**جامعة حلب في المناطق المحررة**

**كلية الهندسة المعلوماتية**

**السنة الخامسة**

**أتمتة شؤون الطلاب والامتحانات**

**مشروع تخرج أُعدَّ لنيل درجة الإجازة في الهندسة المعلوماتية**

**إعداد الطلاّب**

**محمد سعد صوّان محمد ضبّيط**

**أحمد عبد العزيز محمد سليمان**

**إشراف**

**م. صهيب بكور**

**العام الدراسي**

**2020/2019**

**­­­­­­­­University of Aleppo in the Liberated Areas**

**Informatics Engineering Faculty**

**Fifth Year**

**AUTOMATION OF STUDENTS AND EXAMS AFFAIRS**

**Graduation Project prepared to obtain a Bachelor (Engineer) of**

**Informatics Engineering degree**

**By**

**Mohammad Dabbit Mohammed Saad Sawan**

**Mohammed Suleiman Ahmed Abdulaziz**

**Supervised by**

**Eng. Sohaib Bakour**

**Academic Year**

**2020/2019**

**الإهداء**

نقف بعجز وافتقار على بابك مولانا ونطرق باب محبتكَ راجين منك القبول، سبحانك عجز لسان القلب عن شكرك، سبحانك سبقت محبتك لعبيدك فأنارت الوجود، فرجاؤنا أن تقبل مِنّا ما هو منك إليك، وما أظهرتَه من بحر جودك وفضلك علينا.

إلى من بماء رعايتهم سقونا وبالعطف غمرونا وبالحنان حملونا وهجروا من أجلنا الرقاد، وأسعفوا ببلسم حنانهم جراحنا وفي مطلع كل يوم ندعو الله أن يحفظهم لنا ويديم عليهم الصحة والعافية.

آباؤنا وأمهاتنا

إلى الذين أناروا دربنا وضحوا بأنفسهم لنحيا بكرامة ونتمكن من تحصيل العلم بلا قيود ولا حواجز.

شهداءنا الأبرار

نهديهم هذا البحث المتواضع راجين من المولى عز وجل أن يجد القبول والنجاح وأن يحقق الهدف الذي وضع من أجله وهو مساعدة الجامعة التي من خلالها استطعنا أن نكمل مسيرتنا التعليمية بعد أن تقطّعت بنا السبل.

**الشكر**

الحمد لله رب العالمين والصلاة والسلام على أشرف الأنبياء والمرسلين سيدنا محمَّد وعلى آله وصحبه ومن تبعهم بإحسانٍ إلى يوم الدين، وبعد..

فإنّنا نشكر الله تعالى على فضله حيث أتاح لنا إنجاز هذا العمل وأعاننا على إكماله بفضله وكرمه، فله الحمد أولاً وآخراً.

وإنطلاقاً من حديث رسول الله عليه الصلاة والسلام "لا يَشْكُرُ اللَّهَ مَنْ لا يَشْكُرُ النَّاسَ" نتقدم بالشكر لكل من ساعد على إتمام هذا البحث وقدم لنا العون ومدّ لنا يد المساعدة وزودنا بالمعلومات اللازمة لإتمام هذا البحث وكانوا عوناً لنا في بحثنا هذا ونوراً يضيء الظلمات التي كانت تقف أحياناً في طريقنا، إلى من زرعوا التفاؤل في دربنا وقدموا لنا المساعدات والتسهيلات والأفكار، ربما دون أن يشعروا بدورهم بذلك فلهم منا كل الشكر، كما نشكر الظروف الصعبة التي مررنا بها والتي زادت من عزيمتنا وإصرارنا على المتابعة والتميز وتحقيق أحلامنا.

**الفهرس**

**الإهداء I**

[**الشكر II**](#_Toc50709595)

[**فهرس المحتويات**](#_Toc50709596) **III**

[**فهرس الأشكال XI**](#_Toc50709597)

[**ملخص 1**](#_Toc50709598)

[**مقدمة 2**](#_Toc50709599)

**الفصل الأول: الفكرة العامة للمشروع 4**

[**مقدمة 5**](#_Toc50709602)

[**حدود المشروع وبدايته 5**](#_Toc50709603)

[**أهداف المشروع 7**](#_Toc50709604)

[**نظام العمل في الجامعة 8**](#_Toc50709605)

[**حسنات النظام المقترح..................................................................................8**](#_Toc50709606)

[**النظام المطور 9**](#_Toc50709607)

[**النتيجة المستخلصة 9**](#_Toc50709608)

**الفصل الثاني: هندسة المتطلبات البرمجية.............................................................10**

[**الهندسة البرمجية للنظام 11**](#_Toc50709609)

[**متطلبات المشروع الوظيفية 11**](#_Toc50709610)

[**متطلبات المشروع غير الوظيفية 12**](#_Toc50709611)

[**المستفيدون من النظام 12**](#_Toc50709612)

[**النتيجة 13**](#_Toc50709613)

[**مخططات UML 13**](#_Toc50709614)

**مخطط حالة الاستخام Use Case Diagram ...................................................13**

**مخطط فئات النظام class diagram...............................................................15**

**المخطط التتابعي sequence diagram: .........................................................16**

**الفصل الثالث: قاعدة البيانات ........................................................................18**

[**مقدمة 20**](#_Toc50709616)

**مفهوم قاعدة البيانات 20**

[**مميزات لغة SQL 21**](#_Toc50709618)

[**الجداول التي تتكون منها قاعدة البيانات 22**](#_Toc50709619)

1. [**جدول الطالب ...........................................................................22**](#_Toc50709620)
2. [**جدول المادة ...........................................................................24**](#_Toc50709621)
3. [**جدول الدرجة...........................................................................24**](#_Toc50709622)
4. [**جدول الكليّة ...........................................................................25**](#_Toc50709623)
5. [**جدول الشهادة الثانوية 25**](#_Toc50709624)
6. **جدول تفاصيل الطالب 25**
7. [**جدول المدرس..........................................................................29**](#_Toc50709625)
8. [**جدول رسوم الطالب ........................................................29**](#_Toc50709626)
9. [**جدول المستخدم .........................................................30**](#_Toc50709627)

[**الربط بين الجداول 31**](#_Toc50709628)

[**فوائد استخدام علاقات الربط 31**](#_Toc50709629)

[**أنواع العلاقات...............................................................................32**](#_Toc50709630)

[**الإجرائيات المخزنة 34**](#_Toc50709631)

**1) إجرائية إضافة رسوم (add\_fees) 34**

**2) إجرائية إضافة مادة (add\_material) 34**

[**3) إجرائية إضافة طالب (add\_student) 34**](#_Toc50709632)

**4) إجرائية إضافة تفاصيل طالب (add\_student\_detile) 34**

**5) إجرائية إضافة مدرس (add\_teacher) 34**

**6) إجرائية عرض جميع الطلاب (all\_degree\_with\_student) 34**

**7) إجرائية عرض حالة الرسوم لجميع الطلاب في كل السنوات(all\_fees\_with\_student) 34**

**8) إجرائية جلب مواد إجدى الكليات (all\_mat) 34**

1. **إجرائية نسخة إحتياطية (Backup\_database) 34**

**10) إجرائية أخذ نسخة عن الدرجات (backup\_degree) 34**

**11) إجرائية التأكد من وجود تفاصيل للطالب (chek\_student\_detile) 34**

**12)** [**إجرائية كشف الدرجات (DEGREE\_SCORE) 34**](#_Toc50709632)

**13) إجرائية حذف الدرجة (delete\_degree) 34**

**14) إجرائية حذف رسوم طالب (delete\_fees) 34**

**15) إجرائية حذف مادة (delete\_material) 34**

**16)** [**إجرائية حذف بيانات طالب (delete\_student) 34**](#_Toc50709632)

**17) إجرائية حذف بيانات مدرس (delete\_teacher) 34**

**18) إجرائية تعديل الدرجة (edit\_degree) 34**

[**19) إجرائية تعديل درجة العملي (edit\_degree\_prac) 36**](#_Toc50709637)

[**20) إجرائية تعديل رسوم طالب (edit\_fess). 36**](#_Toc50709638)

[**21) إجرائية تعديل مادة (edit\_material) 36**](#_Toc50709639)

[**22) إجرائية تعديل الطالب (edit\_student). 36**](#_Toc50709640)

[**23)إجرائية تعديل بيانات مدرس (edit\_teacher) 36**](#_Toc50709641)

[**24) إجرائية السجل الامتحاني (exam\_record) 36**](#_Toc50709642)

[**25) إجرائية عدد الطلاب المتقدمين للامتحان (exam\_takers) 36**](#_Toc50709643)

[**26) إجرائية عرض المواد الراسبة (fail\_degree) 36**](#_Toc50709644)

[**27) إجرائية عدد الطلاب الراسبين (fail\_num) 36**](#_Toc50709645)

[**28) إجرائية جلب الدرجات (Get\_All\_Degree) 36**](#_Toc50709646)

[**29) إجرائية جلب الرسوم (get\_all\_fees) 36**](#_Toc50709647)

[**30) إجرائية جلب بيانات المواد (Get\_All\_Material) 36**](#_Toc50709648)

[**31) إجرائية عرض أسماء طلاب سنة ما (get\_all\_name\_in\_year) 37**](#_Toc50709649)

[**32) إجرائية جلب بيانات الطلاب (Get\_All\_student) 37**](#_Toc50709650)

[**33) إجرائية جلب بيانات الطلاب (Get\_All\_student\_inYear) 37**](#_Toc50709651)

[**34) إجرائية جلب بيانات المعلمين (Get\_all\_teacher) 37**](#_Toc50709652)

[**35) إجرائية جلب المعدل (Get\_Avg\_Degree) 37**](#_Toc50709653)

[**36) إجرائية جلب الدرجة في الفصل (get\_degreesWithSeson) 37**](#_Toc50709654)

[**37) إجرائية جلب درجات المواد لسنة ما (get\_mat\_year) 37**](#_Toc50709655)

[**38) إجرائية جلب درجات المواد الناجحة لسنة ما (get\_mat\_year2) 37**](#_Toc50709656)

[**39) إجرائية جلب درجات المواد الراسبة لسنة ما (get\_mat\_year3) 37**](#_Toc50709657)

[**40) إجرائية جلب جميع المواد (Get\_Material) 37**](#_Toc50709658)

[**41) إجرائية جلب بيانات رسوم طالب (get\_std\_fees) 37**](#_Toc50709659)

[**42) إجرائية جلب اسم طالب (Get\_stdname) 37**](#_Toc50709660)

[**43) إجرائية جلب الرقم الجامعي لطالب (Get\_stdnum) 38**](#_Toc50709661)

[**44) إجرائية جلب بيانات مدرس (Get\_stdnum) 38**](#_Toc50709662)

[**45) إجرائية جلب العشرة الأوائل (get\_top10) 38**](#_Toc50709663)

[**46) إجرائية إدخال الدرجة (isert\_degree) 38**](#_Toc50709664)

[**47) إجرائية عدد المواد الراسبة (not\_success\_year) 38**](#_Toc50709663)

[**48) إجرائية جلب المواد غير المقدمة (null\_matirial) 38**](#_Toc50709664)

[**49) إجرائية جلب بيانات المادة الراسبة (replace\_fail\_degree) 38**](#_Toc50709663)

**50) إجرائية استعادة نسخة احتياطية (restore\_db) 38**

**51) إجرائية البحث عن الطالب بالرقم (Search\_by3id) 38**

**52) إجرائية البحث عن الدرجة (Search\_Degree) 38**

**53) إجرائية البحث عن مادة عبر الرقم (search\_material\_by\_id) 38**

**54) إجرائية البحث مادة عبر الاسم (search\_material\_by\_name) 38**

[**55) إجرائية البحث عن الطالب بالاسم (Search\_Student\_name) 39**](#_Toc50709666)

**56) إجرائية البحث عن معلم عبر الرقم (search\_teacher\_by\_id) 39**

[**57) إجرائية البحث عن معلم عبر الاسم (search\_teacher\_by\_name) 39**](#_Toc50709667)**7**

[**58) إجرائية تسجيل الدخول (Sp\_Login) 39**](#_Toc50709668)

[**59) إجرائية تفاصيل الطالب (Student\_detiles) 39**](#_Toc50709669)

[**60) إجرائية عدد الطلاب الناجحين (success\_num) 39**](#_Toc50709670)

**61) إجرائية إيجاد الطلاب الناجحين (success\_year) 39**

**62) إجرائية ترفيع الطلاب (success\_year\_ok) 39**

**القيود ..............................................................................................38**

**قيود جدول المواد................................................................................38**

**قيود جدول الدرجات.............................................................................39**

**قيود جدول الطلاب..............................................................................39**

**الفصل الرابع: تصميم واجهة البرنامج.................................................................40**

**مقدمة.............................................................................................41**

**أهمية لغة.C# ....................................................................................41**

**خصائص لغة.C# .................................................................................42**

**تقسيم البرنامج.....................................................................................43**

**الواجهة الأولى "واجهة تسجيل الدخول".............................................................44 الواجهة الثانية "الواجهة الرئيسية"..................................................................45 User control.....................................................................................47**

**أولا: إدارة الطلاب..................................................................................48**

**تسجيل طالب...................................................................................48**

**قائمة الطلاب..................................................................................54**

**ترفيع الطلاب..................................................................................55**

**إدارة الرسوم....................................................................................58**

**ثانياً: إدخال الدرجات...............................................................................63**

**طالب واحد.....................................................................................64**

**عدة طلاب.....................................................................................66**

**ثالثاً: إدارة الدرجات................................................................................67**

**كشف العلامات.................................................................................68**

**صحيفة الطالب.................................................................................70**

**إصدار مادة....................................................................................72**

**السجل الامتحاني...............................................................................74**

**العشرة الأوائل..................................................................................75**

**رابعاً: إدارة المواد والمدرسين.......................................................................76**

**إدارة المواد.....................................................................................76**

**إدارة المدرسين.................................................................................78**

**خامساً: إنشاء نسخة احتياطية.....................................................................80**

**سادساً: استعادة نسخة احتياطية...................................................................82**

**القسم البرمجي........................................................................................84**

**البرمجة الكائنية.......................................................................................84**

**مجلدات البرنامج......................................................................................84**

**صف (DataAccessLayer).......................................................................86**

**صف تسجيل الدخول (CLs\_Login).................................................................87**

**صف الطالب (CLs\_student)......................................................................88**

**صف الدرجة (insert\_degree).....................................................................90**

**صف المدرسين (CLs\_TeacherandMaterial)...................................................93**

**صفوف PL و UserControl.......................................................................96**

**صف واجهة تسجيل الدخول(frmlogin)..............................................................96**

**صف الواجهة الرئيسية (FRM\_MAIN)...............................................................97**

**صف الـUser Control الخاص بإدارة الطلاب (US\_Student\_Managament)..................99**

**صف الـUser Control الخاص بإدخال البيانات (US\_Degree\_Insert).........................105**

**صف الـUser Control الخاص بإدارة الدرجات (US\_Degree\_Management)................110**

**صف الـUser Control الخاص بالنسخ الاحتياطي(US\_Backup)...............................117**

**صف الـUser Control الخاص باستعادة النسخة الاحتياطية (US\_Restore)....................119**

**الخلاصة............................................................................................120**

**المقترحات و التوصيات..............................................................................121**

**المراجع.............................................................................................122**

**الملاحق............................................................................................123**

**الاجرائيات المخزنة..................................................................................123**

**فهرس الأشكال**

الشكل (1-1) جدول مهام المشروع...............................................................6

الشكل (1-2) مخطط غانت............................................................................7

الشكل (2-1) مخطط حالات الاستخدامللفاعل "المدير"................................................14

الشكل (2-2) مخطط حالات الاستخدام للفاعل "موظف شؤون الطلاب"................................14

الشكل (2-3) مخطط حالات الاستخدام للفاعل "موظف شؤون الامتحانات"............................15

الشكل (2-4) مخطط فئات النظام Class diagram................................................16

الشكل (2-5) المخطط التتابعي sequence digram..............................................17

الشكل (3-1) تصميم جدول الطلاب داخل قاعدة البيانات.............................................22

الشكل (3-2) تصميم جدول المواد داخل قاعدة البيانات..............................................23

الشكل (3-3) تصميم جدول الدرجات داخل قاعدة البيانات............................................24

الشكل (3-4) تصميم جدول الكليات داخل قاعدة البيانات.............................................24

