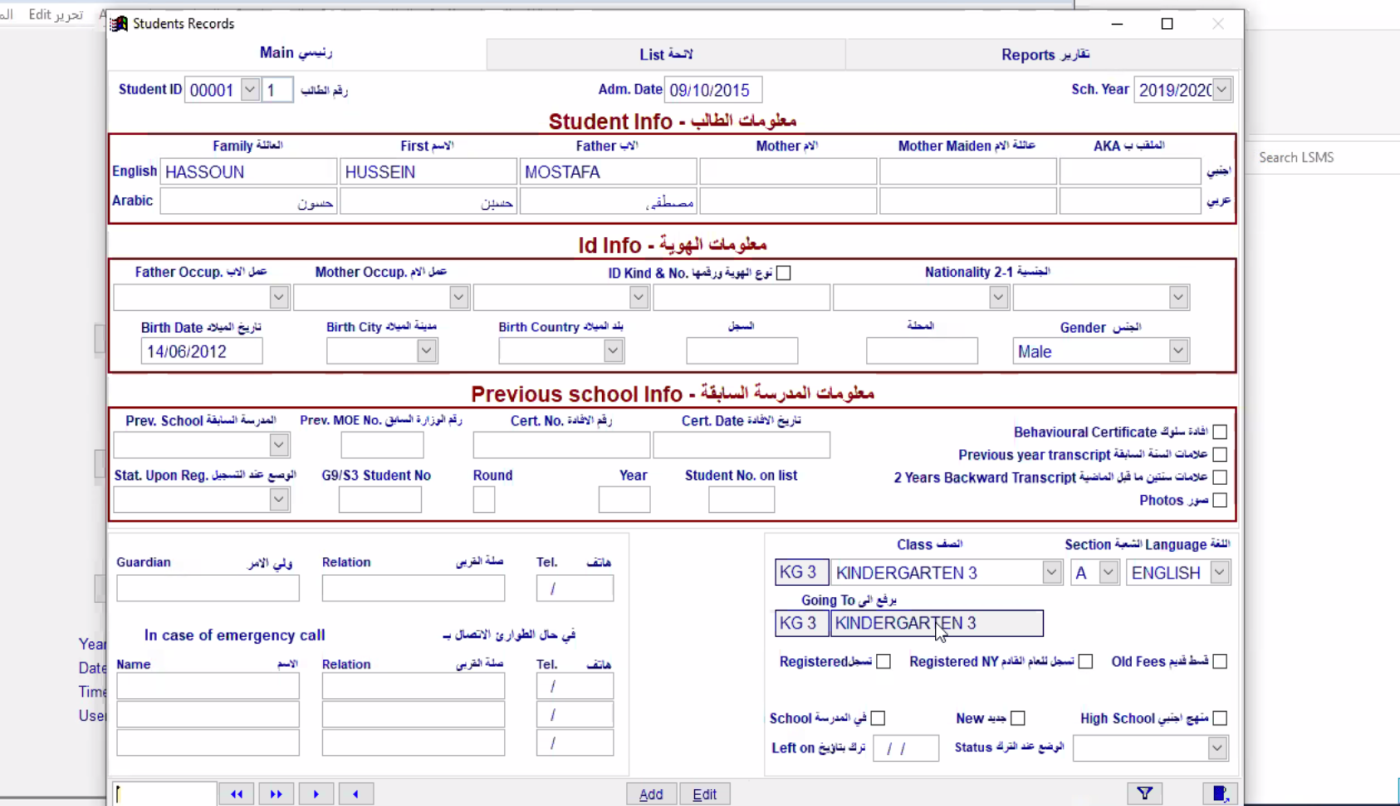
Qqqq



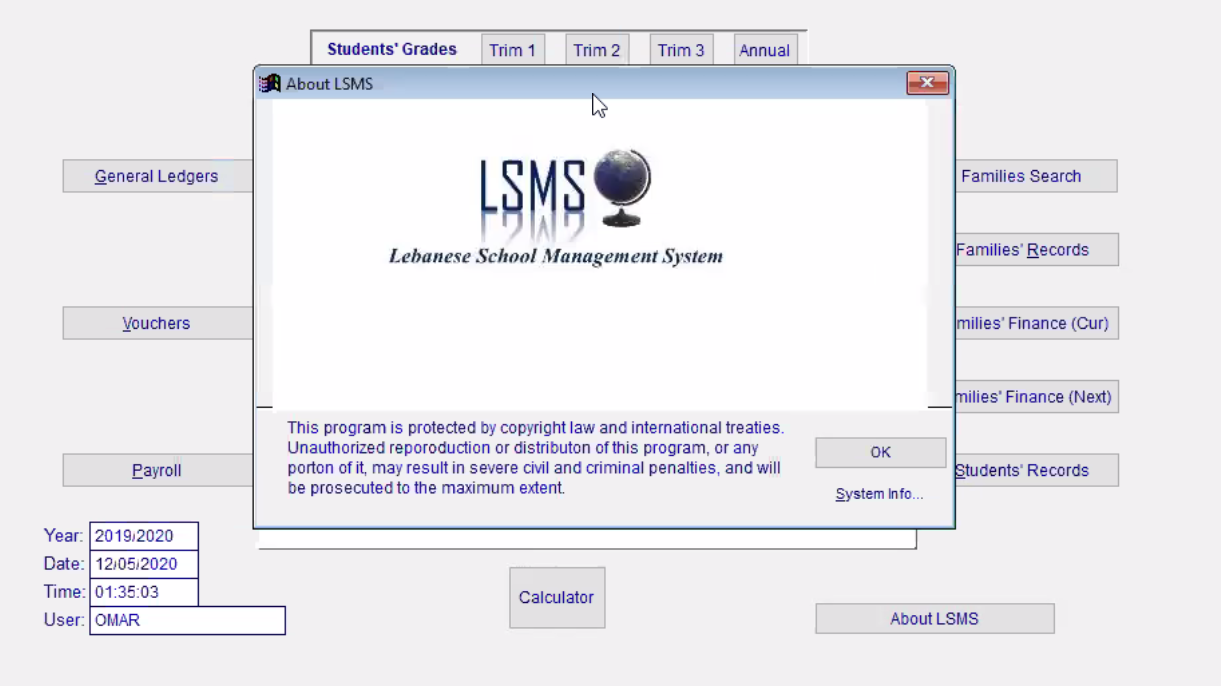
Family finance for the current year: Shows all family finance issues for the current year (ex: tuition fees)



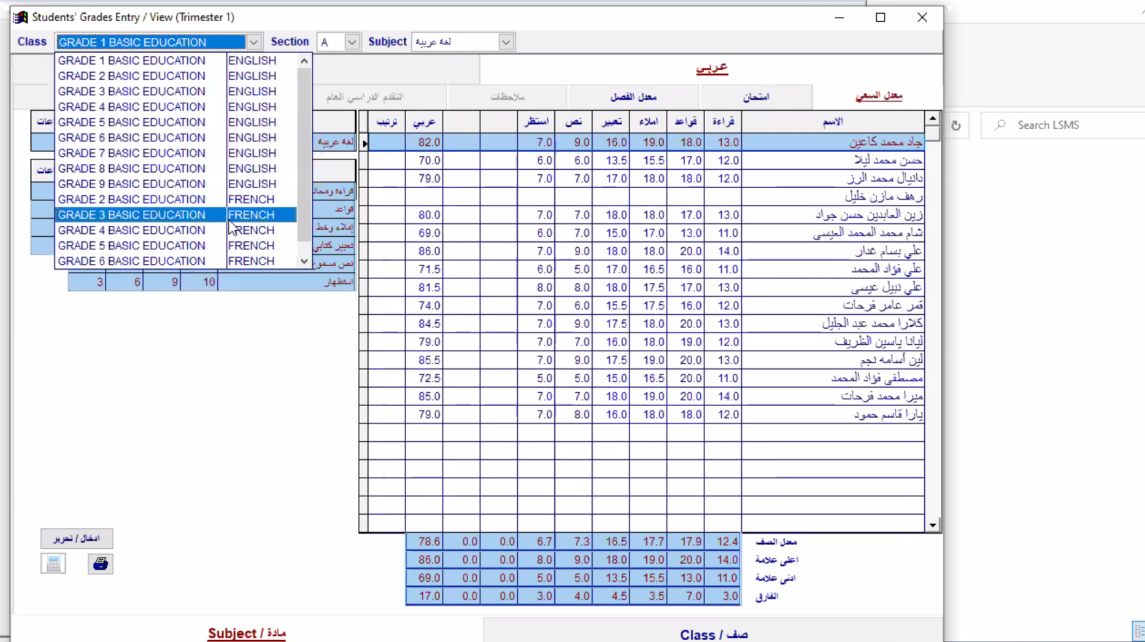
Family finance for the following year: similarly ,shows all the finance issues for the next year(ex:tuition fees)