الشكل (3-5) تصميم جدول الشهادة الثانوية داخل قاعدة البيانات....................................25

الشكل (3-6) تصميم جدول تفاصيل الطالب داخل قاعدة البيانات.....................................27

الشكل (3-7) تصميم جدول المدرس داخل قاعدة البيانات............................................28

الشكل (3-8) تصميم جدول رسوم الطالب داخل قاعة البيانات........................................29

الشكل (3-9) تصميم جدول المستخدم داخل قاعدة البيانات...........................................30

الشكل (3-10) مخطط العلاقات بين جداول قاعدة البيانات...........................................32

الشكل (4-1) تصميم واجهة تسجيل الدخول داخل بيئة الفيجول استديو..............................45

الشكل (4-2) تصميم الواجهة الرئيسية داخل بيئة الفيجول استديو...................................46

الشكل (4-3) تصميم الـUser Control داخل بيئة الفيجول استديو................................48

الشكل (4-4) نافذة تسجيل الطالب ..................................................................49

الشكل (4-5) تعبئة البيانات آليا لأجل التعديل........................................................52

الشكل (4-6) ظهور رسالة لتأكيد عملية الحذف......................................................53

الشكل (4-7) قائمة بيانات كافة الطلاب..............................................................54

الشكل (4-8) الرسالة التنبيهية الأولى لترفيع الطلاب.................................................55

الشكل (4-9) واجهة ترفيع الطلاب...................................................................56

الشكل (4-10) المعاينة الأولية للطلاب الناجحين و الراسبين ........................................57

الشكل (4-11) الرسالة التأكيدية النهائية لترفيع الطلاب..............................................58

الشكل (4-12) نافذة إدارة الرسوم....................................................................59

الشكل (4-13) واجهة إضافة رسوم..................................................................60

الشكل (4-14) واجهة قائمة الرسوم من أجل التعديل.................................................61

الشكل (4-15) ظهور رسالة تأكيد لحذف رسوم طالب................................................62

الشكل (4-16) تصميم الـUser Control الخاص بإدخال الدرجات لطالب واحد.....................63

الشكل (4-17) البحث عن درجات طالب عن طريق رقمه الجامعي....................................65

الشكل (4-18) تصميم الـUser Conrol الخاص بإدخال الدرجات لعدة طلاب.......................66

الشكل (4-19) إدارة الدرجات لطالب واحد (كشف العلامات)..........................................68

الشكل (4-20) إدارة الدرجات لطالب واحد (صحيفة الطالب)..........................................70

الشكل (4-21) واجهة إصدار مادة...................................................................72

الشكل (4-22) إصدار علامات مادة معينة...........................................................73

الشكل (4-23) نافذة السجل الامتحاني السنوي......................................................74

الشكل (4-24) نافذة العشرة الأوائل..................................................................75

الشكل (4-25) نافذة إدارة المواد.....................................................................76

الشكل (4-26) نافذة قائمة المواد....................................................................77

الشكل (4-27) نافذة إدارة المدرسين.................................................................78

الشكل (4-28) نافذة إدارة المدرسين(تعديل بيانات مدرس)............................................79

الشكل (4-29) تصميمم الـUser Control الخاص بإنشاء نسخة احتياطية........................80

الشكل (4-30) تصميمم الـUser Control الخاص باستعادة نسخة احتياطية......................83

الشكل (4-31) المجلدات ضمن مستعرض مكونات المشروع..........................................86

الشكل (4-32) ظهور رسالة تدل على نجاح عملية تسجيل الدخول...................................96

الشكل (4-33) ظهور رسالة تدل على فشل عملية تسجيل الدخول...................................97

**ملخص**

مشروع أتمتة شؤون الطلاب والامتحانات هو مشروع خدمي لمساعدة موظفي شؤون الطلاب والامتحانات في جامعة حلب في المناطق المحررة، ويتمثل ببناء وتطوير برنامج مخصص للجامعة، ذو واجهة سهلة الاستخدام ومرنة، حيث يقوم البرنامج بحل مشكلة تخزين البيانات بعدة ملفات وذلك بتجميع البيانات داخل هيكل واحد منظم، وتسهيل وتسريع أعمال موظفي الشؤون، تم إنجاز هذه الأمور على ثلاث مراحل، المرحلة الأولى تتمثل بتعريف النظام المستخدم وتحديد مشاكله وعيوبه بالإضافة إلى تحليل النظام الجديد وتحديد المتطلبات باستخدام هندسة البرمجيات، والمرحلة الثانية هي بناء قاعدة البيانات والجداول والربط بينها باستخدام لغة الاستعلامات SQL، والمرحلة الثالثة هي تصميم واجهة سلسة ومرنة وبرمجتها باستخدام لغة البرمجة C#.

**مقدمة**

مع التقدم العلمي خلال القرن الواحد والعشرين والتطور المعلوماتي الذي نعيشه أو بالمعنى الأصح الذي نتعايش معه، وبعد الثورة المباركة في بلاد الشام التي تسببت بدخول التكنولوجيا بشكل كبير وسريع إلى مناطقنا، مما أحدث فجوة كبيرة بين الأساليب المعتمدة في التكنولوجيا القديمة والأساليب المعتمدة في التكنولوجيا الحديثة، وصعوبة تكيّف الموظفين والمستخدمين مع هذا التطور، والتكلفة المرتفعة والغلاء الذي يجعل المستخدمين يتجنبون تحديث وتطوير أساليب عملهم.

إن من المشاكل التي التمسناها في الجامعة عموماً وفي كليات الهندسة والعلوم خصوصاً هو اعتماد موظفيها على الأساليب التقليدية في القيام بأعمالهم عن طريق استخدام برامج مكتبية جرت العادة في استخدامها بسبب سهولة تعلمها من جهة ولأنها تعتبر مجانية من جهة أخرى، ولكن المشكلة تتجسد في أن هذه البرامج تم اعتمادها في بداية تأسيس الجامعة وكان من الممكن الاعتماد عليها وذلك بسبب قلّة أعداد الطلاب ولكن بعد مرور عدة سنوات وازدياد أعداد الطلاب في كل عام جعل من الصعب البقاء على هذا الحال والاعتماد على الروتين القديم في قضاء الأعمال.

من الصعب التعامل مع كل المشاكل وحلها جميعها في وقت واحد، لذلك قمنا بالبحث عن أكثر المشاكل التي تحتاج إلى حل، ومن أهم هذه المشاكل وعدم وجود نسخ إلكترونية لبيانات الطلاب وتأخر إصدار كشف الدرجات وكذلك تأخر إصدار درجات العملي والنظري للمواد، ويعود السبب في ذلك إلى ضغط العمل الكبير الواقع على موظفي شؤون الطلاب الامتحانات، وعدم وجود قاعدة بيانات مركزية تسهل العمل عليهم.

وبعد الاطلاع على هذه المشاكل بدأنا بوضع خطة لحل هذه المشكلة، وتتمثل هذه الخطة بإنشاء مشروع خدمي مجاني يتمثل بتطوير برنامج لأتمتة شؤون الطلاب والامتحانات.

تتجلى أهمية المشروع وأهدافه في القيام بخدمة رائعة للجامعة وطلابها، وذلك بتسهيل عمل موظفين شؤون الطلاب والامتحانات وتخفيف العبء عليهم من جهة، واختصار وقت الطلاب عند مراجعتهم لمكاتب الشؤون من جهة أخرى، مما يحقق الفائدة لكلا الطرفين.

**مراحل العمل على المشروع**

من أجل سلاسة العمل وتنظيمه قمنا بتقسيم المشروع على أربعة فصول كالتالي:

* **الفصل الأول:** يتحدث عن الفكرة العامة للمشروع أي وحدوده وبدايته وأهدافه، بالإضافة إلى المقارنة بين النظام المستخدم والنظام المقترح من حيث الميزات والإيجابيات.
* **الفصل الثاني:** مخصص لتحليل النظام الجديد وتحديد المتطلبات وطريقة تنظيم العمل في بنية هرمية باستخدام هندسة البرمجيات لتسهيل القيام بالمشروع.
* **الفصل الثالث:** يتضمن شرح عن مفهوم قاعدة البيانات ونوع اللغة المستخدمة ومزاياها، ويحتوي شرح مفصل عن الجداول التي تم إنشائها، وتوضيح عملية الربط بين هذه الجداول، والإجرائيات التي تم إنشائها وتخزينها ضمن قاعدة البيانات، والقيود التي تم وضعها على حقول الجداول.
* **الفصل الرابع:** يحتوي تفصيل عن اللغة المستخدمة في تصميم وتطوير البرنامج، وشرح مفصل عن الصفوف التي تم إنشائها والإجرائيات التي تم بنائها والواجهات التي تم تصميمها.

**الفصل الأول**

**الفكرة العامة للمشروع**

**الفصل الأول**

**الفكرة العامة للمشروع**

**مقدمة**

يعرض هذا الفصل أسباب المشروع وفكرته العامة ويبين حدوده وأهدافه ويفصّل خطوات تنفيذ المشروع في بنية هرمية تنظّم سيره.

من خلال البحث والاطلاع على المشاكل التي تواجه الإداريين في عملهم في كليات الهندسة والعلوم في جامعة حلب في المناطق المحررة، وجدنا أنه لدى موظفي الشؤون مشكلة كبيرة وهي تخزين البيانات بعدة ملفات وذلك لعدم وجود برنامج مخصص يساعدهم على التخزين ضمن قاعدة بيانات مركزية، حيث يجدون صعوبة في تسيير الأعمال ويبذلون جهد كبير لإتمام عملهم، فقررنا مساعدتهم في أداء أعمالهم عن طريق أتمتة العمل بشكل شبه كامل وذلك بتصميم برنامج بواجهة سهلة ومرنة، مما يسهل ويسرّع إنجاز عملهم، وهذه الواجهة مرتبطة بقاعدة بيانات ذات تنظيم احترافي تضم جميع البيانات في هيكل واحد.

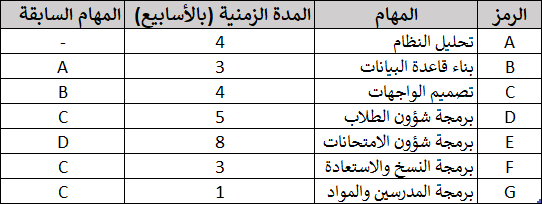
**حدود المشروع وبدايته**

إن هذا المشروع هو مشروع خدمي يستهدف جامعة حلب في المناطق المحررة بغية مساعدة الموظفين وتسهيل عملهم، حيث تطلب المشروع مدة لا تقل عن سبعة أشهر.

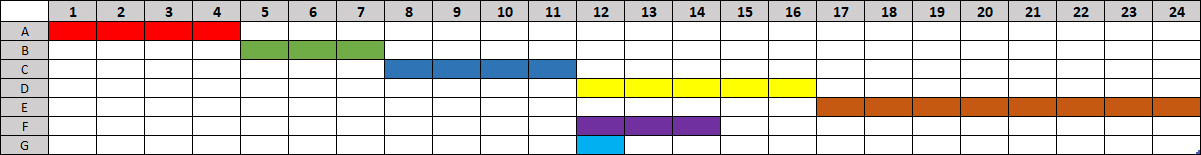
مع بداية الشهر الثاني عشر قمنا بزيارة إلى موظفي الشؤون، حيث تحدثنا عن المشاكل التي تعترض وتبطئ سير عملهم، ففي اللقاء الأول طُرح علينا مشكلة عدم وجود آلية واحدة تنظّم ملفات وبيانات الطلاب، حيث كانت هذه المشكلة هي سبب بدأنا بالمشروع الذي يقوم على حلها، والحل الذي دار في خلدنا هو بناء وتصميم برنامج يسهل قيام الموظفين بأعمالهم، ويوحّد البيانات في كيان واحد منظم، وفي اللقاء الثاني ومن خلال طرح عدة أسئلة على الموظفين أصبح لدينا تصور أولي عن شكل قاعدة البيانات التي يتطلبها المشروع، ومن أجل التعمق في تفاصيل المشروع و الحصول على المعلومات اللازمة من أجل بناء المشروع قمنا بالذهاب إلى شعبة الامتحانات وشؤون الطلاب الخاصة بكليات الهندسة والعلوم والتقينا بالموظفين، ومن خلال هذا اللقاء تم عرض العديد من المشاكل التي تواجههم وقمنا باختيار المشاكل ذات الأهمية والأولية الأكبر من أجل حلها، حيث طلب الموظف حسب تصوره برنامج يقوم بما يلي:

* جمع البيانات في ملف واحد من أجل سهولة الوصول إليها.
* برنامج سهل ومرن يمكن من خلاله إدخال وجلب البيانات.
* يمكن من خلال البرنامج إضافة بيانات الطلاب المستجدين والتعديل على بيانات الطلاب القدامى، وإمكانية حذف بيانات أي طالب.
* يمكن من خلال البرنامج إدخال علامات الطلاب العملية والنظرية بطريقة سهلة.
* يمكن من خلال ضغطة زر إصدار كشف علامات لأي طالب.
* حفظ البيانات في ملف واحد كنسخة احتياطية.
* إمكانية استعادة البيانات المحفوظة إذا حدثت أي مشكلة في الحاسب.

وبعد هذه الطلبات أصبح لدينا تصور كامل عن البرنامج الذي سنقوم بإنشاءه وتصميمه، حيث كانت الخطوة الأولى هي تنظيم عملنا باستخدام مخطط Gantt وهو عبارة عن أداة هامة جدًا وشائعة الاستخدام في إدارة المشاريع واستعراض الأنشطة مقرونة مع الزمن، مخطط غانت ويمثل وسيلة فعالة في فرض الرقابة على المشاريع وإدارتها، والمساعدة في التخطيط والجدولة لكافة المشاريع بغض النظر عن حجمها، ويؤدي المخطط دوراً مميزا ًفي تبسيط المشاريع المعقدة والتعامل معها بكل بساطة.

ويبين الشكل التالي مهام المشروع:

**الشكل (1-1) جدول مهام المشروع.**

والشكل التالي يوضح مخطط Gantt للمشروع:

**الشكل (1-2) مخطط Gantt للمشروع.**

يستكشف الجزء الأيسر من المخطط المهام الواجب تنفيذها، أما الجزء العلوي فيوضح مقياساً زمنياً ملائماً.

**أهداف المشروع**

إن الهدف من هذا المشروع في المقام الأول هو العمل من أجل إفادة الجامعة وحل بعض مشاكلها، وتتجلى بـ:

* حل مشكلة تخزين البيانات بعدة ملفات، وذلك بدمج البيانات ضمن هيكل واحد منظم.
* حل مشكلة تأخر إصدار كشف العلامات مما يؤدي إلى توفير وقت وجهد الموظفين داخل الجامعة والطلاب أيضاً.
* الانتقال من العمل اليدوي المُجهد إلى أسلوب تقني حديث يستفيد من التطور الحاصل في التقنيات المعلوماتية الحديثة.
* الحصول على وثائق واستعلامات سريعة ودقيقة ومنظمة.
* تقليل نسبة الخطأ وبالتالي رفع جودة الخدمات المقدمة.
* تأمين نسخ إلكترونية عن البيانات ودرجات الطلاب وحمايتها من الضياع.

**نظام العمل في الجامعة**

يعتمد نظام العمل ضمن شؤون الطلاب على تسجيل بيانات الطلاب ورقياً وتخزينها ضمن مصنفات مرتبة تستهلك الكثير من الوقت عند الحاجة إلى أيّ منها.

أما في شؤون الامتحانات فيحصل الموظف على بيانات بأسماء الطلاب وأرقامهم الجامعية من موظف شؤون الطلاب ويتم تنظيمها ضمن ملفات مجدولة، يتم إدخال درجات الطلاب على مرحلتين وهما مرحلة إدخال درجات العملي ثم إصدارها على شكل جداول، ومرحلة إدخال درجات النظري ثم إصدارها على شكل جداول أيضاً، يتم إصدار وثائق كشف الدرجات والحياة الجامعية والسجل الامتحاني السنوي عبر جمع البيانات من الجداول المتفرقة وهنا من أهم الصعوبات مشكلة التأخر في إصدار الكشف بسبب تبعثر البيانات وتشتتها، ومن المشاكل أيضاً البحث عن الدرجة ضمن أكثر من ملف بسبب رسوبها في الدورة الأولى، يتم أيضاً في نهاية العام جمع المواد الراسبة لكل طالب ونقله إلى السنة التالية بحسب قرار الترفع الصادر عن رئاسة الجامعة والذي يحدد عدد المواد المسموح للطالب الرسوب بها.