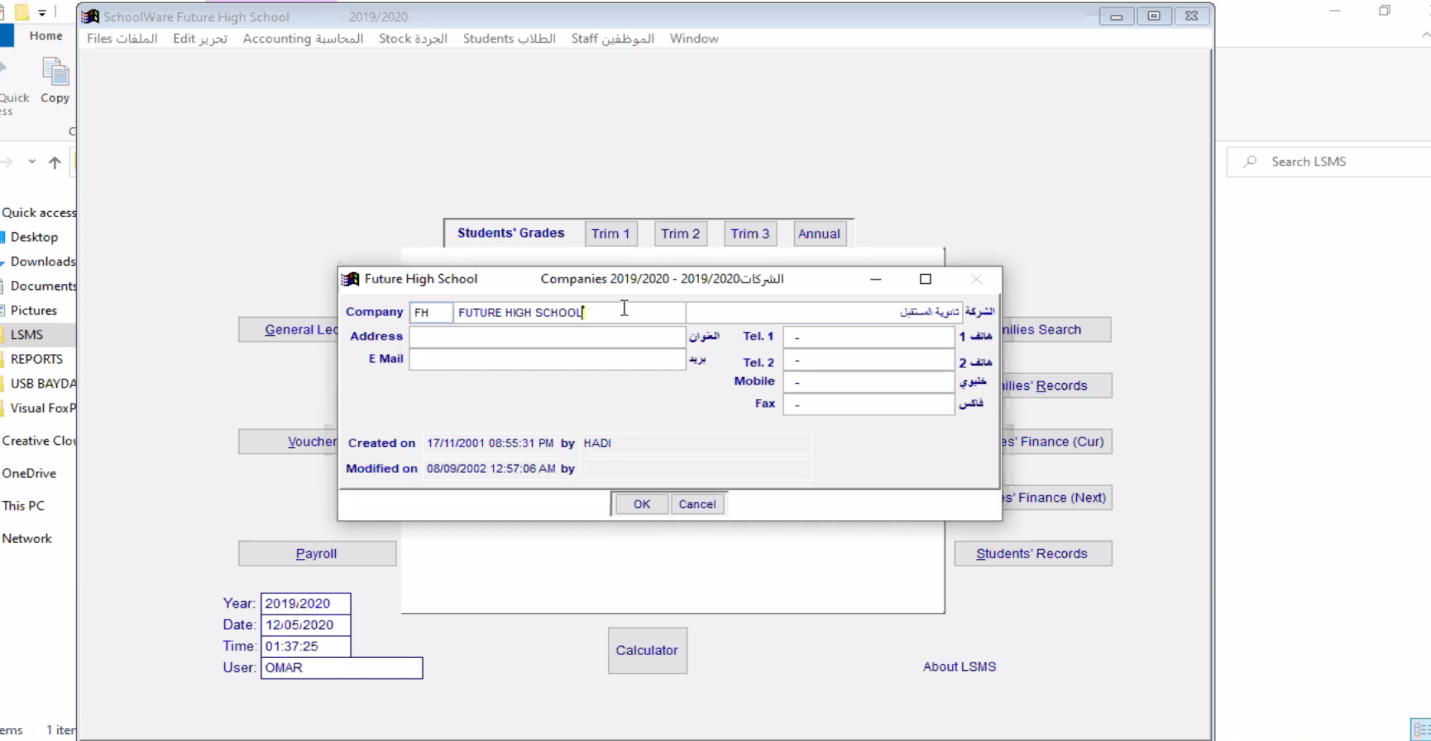
Students Record : In this tab we can import and export all student records and information



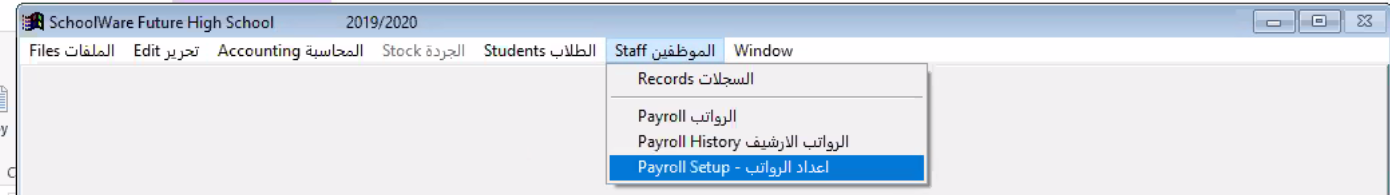
About LSMS



Grading: On this tab we can choose the class/section and subject , to enter their grades and averages

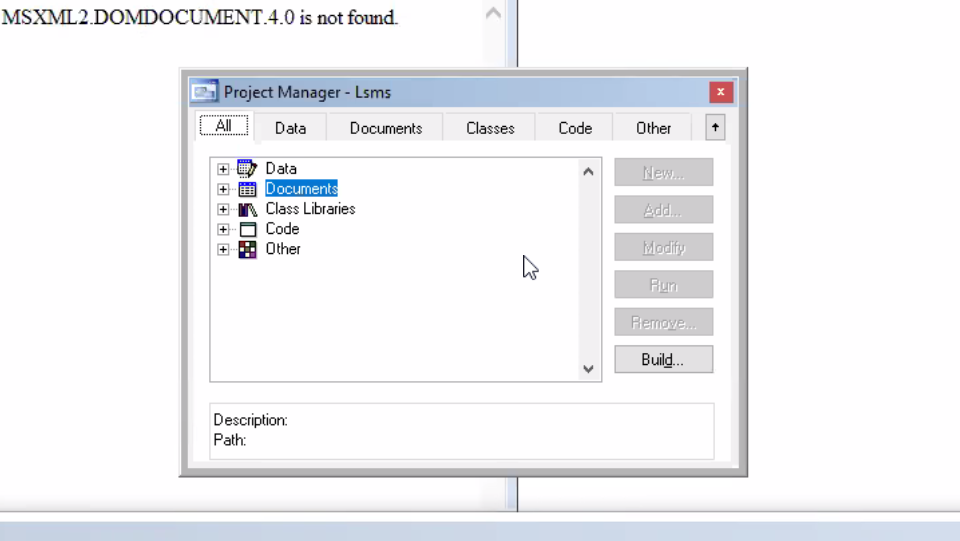


By clicking Files->Companies for example : This tab is used to add any company that deals with the school(could be used also in the vouchers’ section

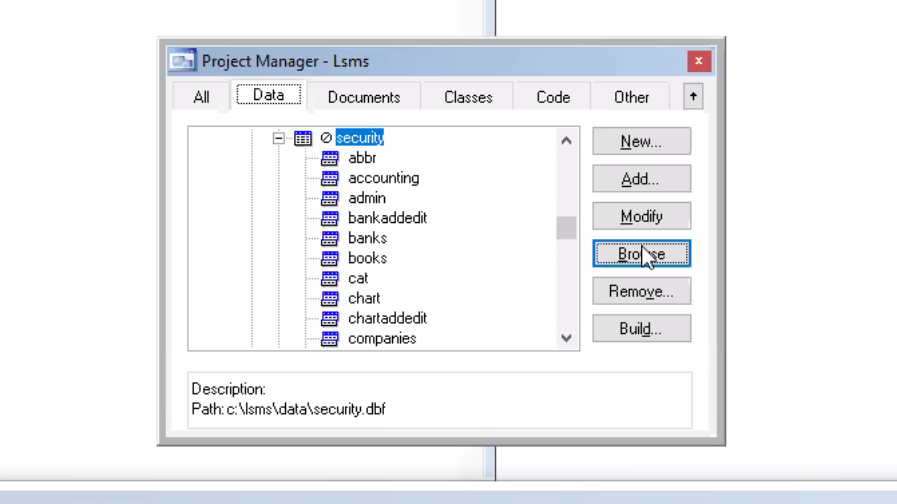


All types of software users(Accountant ,student,teachers..) can find all what they are concerned with in their own section.

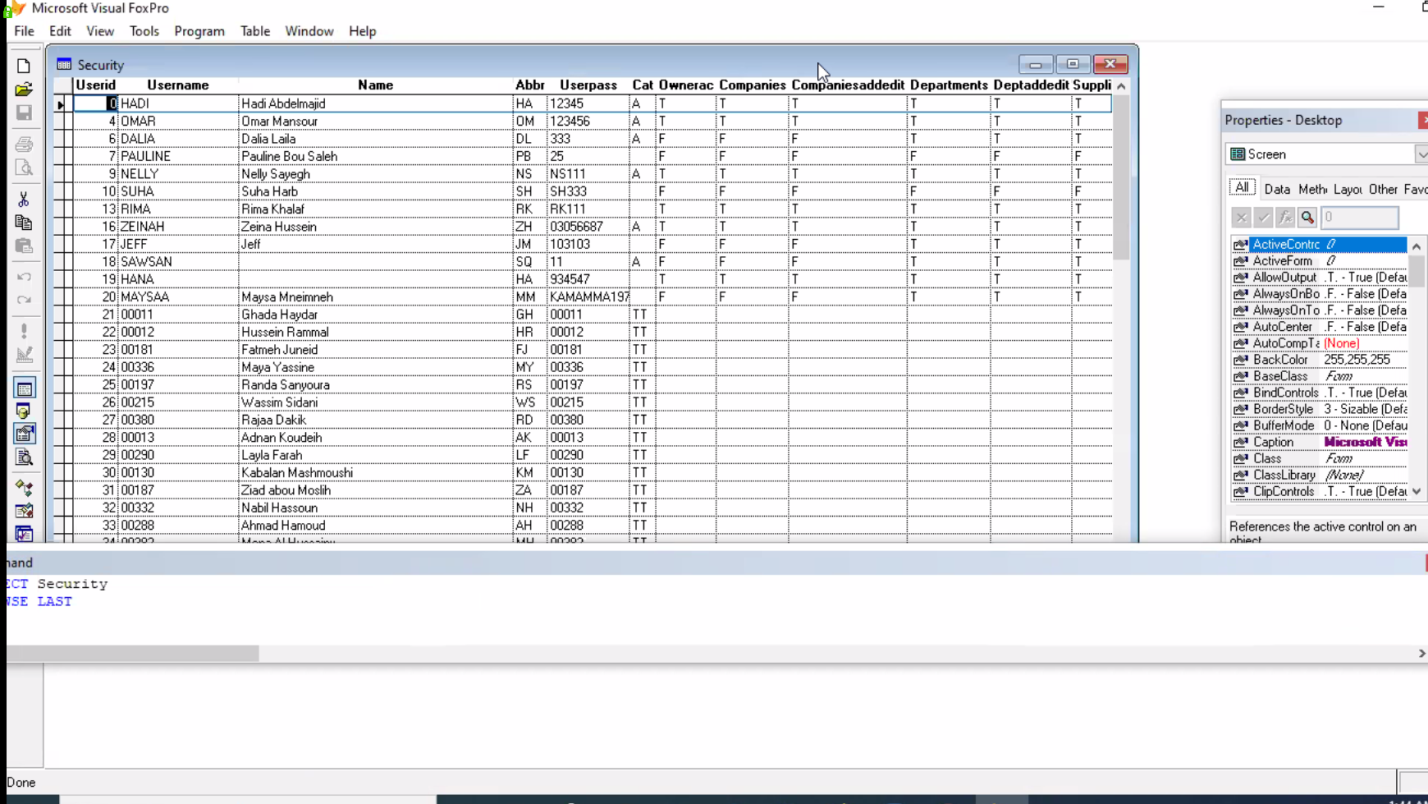
Back-end Version:



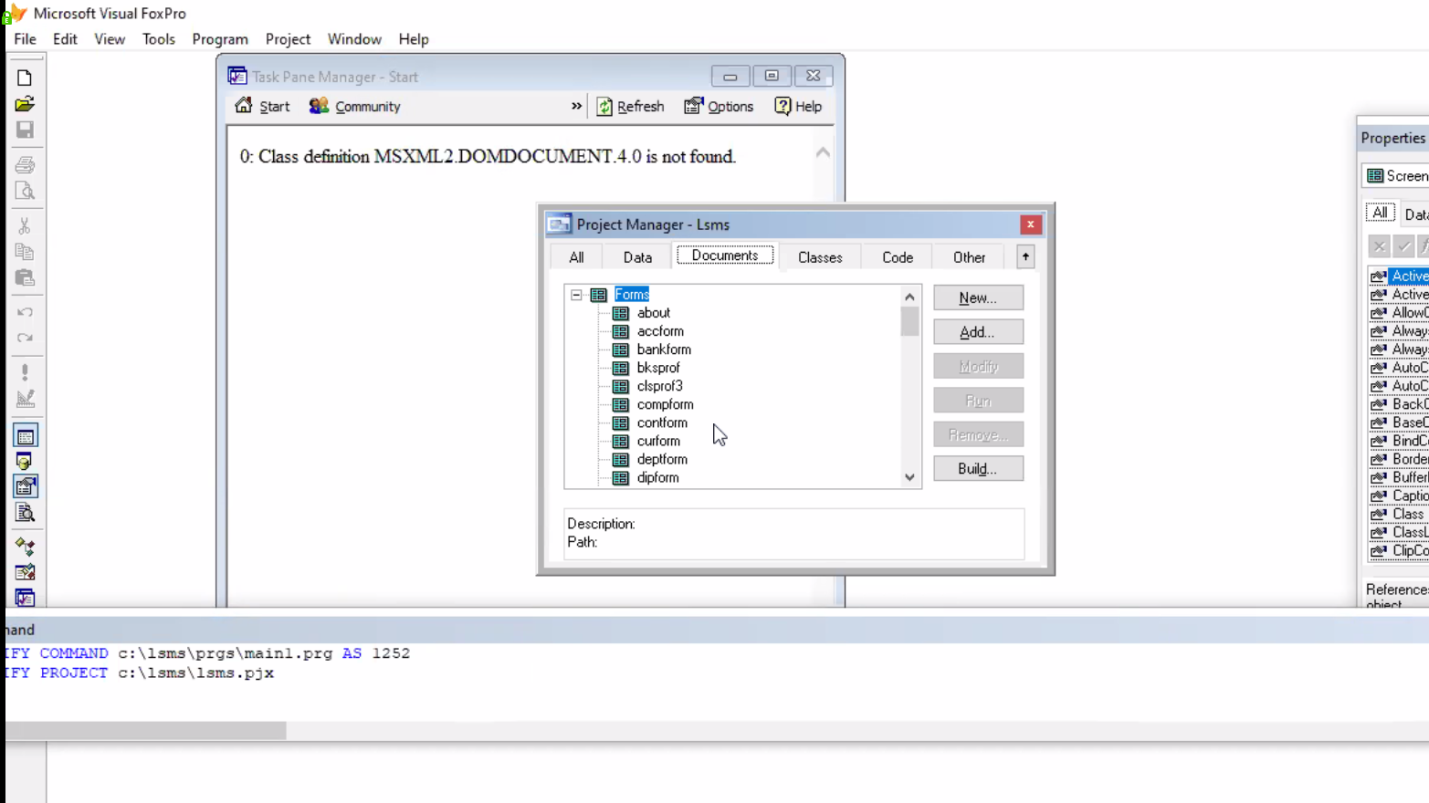
This is the project manager that enables us the founders, to manage our data,documents and classes.