**حسنات النظام المقترح**

* برنامج سهل ومرن يمكن من خلاله إدخال وجلب البيانات.
* عرض قائمة ببيانات جميع الطلاب وإمكانية تصديرها كملف إكسل.
* عرض قائمة ببيانات جميع الدرجات وإمكانية تصديرها كملف إكسل.
* جمع البيانات في ملف واحد من أجل سهولة الوصول إليها.
* يمكن من خلال البرنامج إضافة بيانات الطلاب المستجدين والتعديل على بيانات الطلاب القدامى، وإمكانية حذف بيانات أي طالب (تشمل بيانات الطالب الشخصية والجامعية والرسوم).
* يمكن من خلال البرنامج إدخال علامات الطلاب العملية والنظرية بطريقة سهلة (طالب واحد أو عدة طلاب) عن طريق جدول أو من خلال استيراد علامات جاهزة من ملف إكسل.
* يمكن من خلال ضغطة زر إصدار كشف علامات لأي طالب.
* يمكن من خلال ضغطة زر إصدار صحيفة الطالب.
* يمكن من خلال ضغطة زر إصدار السجل الامتحاني السنوي للسنة الدراسية المحددة وتصديرها كملف PDF.
* إصدار نتيجة مادة معينة وتصديرها كملف PDF.
* إظهار ترتيب العشرة الأوائل عن كل دفعة.
* حفظ البيانات في ملف واحد كنسخة احتياطية.
* إمكانية استعادة البيانات المحفوظة عند حدوث أي مشكلة.
* يعمل البرنامج عن طريق الشبكة المحلية (LAN) وبالتالي يمكن أن يعمل أكثر من حاسب على نفس قاعدة البيانات في وقت واحد.

**النظام المقترح**

نظام جديد سيتخلص قدر الإمكان من المساوئ الموجودة في النظام القديم، وسيحقق النتائج المرغوبة بأقل كلفة مادية وزمنية بالإضافة إلى زيادة جودة العمل وفعاليته.

**النتيجة المستخلصة**

نظام مؤتمت يستفيد من التقنيات المعلوماتية الحديثة ويوفر الوقت والجهد والمال وهو قابل للتطوير مستقبلاً عند تغير المتطلبات أو عند ظهور تقنيات جديدة ترغب إدارة الجامعة بالاستفادة منها.

**الفصل الثاني**

**هندسة المتطلبات البرمجية**

**الفصل الثاني**

**الهندسة البرمجية للنظام**

إنّ هندسة البرمجيات هي علم يهتم بتطوير البرمجيات بجودة عالية مع مراعاة احتياجات المستخدم ومتطلباته على جميع المستويات.

يهتم علم هندسة البرمجيات بتحليل البرمجيات وتصميمها وتطويرها، ويعمل على التحقق منها وإدارتها من أجل القيام بهندسة البرمجية بشكل صحيح يجب اتباع منهجيات معينة في هندسة البرمجيات وتحليل وتصميم نظم المعلومات. انطلاقا من ذلك لابد من اتباع منهجية محددة تتناسب مع مشروعنا المطروح وتتوافق مع أهدافه وغاياته.

**متطلبات المشروع الوظيفية**

يقصد بها وصف وظائف النظام التي تتعلق بنوعية البرامج والمستخدمين المتوقعين وطبيعة العمل الذي سيستخدم فيه النظام.

بعد دراسة عمل الشؤون تم تحديد وظائف النظام المطلوب بـ:

**إدارة شؤون الطلاب:**

1. الوصول لبيانات الطلاب الشخصية والجامعية وإمكانية الاضافة والتعديل والحذف عليها.
2. الوصول لبيانات الرسوم المدفوع والمتبقية على كل طالب واضافة أو التعديل والحذف عليها.
3. يجب وضع قائمة للطلاب للوصول إلى بيانات أي طالب بسهولة.

**إدارة شؤون الامتحانات:**

1. اصدار علامات المواد الخاصة بكل سنة دراسية.
2. اصدار وثيقة كشف علامات.
3. اصدار وثيقة صحيفة الطالب.
4. وضع السجل الامتحاني.
5. نقل الطلاب إلى السنة الأعلى في نهاية العام الدراسي.
6. وضع قائمة الأوائل على الدفعات.

**متطلبات المشروع غير الوظيفية**

هي المتطلبات التي تصف خصائص المشروع وقيوداً إضافية عليه، تعتبر هذه المتطلبات متطلبات هامة لجودة النظام من حيث درجة الأمان، ومتطلبات الأداء والقيود الزمنية ومعايير وقيود عملية التطوير، أهم المتطلبات غير الوظيفية:

1. تأمين الوثوقية للبيانات من خلال حفظها وأرشفتها والنسخ الاحتياطي لها.
2. حماية النظام من الاختراق عن طريق وضع الصلاحيات للوصول اليه.
3. توفير الوقت والجهد اللازم للبحث وجمع البيانات المطلوبة.
4. سرعة الاستجابة عند تنفيذ أي عملية.
5. تمييز المدخلات بحيث لا يكون تداخل او ضياع او تكرار للبيانات المدخلة الى قاعدة البيانات.
6. سهولة استخدام المشروع ووضوحه عبر الملاحظات الموجودة في كافة واجهات البرنامج.
7. تنظيم المشروع عبر تقسيمه على نوافذ محددة وواضحة.
8. إمكانية العمل على قاعدة البيانات عبر أكثر من جهاز.

**المستفيدون من النظام**

هم الجهات التي بإمكانها الاستفادة من النظام المطروح بشكل مباشر أو غير مباشر. يقصد بغير المباشر الحاجة إلى إحداث بعض التغيير في النظام من أجل تلبية متطلبات جديدة لخدمة كليات أخرى.

المستفيدون من النظام:

* موظفو شؤون الطلاب والامتحانات.
* الطلاب.

**النتيجة**

نتيجة هذا المشروع هي تطبيق يعمل على الحاسب، يقوم بتنفيذ المهام المطلوبة بدقة وسرعة عالية، ويمتاز بالمرونة وقابلية التطوير.

وبما أنه تم تقسيم وتوضيح خطوات العمل، سنبدأ بالخطوة الأولى الفعلية وهي البدء ببناء قاعدة البيانات، وهذا ما سيتم تفصيله في الفصل التالي.

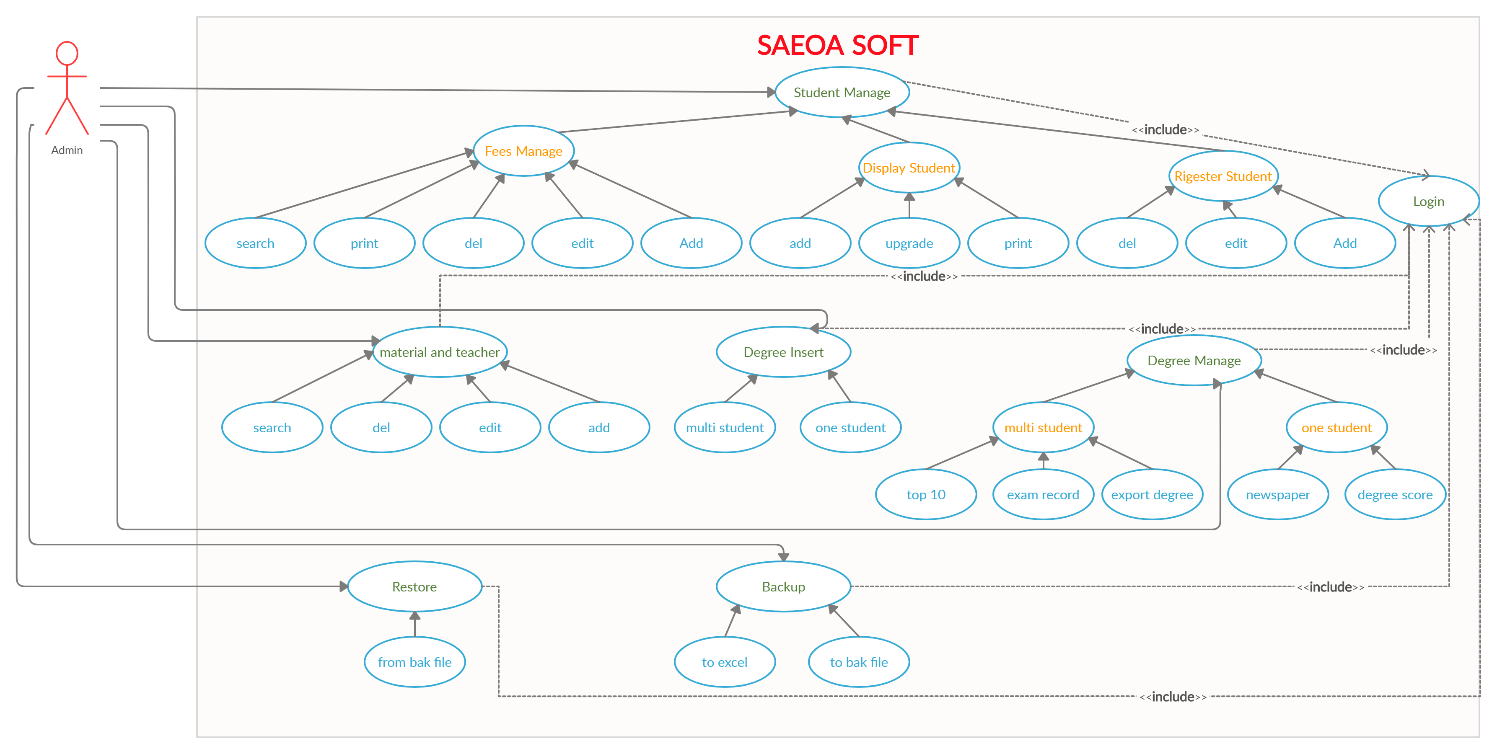
**مخططات UML**

لغة النمذجة الموحدة UML هي اختصار ل Unified Modeling Language وهي لغة قياسية تستخدم لبناء تصور عن الأعمال المختلفة لبرمجيات الأنظمة، فضلاً وهذه اللغة هي جزء هام لتطوير البرمجة الكائنية التوجه "object oriented programming" وتطوير البرمجيات "software development"، وتستخدم هذه اللغة الرموز الرسومية في الغالب للتعبير عن طريقة تصميم مشاريع البرامج، تساعد لغة UML فريق العمل في المشروع على التواصل، استكشاف إمكانات التصاميم، والتحقق من صحة التصميم الهندسي للبرنامج.

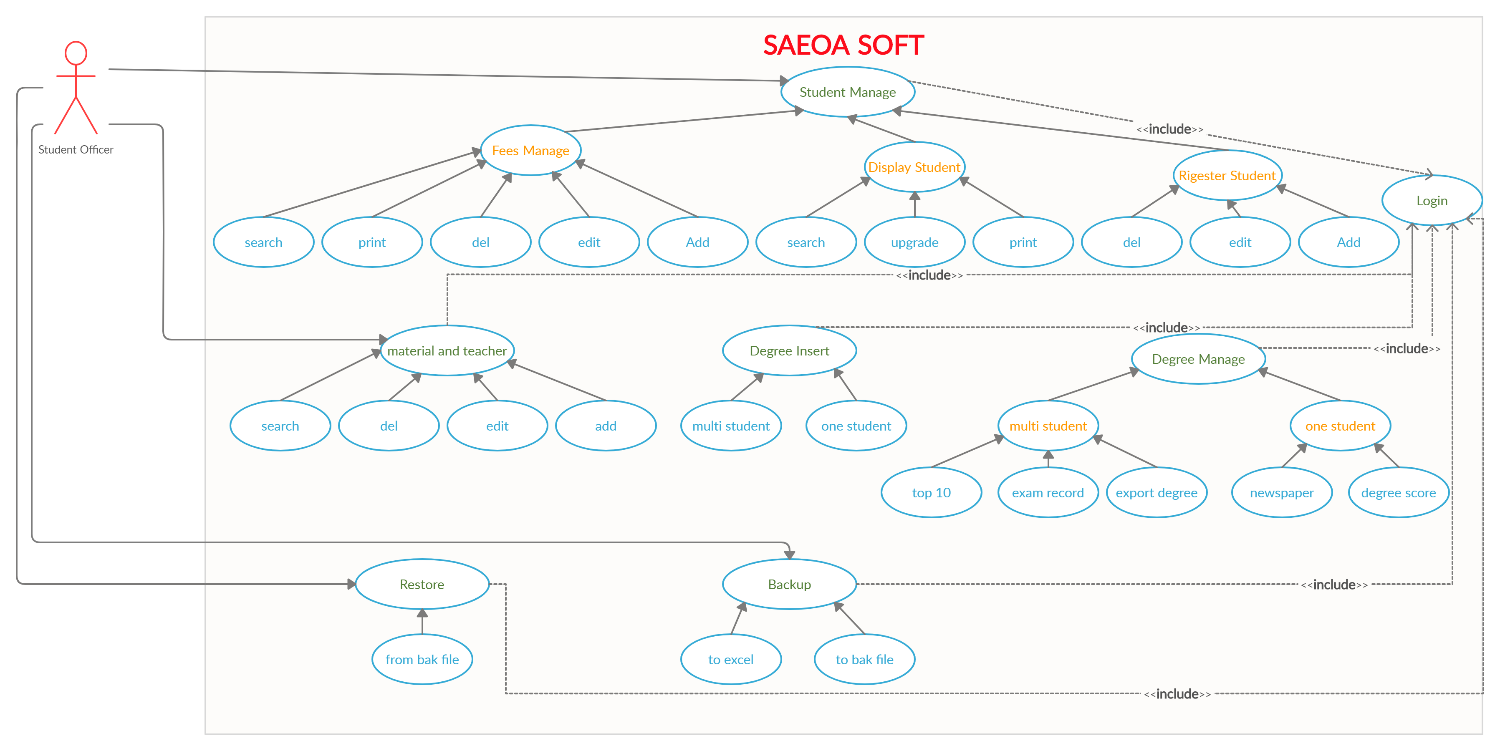
يوجد العديد من المخططات التابعة للغة UML ومن أشهرها:

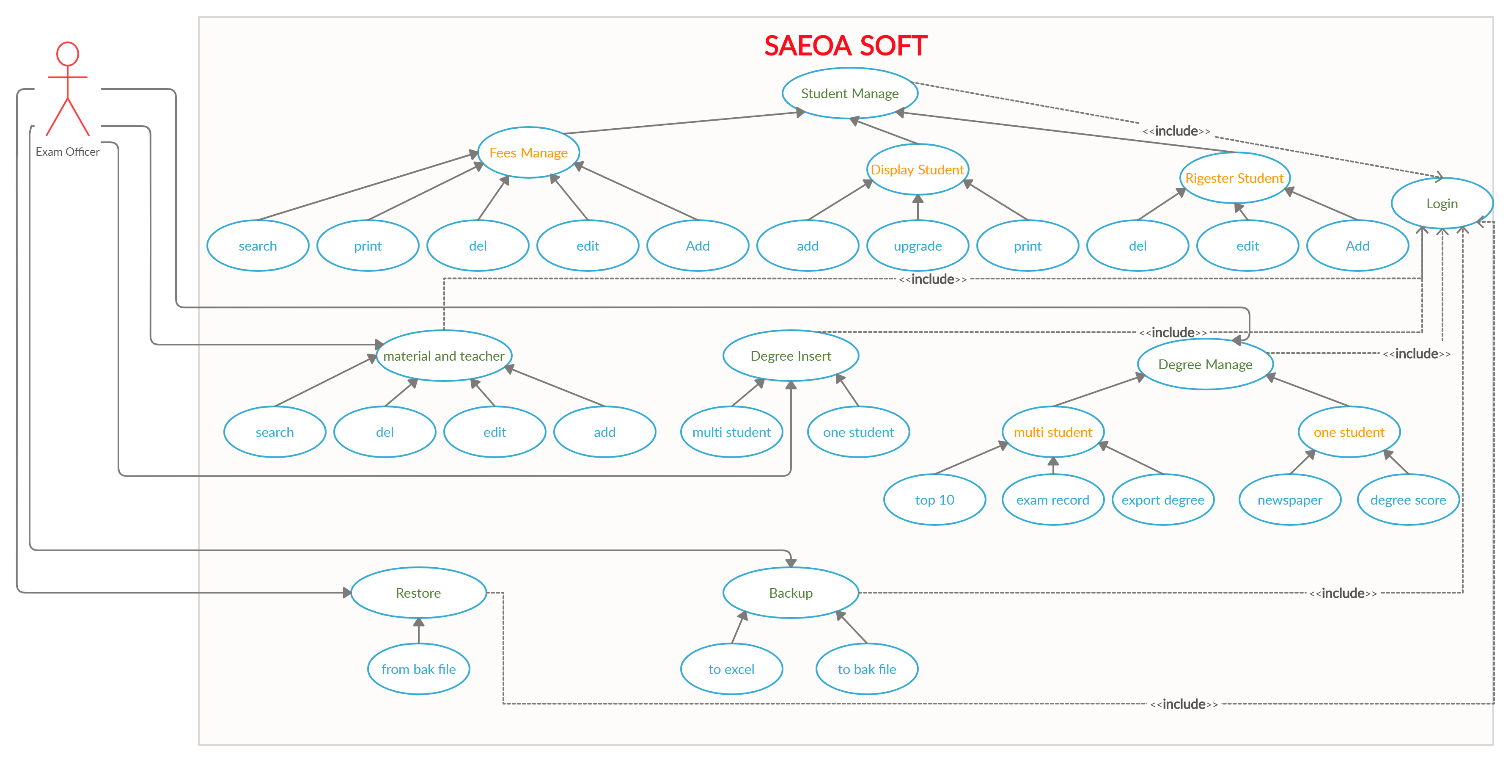
1. مخطط حالة الاستخدام "Use Case Diagram": هذا المخطط عبارة عن مجموعة من السيناريوهات التي تصف التفاعل بين المستخدم والنظام، فهو يعرض العلاقة بين الجهات الفاعلة "actors" وحالات الاستخدام "use cases"، ويعتبر كل من حالات الاستخدام "use cases" والجهات الفاعلة "actors" المكونان الرئيسيان لهذا المخطط، حيث الجهة الفاعلة "actor" تمثل المستخدم أو النظام الذي سيتفاعل مع النظام الذي تتم نمذجته، أما حالة الاستخدام "use case" عبارة عن رؤية خارجية للنظام بحيث تستعرض بعض الإجراءات التي يمكن أن يقوم بها المستخدم لإكمال المهمة.

يتكون مشروعنا من ثلاث جهات فاعلة وهم (المدير، موظف شؤون الطلاب، موظف شؤون الامتحانات)، حيث قمنا بإنشاء مخطط لكل فاعل، ويبين الشكل (2-1) مخطط حالات الاستخدام للفاعل "المدير":

**الشكل (2-1) مخطط حالات الاستخدام للفاعل "المدير".**

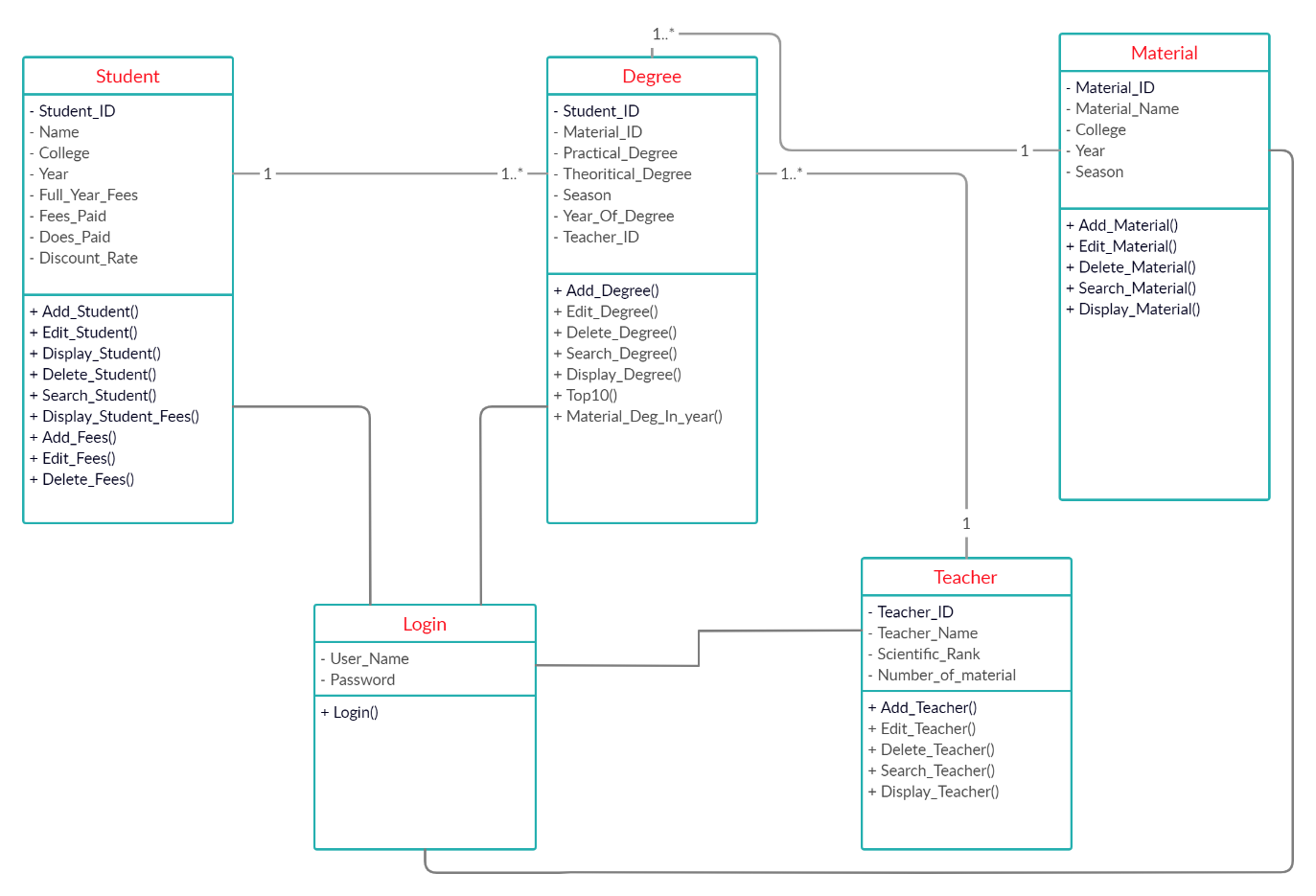
حيث يمكن لهذا الفاعل الوصول إلى جميع حالات الاستخدام الخاصة بالنظام، والشكل التالي يبين مخطط حالات الاستخدام للفاعل "موظف شؤون الطلاب":

**الشكل (2-2) مخطط حالات الاستخدام للفاعل "موظف شؤون الطلاب".**

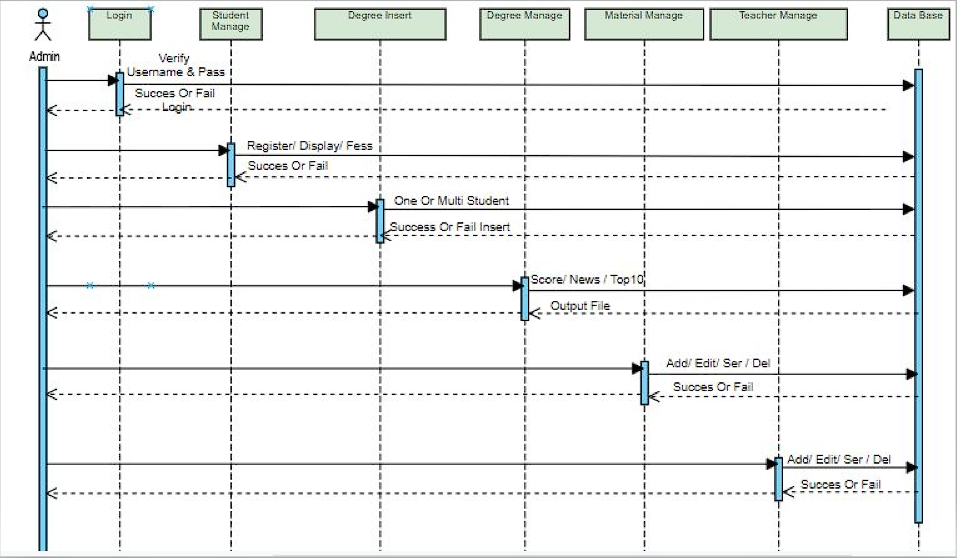
الفاعل السابق يستطيع الوصول إلى حالات الاستخدام التي تتعلق بإدارة بيانات الطلاب وبيانات المدرسين والمواد بالإضافة إلى نسخ البيانات احتياطياً واستعادتها، ويعرض الشكل التالي مخطط حالات الاستخدام للفاعل "موظف شؤون الامتحانات":

**الشكل (2-3) مخطط حالات الاستخدام للفاعل "موظف شؤون الامتحانات".**

أما بالنسبة لهذا الفاعل يستطيع الوصول إلى حالات الاستخدام التي تتعلق بإدارة وإدخال درجات الطلاب، بالإضافة إلى بيانات المدرسين والمواد، وأيضاً نسخ البيانات احتياطياً واستعادتها.

1. مخطط فئات النظام class diagram: هو عبارة عن مخطط هيكلي مهمته عرض الفئات بنظامٍ معيّن مع جميع العلاقات التي تربط بينها، وهو يساعد على فهم المشكلة بشكل أكبر، المكون الأساسي لهذا المخطط هو الفئات أو الصفوف والتي تأخذ شكل الصندوق box، كل صندوق يمكن أن يعرّف الدوال methods والخصائص attribute الخاصة بكل فئة، مخطط فئات النظام الخاص بمشروعنا مبين في الشكل التالي:

**الشكل (2-4) مخطط فئات النظام class diagram.**

1. ****المخطط التتابعي sequence diagram: مخطط التتابع أو مخطط التسلسل يستخدم لتوضيح التفاعل بين العناصر خلال تسلسل زمني ويصور الكائنات (objects) والأصناف (classes) التي يتضمنها سيناريو برمجي معين وتسلسل الرسائل المتبادلة بين الكائنات لتنفيذ السيناريو بشكل صحيح، مخططات التتابع مرتبطة مع حالة استخدام use case في العرض المنطقي للنظام قيد التنفيذ، يشار إلى مخطط التتابع أحيانا بمخطط الحدث أو سيناريو الحدث، ويعرض هذا المخطط العمليات أو الكائنات المختلفة على خطوط طولية تسمى (خطوط الحياة)، وأسهم أفقية توضح الرسائل المتبادلة بينها، تسمح هذه الطريقة بتفصيل طريقة عمل السيناريو أثناء وقت التنفيذ بطريقة رسومية، المخطط التتابعي لمشروعنا مبين في الشكل التالي:

**الشكل (2-5) المخطط التتابعي sequence diagram.**

**الفصل الثالث**

**بناء قاعدة البيانات**

**الفصل الثالث**

**قاعدة البيانات**

**مقدمة**

يبين هذا الفصل كيف قمنا ببناء قاعدة البيانات حيث أولاً قمنا بإنشاء الجداول ثم قمنا بالربط بين هذه الجداول بعد ذلك قمنا بوضع قيود مناسبة لبعض الحقول في الجداول وبعدها قمنا بإنشاء إجرائيات من أجل تسهيل التحكم بالجداول لإضافة البيانات والتعديل عليها.

**مفهوم قاعدة البيانات**

قاعدة البيانات هي مجموعة من عناصر البيانات المنطقيّة المرتبطة مع بعضها البعض بعلاقة رياضيّة، وتتكوّن قاعدة البيانات من جدول واحد أو أكثر، مثل السجل الخاص بالموظف الذي يتكوّن من عدّة حقول، مثل: رقم الموظف، واسم الجهاز، ودرجة الموظف، وتاريخ التعيين، والراتب، وبيانات الموظف التي تخزن في جهاز الحاسوب على نحوٍ منظّم، حيث يسهل لنا الحاسوب التعامل مع البيانات والبحث ضمن هذه البيانات، والتمكين من الإضافة والتعديل.

تتمتع قاعدة البيانات بخاصية هامة جداً تعتبر أحد أهم الأسباب الأساسية التي تدفع المستخدمين لاستخدامها، ألا وهي "الاستمرارية" أو "الدوام" في حفظ البيانات.

في الجانب التقني والبرمجي، فإن قاعدة البيانات Database هي عبارة عن مستودع تُحفظ البيانات فيه داخل جهاز الحاسوب أو المخدم، ويتمتع هذا المستودع بخاصية الاستمرارية في حفظ البيانات، ونعني بخاصية الاستمرارية هنا أنه في حال إطفاء جهاز الحاسوب أو إعادة تشغيله أو انقطاع التواصل معه، فإن قاعدة البيانات وما تحتويه من بيانات تبقى موجودة ومحفوظة دون أي خلل.

إن تنظيم عمل شؤون الطلاب يتطلب تخزين كم كبير من المعلومات و تصنيفها بشكل يسهل استخراج هذه البيانات عند حاجتها و كان أفضل حل هو استخدام لغة الاستعلامات القياسية SQL Server وهي اختصار لـ Structured Query Language ومعناها (لغة الاستعلامات القياسية)، وتستخدم لإدارة قواعد البيانات وللاستعلام عن البيانات و إدراجها و تحديثها وتعديلها، تتمتع لغةSQL بالعديد من المزايا، وقد تزايد استخدامها بشكل كبير في السنوات الأخيرة، وبخاصة كون الشركات تجمع المزيد والمزيد من المعلومات والبيانات التي يتوجب عليها تخزينها بشكل مستمر.

**مميزات لغة SQL**

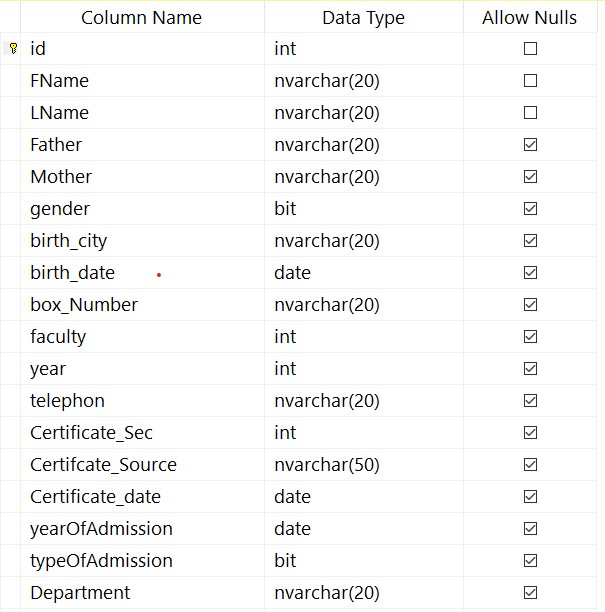
1. لغة عالمية: تعتبر هذه اللغة إحدى أكثر التقنيات استخدامًا في مجالات واختصاصات عديدة في كل مكان، وعند استخدامك لهذه اللغة فإنها ستحفزك بشكل تلقائي على تعلم لغات البرمجة المختلفة مثل C++ أو جافا أو بايثون وغيرها.
2. مفتوحة المصدر وسهلة التعلم والاستخدام: تعتبر هذه اللغة أسهل للتعلم نسبيًا من بقية لغات البرمجة. كما أنها مفتوحة المصدر، الأمر الذي يفتح الطريق واسعًا أمام المطورين. كما تدعم العديد من قواعد البيانات هذه اللغة مما يجعل التعامل معها أمرًا سهلًا.
3. إدارة ملايين صفوف البيانات: يمكنك استخدام جداول البيانات التقليدية لإدارة مجموعات البيانات والمعلومات الصغيرة والمتوسطة الحجم، لكننا سنحتاج إلى بديل عنها عند التعامل مع السجلات الضخمة. لذا سنتجه بشكل مباشر إلى لغة SQL التي يمكنها التعامل مع آلاف وملايين السجلات.
4. تطور التكنولوجيا:يمكن استخدام العديد من قواعد البيانات المتطورة للعمل على اللغة سواء أكان على جهاز الحاسب أو الهاتف المحمول وغيرهما.
5. تزايد الطلب: تقوم الشركات بالبحث عن الأفراد المؤهلين للتعامل مع SQL بشكل دائم، مع العلم أن أجور هؤلاء تكون مرتفعة جدًا.
6. لا حاجة للترميز: إذ يعتبر التعامل مع اللغة سهلًا جدًا ولا حاجة لكتابة الكثير من التعليمات البرمجية.
7. لغة تفاعلية: يمكن استخدامها للتواصل مع قواعد البيانات وتلقي إجابات على الأسئلة المعقدة في ثوانٍ معدودة.
8. طرق متعددة لعرض البيانات: يمكن للمستخدمين بمساعدة هذه اللغة عرض بنية قاعدة البيانات بطرق مختلفة.