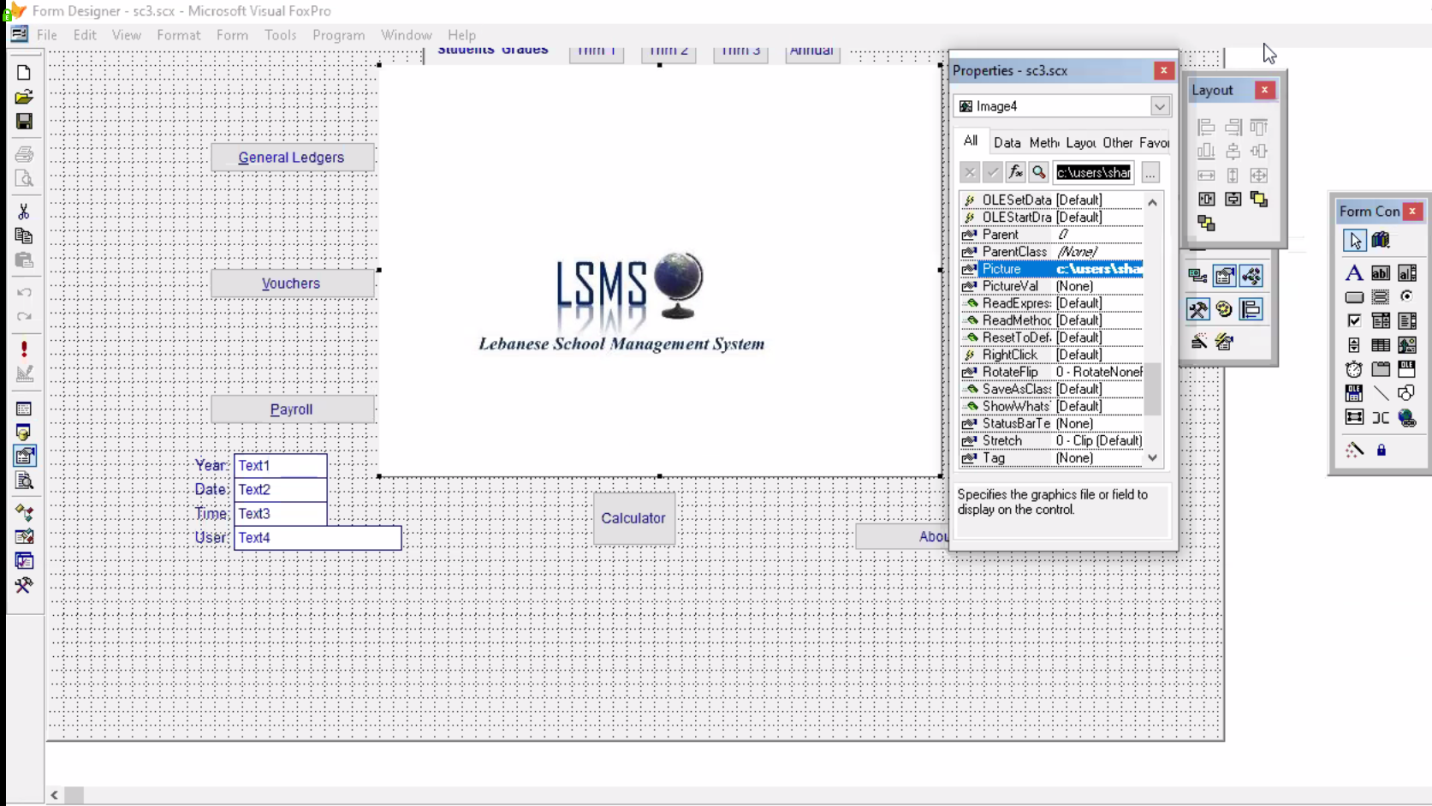
Data:In this section we can handle our database tables .