**الجداول التي تتكون منها قاعدة البيانات**

تُعد الجداول كائنات أساسية في قاعدة البيانات لأنها تتضمّن كل البيانات أو المعلومات، يُعرّف الجدول على أنه مجموعة من عناصر البيانات المنظمة على شكل أعمدة (وهي تُعرّف باسمها) وصفوف أفقية، يحتوي الجدول على عدد محدد من الأعمدة لكنه يمكن أن يحتوي على أي عدد من الصفوف.

1. **جدول الطلاب**

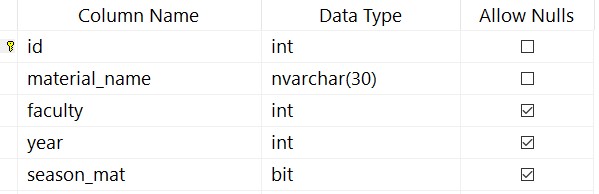
وهو جدول رئيسي يحتوي على بيانات الطلاب الشخصية والتي نستخدم بعضاً منها في القسم المخصص من هذا المشروع لقسم الامتحانات، أما بقية البيانات فهي معتمدة من قِبل شؤون الطلاب، ويبين الشكل (3-1) تصميم جدول الطلاب داخل قاعدة البيانات، حيث يتكون من الحقول التالية:

* الرقم الجامعي (id): وهو حقل المفتاح الرئيسي للجدول الذي لا يجب أن يتكرر ولا يجب أن يكون فارغاً (primary key).
* اسم الطالب (FName).
* الكنية (LName).
* اسم الأب (Father).
* اسم الأم (Mother).
* الجنس (gender).
* مكان الميلاد (birth\_city).
* تاريخ الميلاد (birth\_date).
* رقم القيد (box\_Number).
* الكلية (faculty).
* السنة الدراسية (year).
* رقم الهاتف (telephon).
* الشهادة الثانوية (Certificate\_Sec): وهذا الحقل له جدول خاص به.
* مصدر الشهادة (Certifcate\_Source).
* تاريخ الشهادة (Certificate\_date).
* عام القبول (yearOfAdmission).
* نوع القبول (typeOfAdmission).
* القسم (Department): اختصاص الطالب ضمن الكلية.

الشكل (3-1) تصميم جدول الطلاب داخل قاعدة البيانات.

1. **جدول المواد**

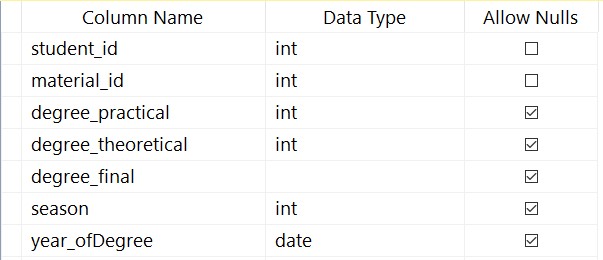
وهو أيضاً جدول رئيسي يحتوي على أسماء المواد الخاصة بكل كلية، حيث يبين الشكل (3-2) تصميم جدول المواد داخل قاعدة البيانات، وهو مكون من الحقول التالية:

* رمز المادة (id): وهو حقل المفتاح الرئيسي للجدول الذي لا يجب أن يتكرر ولا يجب أن يكون فارغ.
* اسم المادة (material\_name): وهو يقابل رمز المادة.
* الكلية (faculty): وهذا الحقل من أجل تصنيف المواد حسب الكلية التابعة لها.
* السنة الدراسية للمادة (year): العام الذي يتم تدريس المادة به (أولى، ثانية، ...).
* الفصل الدراسي (season\_mat): ترتيب المادة حسب الخطة الدرسية (فصل أول أو ثاني).

الشكل (3-2) تصميم جدول المواد داخل قاعدة البيانات.

1. **جدول الدرجات**

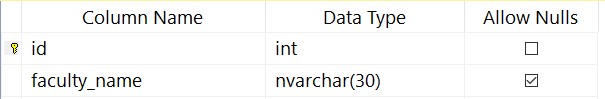
وهو جدول محوري يقوم عليه المشروع، حيث يبين الشكل (3-3) تصميم الجدول داخل قاعدة البيانات، وهو يتكون من الحقول التالية:

* الرقم الجامعي (student\_id).
* رمز المادة (material\_id).
* درجة العملي (degree\_practical).
* درجة النظري (degree\_theoretical).
* الدرجة النهائية (degree\_final).
* الدورة الامتحانية (season): ويمثل الدورة الامتحانية التي تم تقديم المادة فيها.
* تاريخ تقديم المادة (year\_ofDegree).

الشكل (3-3) تصميم جدول الدرجات داخل قاعدة البيانات.

1. **جدول الكليّات**

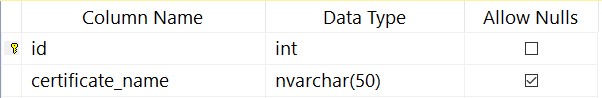
وهو جدول يضم كافة كليات الجامعة، ويبين الشكل (3-4) تصميم جدول الكليات داخل قاعدة البيانات، ويتكون من الحقلين التاليين:

* رمز الكلية (id).
* اسم الكلية (faculty\_name).

الشكل (3-4) تصميم جدول الكليات داخل قاعدة البيانات.

1. **جدول الشهادة الثانوية**

يبين الشكل (3-5) تصميم جدول الشهادة الثانوية داخل قاعدة البيانات، حيث يحتوي الحقلين التاليين:

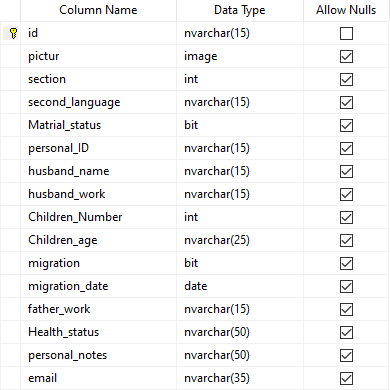
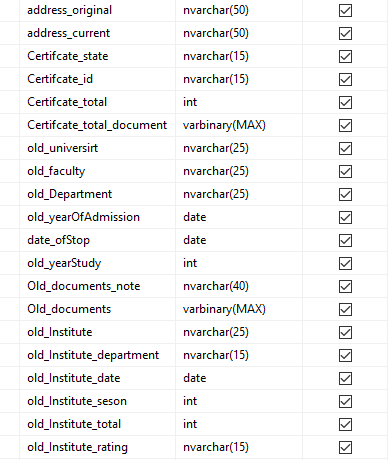
* رقم الشهادة (id).
* ****نوع الشهادة (certificate\_name).

الشكل (3-5) تصميم جدول الشهادة الثانوية داخل قاعدة البيانات.

1. **جدول تفاصيل الطالب**

يضم هذا الجدول بيانات الطالب التفصيلية، يبين الشكل (6-3) تصميم الجدول داخل قاعدة البيانات، حيث يتضمن هذا الجدول الحقول التالية:

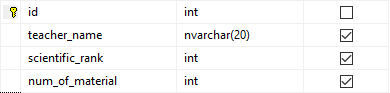
* الرقم الجامعي(id).
* الصورة الشخصية للطالب (pictur).
* السنة الدراسية (section).
* اللغة الأجنبية (second\_language).
* الحالة الاجتماعية (Matrial\_status).
* رقم الهوية (personal\_ID).
* اسم الزوج (husband\_name).
* عمل الزوج (husband\_work).
* عدد الأولاد (Children\_Number).
* عمر الأولاد (Children\_age).
* نازح أم مقيم (migration).
* تاريخ النزوح (migration\_date).
* عمل الأب (father\_work).
* الحالة الصحية (Health\_status).
* ملاحظات شخصية (personal\_notes).
* الايميل (email).
* العنوان\_الأصلي (address\_original).
* العنوان\_الحالي (address\_current).
* حالة الشهادة (Certifcate\_state).
* رقم الشهادة (Certifcate\_id).
* المجموع (Certifcate\_total).
* ملف الشهادة (Certifcate\_total\_document).
* اسم الجامعة السابقة (old\_universirt).
* اسم الكلية السابقة (old\_faculty).
* اسم القسم السابق (old\_Department).
* عام القبول (old\_yearOfAdmission).
* عام التوقف (date\_ofStop).
* السنة الدراسية السابقة (old\_yearStudy).
* ملاحظات المستندات القديمة (Old\_documents\_note).
* المستندات القديمة (Old\_documents).
* المعهد القديم (old\_Institute).
* قسم المعهد القديم (old\_Institute\_department).
* تاريخ المعهد القديم (old\_Institute\_date).
* فصل المعهد القديم (old\_Institute\_seson).
* مجموع المعهد القديم (old\_Institute\_total).
* تصنيف المعهد القديم (old\_Institute\_rating).

**الشكل (6-3) تصميم جدول تفاصيل الطالب داخل قاعدة البيانات.**

1. **جدول المدرس**

يضم أسماء المدرسين في الجامعة، حيث يبين الشكل (7-3) تصميم الجدول داخل قاعدة البيانات، ويتضمن هذا الجدول الحقول التالية:

* رقم المدرس (id).
* اسم المدرس (teacher\_name).
* المرتبة العلمية (scientific\_rank).
* عدد المواد (num\_of\_material).

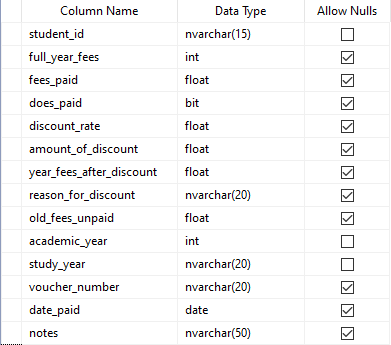


الشكل (7-3) تصميم جدول المدرس داخل قاعدة البيانات.

1. **جدول رسوم الطالب**

يضم هذا الجدول رسوم الطالب، حيث يبين الشكل (3-8) تصميم الجدول داخل قاعدة البيانات، ويتضمن هذا الجدول الحقول التالية:

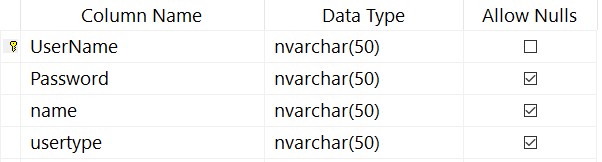
* الرقم الجامعي (student\_id).
* رسوم السنة الكاملة (full\_year\_fees).
* الرسوم المدفوعة (fees\_paid).
* هل تم الدفع (does\_paid).
* معدل الخصم (discount\_rate).
* مبلغ الخصم (amount\_of\_discount).
* رسوم السنة بعد الخصم (year\_fees\_after\_discount).
* سبب الخصم (reason\_for\_discount).
* الرسوم القديمة غير المدفوعة (old\_fees\_unpaid).
* العام الدراسي(academic\_year).
* سنة الطالب الدراسية (study\_year).
* رقم الوصل (voucher\_number).
* تاريخ الدفع (date\_paid).
* ملاحظات (notes).



الشكل (8-3) تصميم جدول رسوم الطالب داخل قاعدة البيانات.

1. **جدول المستخدم**

يضم المستخدمين المسموح لهم بالدخول لقاعدة البيانات، حيث يبين الشكل (9-3) تصميم الجدول داخل قاعدة البيانات، ويتضمن هذا الجدول الحقول التالية:

* الاسم المُستخدم الخاص بقاعدة البيانات (UserName).
* كلمة المرور (Password).
* اسم المستخدم الحقيقي (name).
* نوع الوصول (usertype): لتحديد صلاحيات المستخدم.

الشكل (9-3) تصميم جدول المستخدم داخل قاعدة البيانات.

**الربط بين الجداول**

عند إنشاء جداول في قاعدة البيانات، فإن الظاهر لنا أننا نقوم ببناء جداول منفصلة وغير مترابطة، ولكننا في الواقع العملي نحتاج لربط هذه الكيانات المنفصلة بحيث تُبنَى علاقات تحكم البيانات الموجودة في هذه الجداول، وتحكم طريقة التعامل مع هذه البيانات.

تنشأ العلاقة بين جدولين عندما يُربط عمودان فيهما مع بعضهما عن طريق وجود قيود مطبقة على العمودين، بحيث يكون قيد المفتاح الرئيسي على عمود في الجدول "الأب" وقيد المفتاح الأجنبي على العمود في الجدول "الابن"، وعادة يكون اسم العمودين واحدًا في كلا الجدولين.

**فوائد استخدام علاقات الربط**

1. التخلص من مشكلة تكرار البيانات عن طريق فصلها وحفظها في أكثر من جدول، فمشكلة تكرار البيانات هي عدو مستخدمي قواعد البيانات ومسؤوليها، لأنها تتسبب بزيادة حجم قاعدة البيانات بقدر كبير وبسرعة، وترفع السرعات المطلوبة لتنفيذ الاستعلامات، وتجعل من موضوع صيانة قاعدة البيانات كابوسا مقلقا.
2. الحفاظ على دقة وسلامة البيانات في قاعدة البيانات، فمع وجود العلاقات بين الجداول، سوف تضمن مثلاً عدم وجود كتاب مُعار ليس له شخص استعاره، أو عنوان وهمي ليس له صاحب، وقس على ذلك العديد من الأمثلة.
3. استخراج البيانات من أكثر من جدول بكفاءة وسرعة عن طريق بناء جمل ربط استعلامية تطلب المعلومات من أعمدة مختلفة في جداول مختلفة، وإخراج النتيجة بطريقة مفيدة ومرتبة.

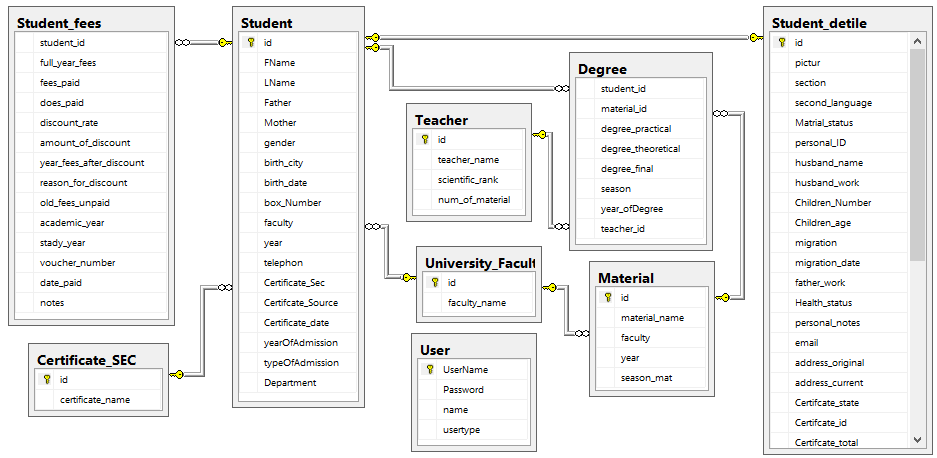
**أنواع العلاقات**

توجد أربعة أنواع من العلاقات بين الجداول كالتالي:

1. علاقة واحد إلى واحد (One-to-One).
2. علاقة واحد إلى كثير أو علاقة كثير إلى واحد (One-to-Many / Many-to-One).
3. علاقة كثير إلى كثير (Many-to-Many).
4. علاقة المرجعية الذاتية (Self-Referencing).

يوضح الشكل (3-10) العلاقات بين الجداول التي تم ذكرها سابقاً، والعلاقات هي:

* تم ربط حقل رقم الطالب من جدول الطالب مع حقل رقم الطالب من جدول الدرجة.
* تم ربط حقل رقم الطالب من جدول الطالب مع حقل رقم الطالب من جدول تفاصيل الطالب.
* تم ربط حقل رمز المادة من جدول الدرجة مع حقل رمز المادة من جدول المادة.
* تم ربط حقل رمز الكلية من جدول الكليّة مع حقل رمز الكلية من جدول المادة.
* تم ربط حقل رمز الكلية من جدول الكليّة مع حقل رمز الكلية من جدول الطالب.
* تم ربط حقل رمز الشهادة من جدول الشهادة مع حقل رمز الشهادة من جدول الطالب.
* تم ربط حقل رقم المدرس من جدول المدرس مع حقل رقم المدرس من جدول الدرجة.
* تم ربط حقل رمز المادة من جدول المادة مع حقل رمز المادة من جدول الدرجة.

الشكل (10-3) مخطط العلاقات بين جداول قاعدة البيانات.

حيث نوع علاقة الربط المستخدمة في معظم العلاقات هي علاقة واحد لكثير (1 to m).

**الإجرائيات المخزنة**

هي عبارة عن جملة استعلام مخزنة في قاعدة البيانات يمكن أن تقوم بمجموعة من الأعمال، كإضافة سجل ثم تعديل البيانات في سجل آخر، والقيام بالحسابات على مجموعة أُخرى من السجلات، وذلك اعتباراً من أمر واحد من المستخدم.

عند استخدام الإجراءات المخزنة نقوم بإنشاء إجرائية مع معاملات دخل وخرج بحيث يمكننا استدعاؤها بالصورة نفسها المُستخدمة في تعبيرات SQL المعيارية.

تحتوي الإجراءات المخزنة تعبير أو مجموعة من تعبيرات SQL والتي تنفذ كجزء من تنفيذ الإجرائية مع إمكانيات التحكم بتدفق التنفيذ أو إضافة حلقات أو التنفيذ الشرطي، مما يعطي الإجرائيات المخزنة تمييزاً كبيراً.

يتكون مشروعنا من 62 إجرائية، ويمكن الإطلاع عليهم من خلال الملحق، والإجرائيات هي:

1. **إجرائية إضافة رسوم (add\_fees):** تقوم هذه الإجرائية بإضافة الرسوم الجامعية إلى جدول الرسوم مع كافة البيانات الخاصة بالرسوم من إعفاء ورقم وصل التسجيل ... الخ.
2. **إجرائية إضافة مادة (add\_material):** هذه الإجرائية مخصة لإضافة مادة جديدة إلى جدول المواد.
3. **إجرائية إضافة طالب (add\_student):** تقوم هذه الإجرائية بإدخال البيانات الشخصية لطالب جديد إلى جدول الطلاب.
4. **إجرائية إضافة تفاصيل طالب (add\_student\_detile):** تقوم هذه الإجرائية بإضافة البيانات التفصيلية للطالب إلى جدول تفاصيل طالب.
5. **إجرائية إضافة مدرس (add\_teacher):** تقوم هذه الإجرائية بإضافة بيانات مدرس جديد إلى جدول المدرسين.
6. **إجرائية عرض جميع الطلاب مع درجاتهم (all\_degree\_with\_student):** تقوم هذه الإجرائية بجلب بيانات جميع الطلاب مع درجاتهم العملية والنظرية في جميع المواد.
7. **إجرائية عرض حالة الرسوم لجميع الطلاب في كل السنوات (all\_fees\_with\_student):** هذه الإجرائية تجلب حالة الرسوم لكل طالب (دفع أم لم يدفع) في السنوات الخمسة.
8. **إجرائية جلب مواد إحدى الكليات (all\_mat):** تقوم هذه الإجرائية بعرض بيانات جميع المواد لإحدى الكليات في الجامعة.
9. **إجرائية نسخة إحتياطية (Backup\_database):** تقوم هذه الإجرائية بأخذ نسخة إحتياطية عن قاعدة البيانات من اجل استعادتها في حال حدث خطأ ما.
10. **إجرائية أخذ نسخة عن الدرجات (backup\_degree):** تستخدم لإنشاء نسخة إحتياطية من الدرجات.
11. **إجرائية التأكد من وجود تفاصيل للطالب (chek\_student\_detile):** تقوم هذه الإجرائية بجلب البيانات التفصيلية لطالب ما اعتماداً على الرقم الجامعي له.
12. **إجرائية كشف الدرجات (DEGREE\_SCORE):** تستخدم هذه الإجرائية لإصدار كشف درجات لطالب، حيث تقوم بجلب اسم المادة والدرجات الخاصة بالمادة (عملي ونظري والدرجة النهائية)، نمرر لهذه الإجرائية الرقم الجامعي للطالب المراد إيجاد كشف درجات له بالإضافة للعام الدراسي والفصل الدراسي للمادة ثم تقوم هذه الإجرائية بجلب البيانات.
13. **إجرائية حذف الدرجة (delete\_degree):** تقوم هذه الإجرائية بحذف درجة محددة مع بياناتها (الدرجة العملية- الدرجة النظرية- تاريخ تقديمها....) وذلك بناءً على الرقم الجامعي للطالب صاحب الدرجة.
14. **إجرائية حذف رسوم طالب (delete\_fees):** هذه الإجرائية تقوم بحذف رسوم الطالب المحدد من جدول الرسوم بناءً على الرقم الجامعي للطالب والسنة الدراسية والعام.
15. **إجرائية حذف مادة (delete\_material):** هذه الإجرائية تقوم بحذف مادة محددة من جدول المواد بناءً على رمز المادة (id).
16. **إجرائية حذف بيانات طالب (delete\_student):** هذه الإجرائية تقوم بحذف بيانات الطالب الشخصية والتفصيلية ذو الرقم الجامعي (id) المحدد مع حذف كافة الدرجات المرتبطة به.
17. **إجرائية حذف بيانات مدرس (delete\_teacher):** هذه الإجرائية مخصصة لحذف مدرس محدد من جدول المدرسين بناءً على الرقم الخاص به (id).
18. **إجرائية تعديل الدرجة (edit\_degree):** تستخدم هذه الإجرائية لتعديل كافة البيانات الخاصة بالدرجة، وذلك بناءً على رقم المادة والرقم الجامعي للطالب.
19. **إجرائية تعديل درجة العملي (edit\_degree\_prac):** تستخدم هذه الإجرائية لتعديل البيانات الخاصة بدرجة العملي، وذلك بناءً على رقم المادة والرقم الجامعي للطالب.
20. **إجرائية تعديل رسوم طالب (edit\_fess):**تستخدم هذه الإجرائية لتعديل البيانات الخاصة برسوم طالب ما.
21. **إجرائية تعديل مادة (edit\_material):** تستخدم هذه الإجرائية لتعديل البيانات الخاصة بمادة.
22. **إجرائية تعديل الطالب (edit\_student):** تُستخدم هذه الإجرائية لتعديل البيانات الشخصية الخاصة بالطالب، وذلك بناءً على الرقم الجامعي للطالب.
23. **إجرائية تعديل بيانات مدرس (edit\_teacher):** تستخدم هذه الإجرائية لتعديل البيانات الخاصة بمدرس.
24. **إجرائية السجل الامتحاني (exam\_record):** صُممت هذه الإجرائية لتقوم بجلب كافة درجات المواد المُقدمة في العام الدراسي الحالي، وذلك بهدف حفظ نسخة من الدرجات، حيث نمرر لهذه الإجرائية سنة المادة والعام الدراسي.
25. **إجرائية عدد الطلاب المتقدمين للامتحان (exam\_takers):** تقوم هذه الإجرائية بحساب عدد الطلاب المتقدمين لامتحان مادة ما لاستخدامه في الإحصائية أثناء إصدار الدرجات، نمرر لهذه الإجرائية رقم المادة والعام الدراسي.
26. **إجرائية عرض المواد الراسبة (fail\_degree):** باستخدام هذه الإجرائية يتم جلب المواد الراسبة لطالب محدد ودرجاته في هذه المواد.
27. **إجرائية عدد الطلاب الراسبين (fail\_num):** تقوم هذه الإجرائية بالاستعلام عن عدد الطلاب الراسبين في إحدى المواد عن طريق تمرير رقم المادة والسنة الدراسية.
28. **إجرائية جلب الدرجات (Get\_All\_Degree):** تقوم هذه الإجرائية بجلب بيانات كافة الدرجات من جدول الدرجة.
29. **إجرائية جلب الرسوم (get\_all\_fees):** عن طريق هذه الإجرائية يتم جلب بيانات الرسوم لكافة الطلاب.
30. **إجرائية جلب بيانات المواد (Get\_All\_Material):**

تظهر البيانات الخاصة بكل المواد من جدول المادة مثل رقم المادة (رمزها)، واسم المادة وذلك بناءً على الكليّة والعام الدراسي والفصل الدراسي للمادة.

1. **إجرائية عرض أسماء طلاب سنة ما (get\_all\_name\_in\_year):** تقوم هذه الإجرائية بجلب أسماء طلاب سنة معينة مع أرقامهم الجامعية.
2. **إجرائية جلب بيانات الطلاب (Get\_All\_student):** يتم ضمن هذه الإجرائية جلب جميع البيانات لكافة الطلاب من جدول الطالب.
3. **إجرائية جلب بيانات الطلاب (Get\_All\_student\_inYear):** يتم ضمن هذه الإجرائية جلب جميع البيانات لطلاب سنة ما من جدول الطالب.
4. **إجرائية جلب بيانات المعلمين (Get\_all\_teacher):** باستخدام هذه الإجرائية يتم جلب بيانات كافة المدرسين من جدول المدرس.
5. **إجرائية جلب المعدل (Get\_Avg\_Degree):** تقوم هذه الإجرائية بحساب معدل درجات الطالب في العام الدراسي وذلك لاستخدامه في كشف الدرجات، حيث نحدد لهذه الإجرائية العام الدراسي والرقم الجامعي للطالب، يتم الحساب عن طريق جمع الدرجات النهائية للمواد ثم تقسيمها على عدد هذه المواد، تحتفظ هذه الإجرائية بالقيمة المحسوبة.
6. **إجرائية جلب الدرجة في الفصل (get\_degreesWithSeson):** هذه الإجرائية تقوم بجلب درجات طالب ما في فصل دراسي محدد وسنة محددة.
7. **إجرائية جلب درجات المواد لسنة ما (get\_mat\_year):** تقوم هذه الإجرائية بجلب جميع درجات مادة ما لطلاب سنة محددة.
8. **إجرائية جلب درجات المواد الناجحة لسنة ما (get\_mat\_year2):** تقوم هذه الإجرائية بجلب الدرجات الناجحة لمادة ما والتابعة لطلاب سنة محددة.
9. **إجرائية جلب درجات المواد الراسبة لسنة ما (get\_mat\_year3):** تقوم هذه الإجرائية بجلب الدرجات الراسبة لمادة ما والتابعة لطلاب سنة محددة.
10. **إجرائية جلب جميع المواد (Get\_Material):** تجلب هذه الإجرائية بيانات جميع المواد من جدول المادة.
11. **إجرائية جلب بيانات رسوم طالب (get\_std\_fees):** تجلب هذه الإجرائية بيانات رسوم طالب محدد من جدول الرسوم.
12. **إجرائية جلب اسم طالب (Get\_stdname):** تعيد هذه الإجرائية اسم الطالب ذو الرقم الجامعي المُدخل.
13. **إجرائية جلب الرقم الجامعي لطالب (Get\_stdnum):** تعيد هذه الإجرائية الرقم الجامعي للطالب ذو الاسم المُدخل.
14. **إجرائية جلب بيانات مدرس (Get\_stdnum):** تعيد هذه الإجرائية بيانات مدرس وفقاً لاسمه أو رقمه.
15. **إجرائية جلب العشرة الأوائل (get\_top10):** تقوم هذه الإجرائية بإرجاع بيانات الطلاب العشرة الأوائل عن سنة ما، حيث تعيد البيانات التالية (اسم الطالب، الرقم الجامعي، المعدل).
16. **إجرائية إدخال الدرجة (isert\_degree):** تقوم هذه الإجرائية بإدخال درجة الطالب والبيانات الخاصة بها إلى جدول الدرجة.
17. **إجرائية عدد المواد الراسبة (not\_success\_year):** تُعيد هذه الإجرائية عدد المواد الراسبة لطالب في سنة ما، حيث نمرر لها البارامترات التالية (السنة، عدد المواد، الكلية).
18. **إجرائية جلب المواد غير المقدمة (null\_matirial):** تستخدم هذه الإجرائية لإرجاع المواد غير المقدمة لطالب محدد.
19. **إجرائية جلب بيانات المادة الراسبة (replace\_fail\_degree):** تستخدم هذه الإجرائية لجلب بيانات الدرجة الراسبة لطالب محدد في مادة ما.
20. **إجرائية استعادة نسخة احتياطية (restore\_db):** باستخدام هذه الإجرائية يتم استعادة نسخة احتياطية سابقة لقاعدة البيانات.
21. **إجرائية البحث عن الطالب بالرقم (Search\_by3id):** بواسطة هذه الإجرائية يمكن البحث عن طالب ما بإدخال آخر ثلاثة أرقام من الرقم الجامعي.
22. **إجرائية البحث عن الدرجة (Search\_Degree):** تقوم هذه الإجرائية بالبحث عن الدرجة إما عن طريق الرقم الجامعي للطالب المراد عرض درجته، أو عن طريق رقم المادة (رمزها).
23. **إجرائية البحث عن مادة عبر الرقم (search\_material\_by\_id):** تعيد هذه الإجرائية بيانات مادة ما وفقاً لرمز المادة المُدخل.
24. **إجرائية البحث عن مادة عبر الاسم (search\_material\_by\_name):** تعيد هذه الإجرائية بيانات مادة ما وفقاً لاسم المادة المُدخل.
25. **إجرائية البحث عن الطالب بالاسم (Search\_Student\_name):** تقوم هذه الإجرائية بالبحث عن الطالب عن طريق اسمه (الاسم والكنية أو الاسم الكامل أو الكنية فقط)، إذا تطابق البحث يتم إرجاع كافة بيانات الطالب.
26. **إجرائية البحث عن معلم عبر الرقم (search\_teacher\_by\_id):** تقوم هذه الإجرائية بالبحث عن المدرس عن طريق رقمه، إذا تطابق البحث يتم إرجاع كافة بيانات المدرس.
27. **إجرائية البحث عن معلم عبر الاسم (search\_teacher\_by\_name):** تقوم هذه الإجرائية بالبحث عن مدرس ما عن طريق اسمه (الاسم فقط أو الاسم والكنية أو الكنية فقط)، إذا تطابق البحث يتم إرجاع كافة بيانات المدرس.
28. **إجرائية تسجيل الدخول (Sp\_Login):** يتم عن طريق هذه الإجرائية التحقق من اسم المستخدم الذي يحاول تسجيل الدخول وكلمة المرور الخاصة به، إذا تم التطابق مع البيانات في جدول المستخدمين تنجح عملية تسجيل الدخول وإلا يقوم بإظهار خطأ.
29. **إجرائية تفاصيل الطالب (Student\_detiles):** تعرض هذه الإجرائية البيانات الشخصية للطالب وبيانات الجامعة وذلك بناءً على الرقم الجامعي، حيث ندخل الرقم الجامعي للطالب فيتم عرض البيانات.
30. **إجرائية عدد الطلاب الناجحين (success\_num):** تعيد هذه الإجرائية عدد الطلاب الناجين في مادة مقدمة في سنة معينة.
31. **إجرائية إيجاد الطلاب الناجحين (success\_year):** تقوم هذه الإجرائية بإيجاد الطلاب الناجحين في سنة ما، وذلك تبعاً لعدد مواد الترفع المُدخل.
32. **إجرائية ترفيع الطلاب (success\_year\_ok):** وظيفة هذه الإجرائية هي ترفيع الطلاب الناجحين في سنة ما إلى السنة التالية وذلك تبعاً لعدد مواد الترفع المُدخل.

**القيود**

يعد موضوع القيود من المواضيع الهامة جداً في قواعد البيانات، ويجب على المبرمج ومسؤول قواعد البيانات الإلمام بأنواع القيود وكيفية تعريفها والتعامل معها.

الهدف العام من القيود هو الحفاظ على سلامة البيانات Data integrity ودقتها، نستطيع أن نعرف القيد على أنه شرط يحكم العمود أو الجدول ويتأكد من الالتزام بهذا الشرط عند إجراء عمليات مثل الإضافة والتعديل والحذف (على سبيل الذكر لا الحصر)، وفي حال لم يحترم الشرط فإن العملية التي أصدرت الأمر ترفض ويوقف تنفيذها ويتراجع عن أي أثر أحدثته.

القيود الموجودة في المشروع:

* **قيود جدول المواد**

قمنا بوضع قيد على حقل السنة الدراسية للمادة حيث يجب ألا يزيد عن 6 باستخدام القيد check.