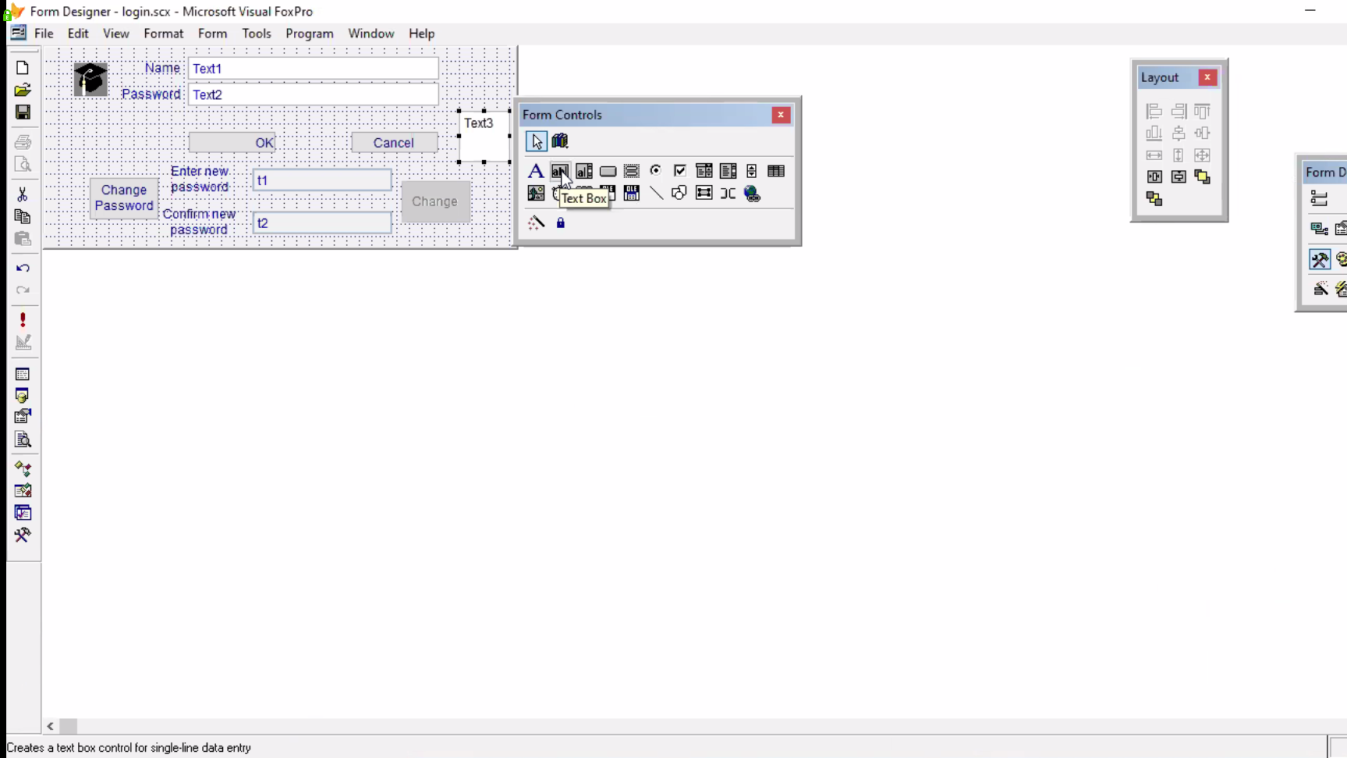
This is the security data base table , where we can settle and save all our usernames and password for the software users.



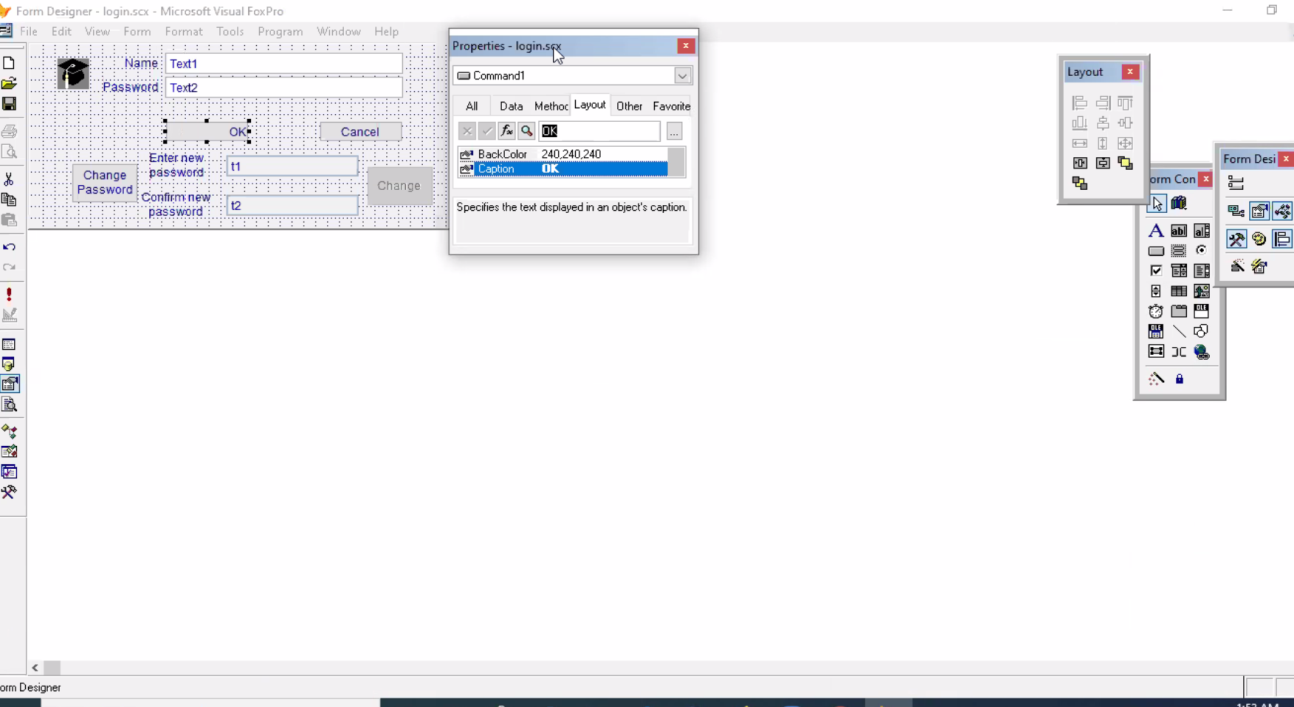
Documents: Here we can manage all our forms who are the software layers(Loading, login , main menu).