* **قيود جدول الدرجات**
  + قمنا بجعل حقل الدرجة النهائية (degree\_final) حقل محوسب computed column حيث سيتم وضع حاصل جمع حقل الدرجة العملية (degree\_practical) مع حقل الدرجة النظرية(degree\_theoretical) بشكل تلقائي.
  + وضعنا قيد على حقل الدرجة العملية (degree\_practical) بحيث لا تتجاوز درجة العملي 40.
  + قيد حقل الدرجة النظرية (degree\_theoretical): حيث يجب ألا تزيد الدرجة عن 100.
  + قيد حقل الدورة الامتحانية (season): حيث لا يزيد عن أربع فصول.
* **قيود جدول الطلاب**
  + - قيد حقل العام الدراسي (year): حيث لا يجب أن يتجاوز الـ 10.
    - قيد حقل عام القبول (yearOfAdmission): حيث لا يجب أن يكون قبل العام 2015 وهو العام الذي تأسست فيه الجامعة.

الآن بعد أن قمنا بإنشاء الجداول وربطها مع بعض ووضع قيود على بعض حقولها وإنشاء إجرائيات من أجل الاستعلام أو التعديل على البيانات ضمن الجداول، نكون قد انتهينا من بناء قاعدة البيانات، حيث سننتقل إلى تصميم واجهة البرنامج في الفصل التالي.

**الفصل الرابع**

**تصميم واجهة البرنامج**

**الفصل الرابع**

**تصميم واجهة البرنامج**

**مقدمة**

يشرح هذا الفصل أهمية البرمجة وخاصةً البرمجة باستخدام لغة C#، ثم يفصل كيف قمنا بتصميم واجهات البرنامج وتنظيم الأكواد في صفوف داخل مجلدات وذلك ضمن بيئة الفيجوال استوديو، وإنشاء إجرائيات وأحداث تساعد وتسهل العمل.

**أهمية لغة C#**

أصبحت البرمجة هي نبض العالم التقني الحديث القائم على الحوسبة والكومبيوتر لذلك زادت أهمية البرمجة منذ عدة سنوات بشكل ملحوظ وذلك يرجع إلى مساهماتها الكبيرة في الثورة التكنولوجية، للبرمجة لغات كما هو الحال بالنسبة للبشر فهناك أكثر من 250 لغة برمجة حول العالم وكل لغة لها استخداماتها الخاصة.

الـ C# لغة من لغات برمجة الحاسوب، وهي لغة برمجة مختلفة الأنماط وذلك لأنها تعتمد على عدة أساليب برمجية مختلفة مثل الأسلوب السكوني والديناميكي والإشتقاقي، طورت من قبل شركة مايكروسوفت (Microsoft) عام 2000م من أجل تطوير برنامج NET. (برنامج خاص بتشغيل البرامج على الحاسوب)، أي لتعتمدها بشكل أساسي في إنتاج تطبيقات لمنصاتها المختلفة، والخروج من مشكلة الجافا وقضية شركة SUN، وقد تأثرت تلك اللغة بالعديد من اللغات مثل لغة ++C وJava، وعلى الجانب الأخر أثرت أيضاً في لغة D و #F.

سميت لغة سي شارب بهذا الاسم لعدة أسباب منها كونها مقتبسة عن لغة ++C، حيث نلاحظ أنها تتكون من علامة + مرتين، بينما #C نلاحظ أنها تتكون من علامة + أربعة مرات وهذا كان تعليل بعض المصادر على أن لغة #C جاءت بإضافات عن ++C، و يشاع أن سبب تسميتها بهذا الاسم يرجع إلى اقتباس رمز # من رمز موسيقي اعتماداً أن سطور البرمجة تشبه النوتة الموسيقية، وقد تم تسمية اللغة في البداية C فقط، وهي كانت اختصار لكلمة Cool ولكن سرعان ما تم تغيير اسمها إلى #C بسبب كون Cool علامة تجارية مسجلة، تضم اللغة 7 إصدارات، تم إصدار النسخة الأخيرة عام 2016م.

**خصائص لغة #C**

* رغم أن C# صممت على بيئة الويندوز، إلا أنها تدعم إنشاء تطبيقات على بيئات متعددة مثل لينوكس وغيرها من البيئات الأخرى.
* امتدت استخدامات C# لتدعم بناء تطبيقات الويب وتطبيقات الهاتف وذلك بالاستعانة بأنظمة خاصة وبرامج مساعدة.
* لغة C# لغة متطورة قوية جداً، وبالرغم من ذلك فتتميز بالسهولة التامة في التعامل والاستخدامات العديدة جداً.
* بما أن لغة C# تكتب ضمن إطار .NET فلن تجد أي صعوبة في كتابة الأكواد وستستطيع العمل مع مختلف البرامج بسهولة حتى البرامج الضخمة والمعقدة جداً.
* مصادر لغة C# عديدة للغاية، فتتواجد الكتب باللغة العربية والكتب الأجنبية وتتواجد الدورات المتنوعة على اليوتيوب وعلى المواقع المختلفة والتي تتوافر بشكل مجاني، وبالتالي ستستطيع تعلم C# بطرق عديدة.
* إن C# ملائمة لبرمجة التطبيقات بالنظم المضمنة والأنظمة المضيفة.
* مقارنة بلغة Java فإن لغة C# تولد ملفات بامتدادات أكبر مما تولده الجافا مثل امتداد exe وdll.
* **مناسبة للمبتدئين**

لأنها لغة بسيطة وسهلة الإستخدام، ولأنها تعتبر لغة عالية المستوى High Level Language، فيمكن للإنسان الذي يتقن اللغة الإنجليزية، فهم معظم مصطلحاتها، لأنها توفر كلمات محجوزة مفهومة، تضعها بين يديك لأداء إجراءات معينة، ثم تحولها إلى لغة وسيطة Intermediate Language، ثم إلى لغة الآلة Machine Language، فكل كلمة من كلمات لغة C#، يقابلها عشرات الجمل في لغة الألة، مما يدعك تركز على فكرة تطبيقك، ولا تقلق بشأن الإجراءات المعقدة تلك، فمثلاً استخدامك للكلمة المحجوزة Write، يخبر الحاسوب بأن يقوم بطباعة جملة على الشاشة، فنحن هنا لا نهتم أبداً كيف قام بالطباعة، بل نهتم أنه قام بذلك فقط، الأمر الذي إن حاولت كتابته بلغة الألة، قد يستغرق منك عشرات الجمل، فضلاً عن الوقت المهدور.

* **تزيد من إنتاجيتك**

لأنها لغة تعتمد على نظام الكتابة Static Type، فيوجد إمكانية اكتشاف الأخطاء في تطبيقك قبل بناءه، تسهل عليك تتبعها وتبسيطها لحلها لاحقاً، بالإضافة إلى أن لغة C# تفرض عليك قواعد معينة لكتابة الأسطر البرمجية، مما يجعل برنامجك سهل الصيانة وذو مرونة عالية.

* **عدد مستخدميها كبير**

فتعداد مستخدمي اللغة من أهم عوامل قوتها، فكلما زاد عدد مستخدميها زاد غناها وتنوعها، وستدرك قيمة ذلك إن تخصصت في تطوير التطبيقات، خصوصاً إن واجهتك مشكلة ولم تستطع حلها، فلو قمت بالبحث عنها ستجد أنها قد طرحت سابقاً، وأنه تم الإجابة عنها أيضاً، بالإضافة إلى أنك ستجد أدوات تم كتابتها بواسطة مطورين، قاموا ببنائها لتسهل عليك عملك وتزيد من إنتاجيتك بشكل أفضل. وبما أن لغة C# من تطوير مايكروسوفت فإن عدد مستخدميها كبير، وهنا بعض الشركات الكبيرة Blizzard، Starbucks التي تستخدم هذه اللغة بشكل أساسي في معظم تطبيقاتها.

فبناءً على ميزات وخصائص لغة C# الآنفة الذكر من جهة، وكون أنه لدينا اطلاع ومعرفة عملية سابقة اكتسبناها خلال مسيرتنا الدراسية في الجامعة، وللربط من أجل تكامل العمل وإظهار النتائج على واجهة سهلة الاستخدام وبسيطة لتسهيل عمل موظفين شؤون الامتحانات، قررنا أن نعتمد هذه اللغة في تصميم وبناء هذا المشروع، حيث يتكون مشروعنا من واجهات وقسم برمجي.

**تقسيم البرنامج:**

تم تقسيم البرنامج إلى ثلاث طبقات لتنظيم وتسهيل العمل وهي:

* + 1. طبقة الاتصال DataAccessLayer: تحوي على تابع نص الاتصال الذي يتم عبره الاتصال بقاعدة البيانات، بالإضافة الى توابع فتح وإغلاق الاتصال وتابع تنفيذ الإجرائيات المخزنة في قاعدة البيانات وتابع استقبال البيانات المجدولة.
    2. طبقة الصفوف BusinessLayer ويحوي خمسة صفوف (Class) وهي:
* صف الدرجة Cls\_degree: يحوي على توابع إضافة وتعديل وحذف الدرجات بالإضافة لبعض التوابع الإضافية.
* صف الطالب Cls\_student: يحوي على توابع إضافة وتعديل وحذف بيانات الطالب بالإضافة لبعض التوابع الإضافية.
* صف تسجيل الدخول ClsLogin: يحوي على تابع تسجيل الدخول وتوابع النسخ الاحتياطي والاستعادة.
* صف المدرسين والمواد Cls\_TeacherandMaterial: يحوي على توابع إضافة وتعديل وحذف المواد والمدرسين.
* صف التشفير Cls\_encrypt: يحوي على توابع التشفير الخاصة بحماية كلمات المرور.
  + 1. طبقة الواجهات المرئية(الفيزيائية) PhysicalLayer:

**الواجهات**

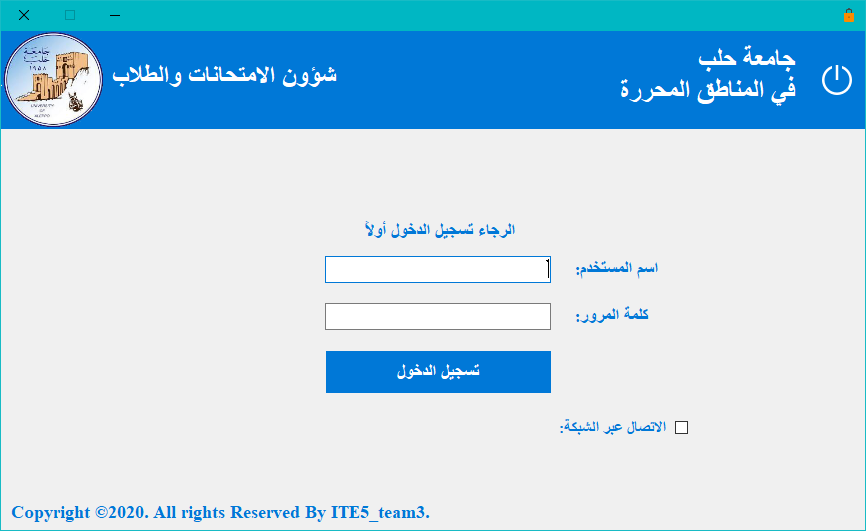
**الواجهة الأولى "واجهة تسجيل الدخول"**

من خلال هذه الواجهة يتم تسجيل الدخول إلى البرنامج للأشخاص المسموح لهم بذلك (أي الذين لديهم اسم مستخدم وكلمة مرور في قاعدة البيانات).

يوضح الشكل (4-1) شكل هذه الواجهة حيث تحتوي على أداتي مربع نص (TextBox) وتمثل الأولى اسم المستخدم الذي يقوم المستخدم بإدخاله، والثانية لإدخال كلمة المرور، وأداة زر (Button) لتسجيل الدخول عند النقر عليها بعد إدخال الاسم وكلمة المرور.

يمكن الاتصال عبر أكثر من حاسب إلى قاعدة البيانات عن طريق الشبكة المحلية LAN وذلك عبر الضغط على مربع "الاتصال عبر الشبكة" (CheckBox) ثم إدخال عنوان IP للسيرفر على الشبكة عبر مربع نص (TextBox).

**آلية العمل:** يتم إرسال اسم المستخدم وكلمة المرور من واجهة الإدخال (تسجيل الدخول) إلى صف تسجيل الدخول (Cls) الذي يحوي تابع الاستدعاء لاجرائية تسجيل الدخول (Sp\_Login) المخزنة ضمن قاعدة البيانات، ثم يعيد النتيجة بنجاح أو فشل تسجيل الدخول.

الشكل (4-1) تصميم واجهة تسجيل الدخول داخل بيئة الفيجوال استوديو.

بعد تسجيل الدخول يتم تحديد صلاحيات الوصول للبرنامج بحسب نوع المستخدم، يوجد ثلاث مستويات لنوع المستخدم:

1. المدير: وله صلاحيات كاملة للوصول لجميع أزرار واجرائيات البرنامج.
2. إدارة الطلاب: وله الصلاحية بالوصول للاجرائيات المتعلقة بشؤون الطلاب.
3. إدارة الامتحانات: وله الصلاحية بالوصول للاجرائيات المتعلقة بشؤون الامتحانات.

**الواجهة الثانية "الواجهة الرئيسية"**

وهي الواجهة التي تظهر للمستخدم بعد نجاح عملية تسجيل الدخول.

الشكل (4-2) تصميم الواجهة الرئيسية داخل بيئة الفيجوال استوديو.

يمثل الشكل (4-2) تصميم الواجهة الرئيسية والتي تحوي مجموعة من العناصر المرقمة على الشكل، وهي:

1. **الحاوية (Panel):** وهي أداة تستخدم لضم العناصر وإظهارها ضمن حدودها.
2. **زر إدخال الطلاب (Button):** عند النقر على هذا الزر تظهر أداة تسمى User control (سيتم شرحها لاحقاً) ضمن الحاوية تمثل واجهة إدخال الطلاب.
3. **زر إدخال الدرجات (Button):** عند النقر على هذا الزر تظهر User control ضمن الحاوية تمثل واجهة ادخال الدرجات.
4. **زر إدارة الدرجات (Button):** عند النقر على هذا الزر تظهر User control ضمن الحاوية تمثل واجهة إدارة الدرجات.
5. **زر إدارة المواد والمدرسين (Button):** عند النقر على هذا الزر تظهر User control ضمن الحاوية تمثل واجهة إدارة المواد والمدرسين.
6. **زر إنشاء نسخة احتياطية (Button):** عند النقر على هذا الزر تظهر User control ضمن الحاوية تمثل واجهة إنشاء نسخة احتياطية.
7. **زر استعادة نسخة احتياطية (Button):** عند النقر على هذا الزر تظهر User control ضمن الحاوية تمثل واجهة استعادة نسخة احتياطية.
8. **زر تسجيل الخروج (Button):** عند النقر على هذا الزر يتم تسجيل خروج المستخدم من البرنامج والرجوع إلى الواجهة الأولى "واجهة تسجيل الدخول".
9. **زر تصغير القائمة اليمينية (Button):** عند الضغط على هذا الزر يتم تصغير القائمة بشكل تدريجي نحو اليمين وذلك عن طريق أداة المؤقت (1Timer)، وعند الضغط على الزر مرة أخرى تعود القائمة إلى موضعها الأصلي بشكل تدريجي.
10. **أداة صورة (picture box):** يحتوي صورة شعار الجامعة.
11. **أداتي لافتة (Label):** حيث تظهر اللافتة الأولى اسم المستخدم الحقيقي، بينما تظهر اللافتة الثانية دور هذا المستخدم (مدير أو عادي).
12. **أداتي لافتة (Label):** تظهر الافتة الأولى الوقت الحالي لنظام التشغيل وذلك عن طريق ربطها بأداة مؤقت أخرى (Timer2)، واللافتة الثانية تظهر التاريخ الحالي لنظام التشغيل.
13. **زر الإغلاق (Button):** عند الضغط عليه يتم إغلاق البرنامج.