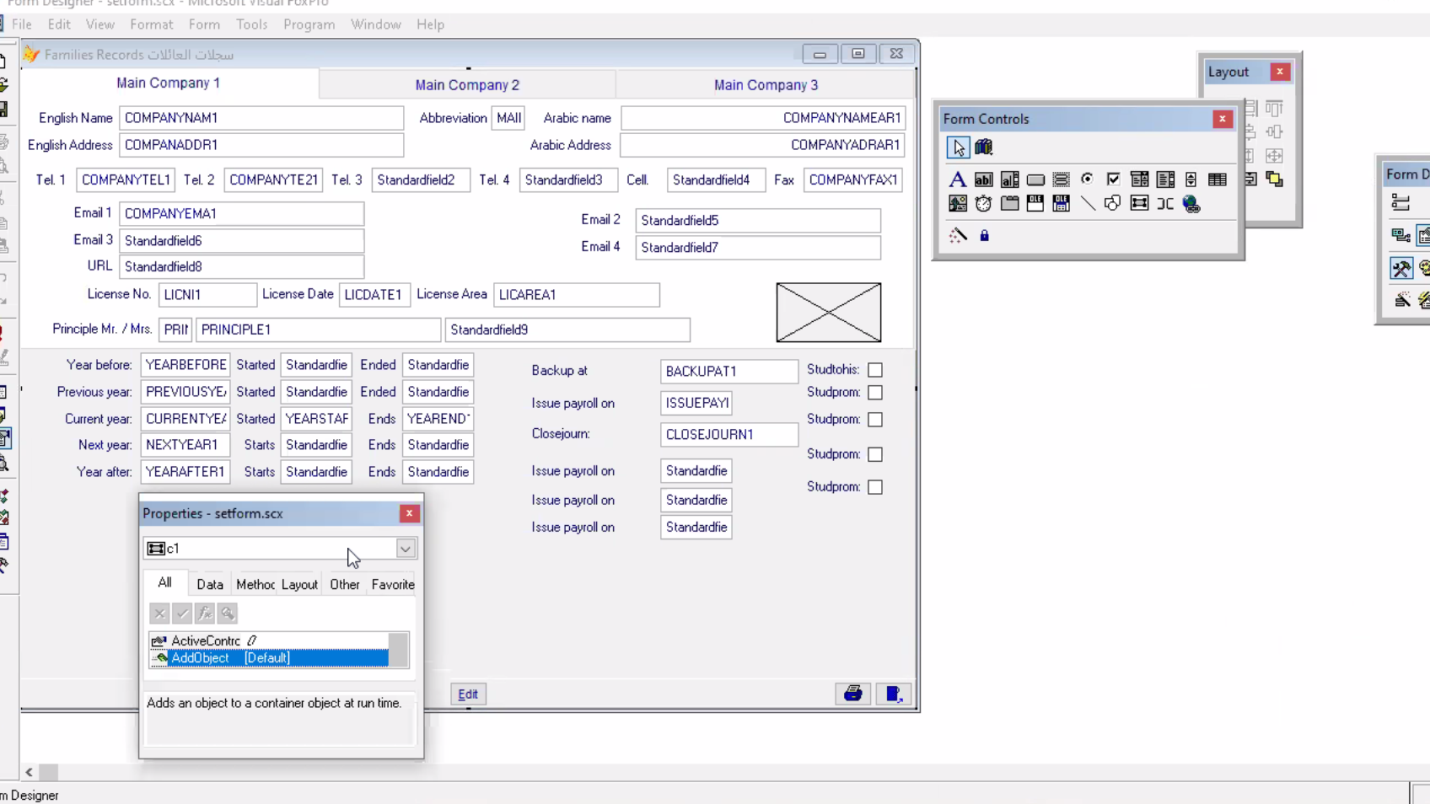
For example this is our main menu , and here we are modifying our logo through the properties tools window by browsing a new picture from the pc.



This is our login window , here we are showing you how we add new botton and boxes through the form controls.



Right here we are trying to change the button’s caption through “properties-log in”.



Here for example we are showing how to add a new object to one of our forms which is the family record form through “propreties-setform”