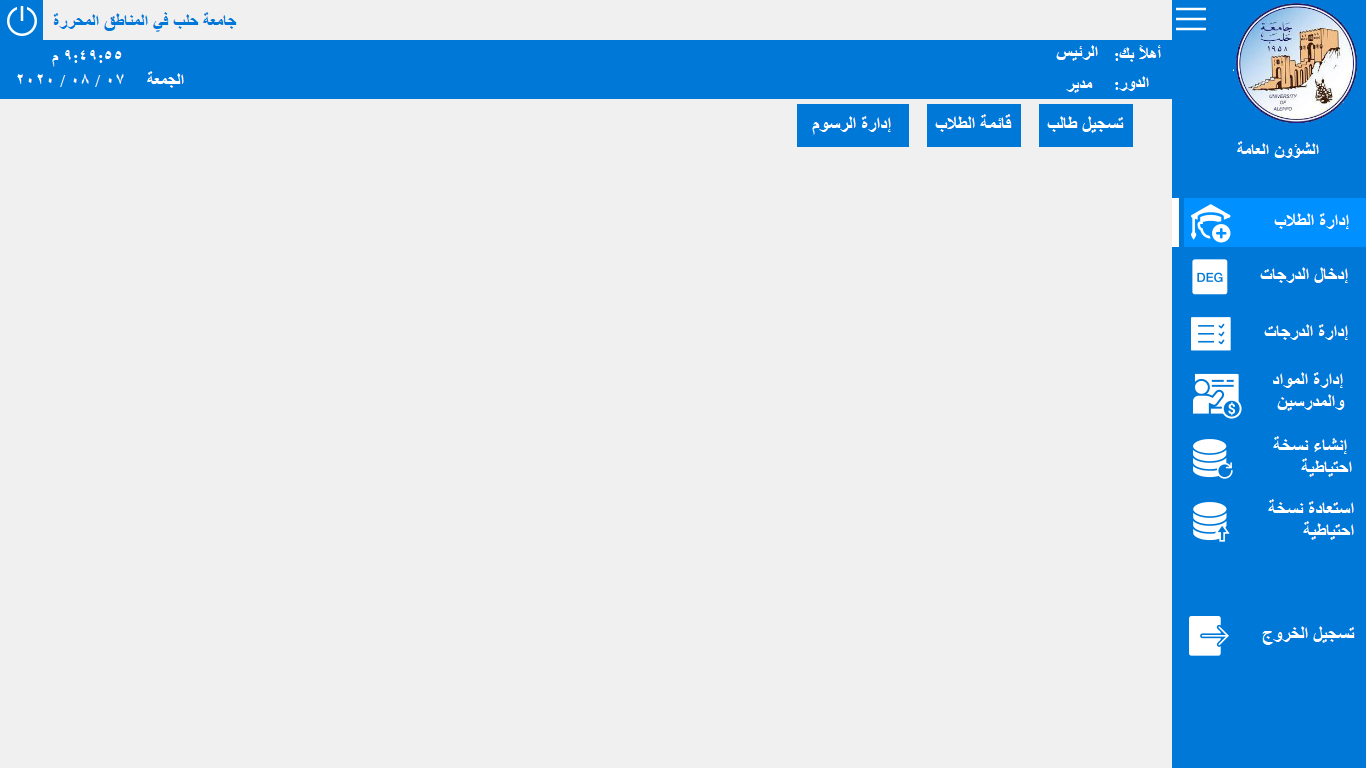
**User control**

هي عبارة عن مكوّن Component له تمثيل مرئي مثل (Button، Lable، …)، يمكن إنشاءه وإعادة استخدامه كأي أداة أخرى من أدوات الـ Visual Studio الموجودة ضمن الـ ToolBox.

وبعبارة أخرى هي أداة تستخدم لتجميع عدد من الأدوات ضمنها وإظهارها على شاشة التنفيذ وهي تشبه الفورم ولكن تختلف عنه بأنها ليست نافذة منبثقة كالفورم حيث يمكن وضعها على أي فورم أو أي أداة حاوية ضمن الفورم.

يتضمن المشروع ستة من User control حيث قمنا بتصميم كل واحدة على حدة لتنفيذ عمل محدد، وهي كالتالي:

**أولاً: إدارة الطلاب**

وهي مخصصة لإدارة شؤون الطلاب إذ تحتوي على ثلاثة أزرار من نوع أداة Button هي (تسجيل طالب & قائمة الطلاب & إدارة الرسوم) كما في الشكل (4-3)، وعند الضغط على أي زر فسوف تظهر نافذة خاصة به.

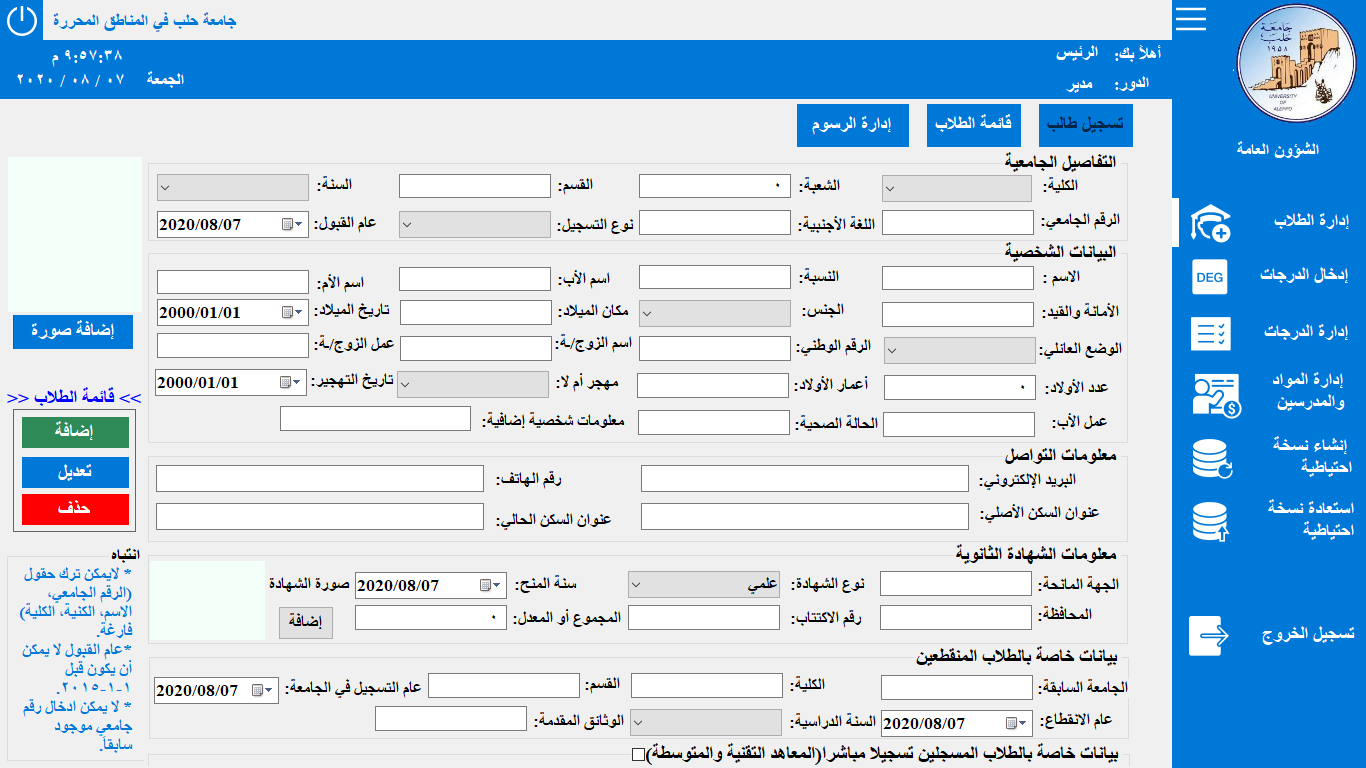
الشكل (4-3) تصميم الـ User Control الخاص بإدارة الطلاب.

النافذة الأولى (تسجيل طالب)

وهي نافذة خاصة بتسجيل بيانات طالب جديد أو تعديل بيانات طالب مُسجل مسبقاً أو حذف بيانات طالب عند اللزوم، وتظهر هذه النافذة عند الضغط على زر تسجيل طالب كما في الشكل (4-4) إذ تحتوي من الجهة اليمنى على ست مجموعات من grobebox وكل مجموعة تحتوي على بيانات محددة من البيانات الخاصة بالطالب.

المجموعة الأولى **(التفاصيل الجامعية)** وتحتوي على مايلي:

1. أداة مربع تحرير وسرد Combobox لاختيار الكلية التي سيسجل فيها الطالب.
2. أداة مربع نص TextBox لإدخال الشعبة في حال كان هناك أكثر من شعبة.
3. أداة مربع نص TextBox لإدخال القسم في حال كانت الكلية تحوي أكثر من قسم.
4. أداة مربع تحرير وسرد Combobox لاختيار السنة الجامعية.
5. أداة مربع نص TextBox لإدخال الرقم الجامعي للطالب.
6. أداة مربع نص TextBox لإدخال اللغة الأجنبية للطالب.
7. أداة مربع تحرير وسرد Combobox لاختيار نوع التسجيل (عام، موازي).
8. أداة التاريخ والوقت DateTimePicker لاختيار عام القبول في الجامعة.



الشكل (4-4) نافذة تسجيل طالب.

المجموعة الثانية **(البيانات الشخصية)** وتحتوي على ما يلي:

1. أداة مربع نص TextBox لإدخال اسم للطالب.
2. أداة مربع نص TextBox لإدخال كنية الطالب.
3. أداة مربع نص TextBox لإدخال اسم الأب للطالب.
4. أداة مربع نص TextBox لإدخال اسم الأم للطالب.
5. أداة مربع نص TextBox لإدخال الأمانة والقيد للطالب.
6. أداة مربع تحرير وسرد Combobox لاختيار جنس الطالب.
7. أداة مربع نص TextBox لإدخال مكان الميلاد للطالب.
8. أداة التاريخ والوقت DateTimePicker لإدخال تاريخ الميلاد للطالب.
9. أداة مربع تحرير وسرد Combobox لاختيار العائلي للطالب (أعزب، متزوج، .....).
10. أداة مربع نص TextBox لإدخال الرقم الوطني للطالب.
11. أداة مربع نص TextBox لإدخال اسم الزوج/ة في حال كان الطالب/ة متزوج/ة.
12. أداة مربع نص TextBox لإدخال عمل الزوج/ة.
13. أداة مربع نص TextBox لإدخال عدد الأولاد.
14. أداة مربع نص TextBox لإدخال أعمار الأولاد.
15. أداة مربع تحرير وسرد Combobox لاختيار هل الطالب مهجر أم لا.
16. أداة التاريخ والوقت DateTimePicker لإدخال تاريخ التهجير في حال كان مهجر.
17. أداة مربع نص TextBox لإدخال ما هو عمل الأب.
18. أداة مربع نص TextBox لإدخال الحالة الصحية للطالب.
19. أداة مربع نص TextBox لإدخال أي معلومات إضافية عن الطالب.

المجموعة الثالثة **(معلومات التواصل)** وتحتوي على ما يلي:

1. أداة مربع نص TextBox لإدخال البريد الإلكتروني للطالب.
2. أداة مربع نص TextBox لإدخال رقم هاتف الطالب.
3. أداة مربع نص TextBox لإدخال عنوان السكن الأصلي للطالب.
4. أداة مربع نص TextBox لإدخال عنوان السكن الحالي للطالب.

المجموعة الرابعة **(معلومات الشهادة الثانوية)** وتحتوي على ما يلي:

1. أداة مربع نص TextBox لإدخال من هي الجهة المانحة للشهادة الثانوية.
2. أداة مربع تحرير وسرد Combobox لاختيار نوع الشهادة الثانوية.
3. أداة التاريخ والوقت DateTimePicker لإدخال تاريخ منح الشهادة الثانوية.
4. أداة مربع نص TextBox لإدخال المحافظة التي تم الحصول على الشهادة الثانوية فيها.
5. أداة مربع نص TextBox لإدخال رقم الاكتتاب للشهادة الثانوية.
6. أداة مربع نص TextBox لإدخال مجموع أو معدل الشهادة الثانوية.
7. أداة button مع أداة pictureBox لإدخال صورة عن الشهادة الثانوية.

المجموعة الخامسة **(بيانات خاصة بالطلاب المنقطعين)** وتحتوي على ما يلي:

1. أداة مربع نص TextBox لإدخال الجامعة السابقة التي كان الطالب يدرس بها سابقاً.
2. أداة مربع نص TextBox لإدخال الكلية.
3. أداة مربع نص TextBox لإدخال الشعبة.
4. أداة التاريخ والوقت DateTimePicker لإدخال تاريخ التسجيل في الجامعة.
5. أداة التاريخ والوقت DateTimePicker لإدخال تاريخ الانقطاع عن الجامعة.
6. أداة مربع تحرير وسرد Combobox لاختيار السنة الدراسية التي وصل إليها الطالب.
7. أداة مربع نص TextBox لإدخال الوثائق المقدمة من الطالب.

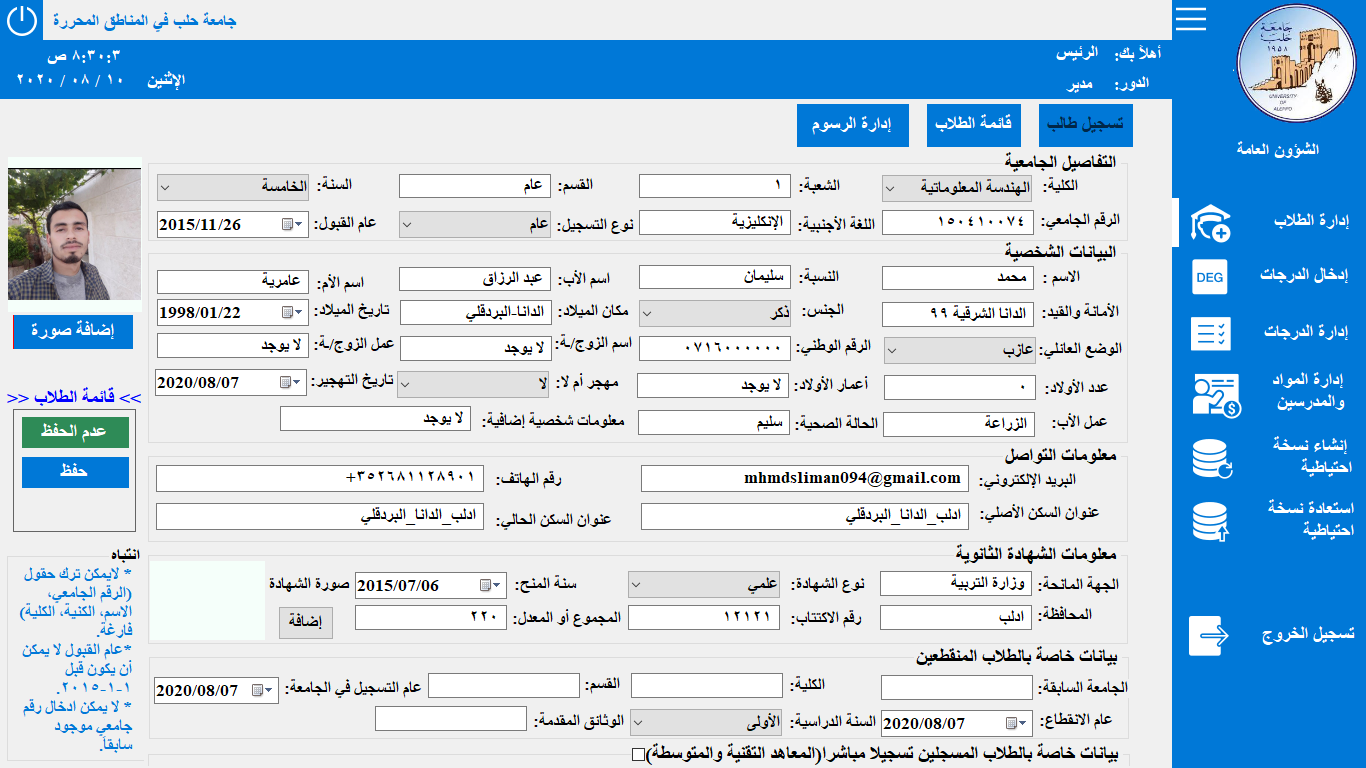
المجموعة السادسة **(بيانات خاصة بالطلاب المسجلين مباشراً والحائزين على شهادة المعاهد التقنية والمتوسطة)** ولا تظهر محتويات هذه المجموعة إلا بعد الضغط على مربع صغير من نوع checkbox موجود بجانب اسم المجموعة من الجهة اليسرى، وتحتوي هذه المجموعة على ما يلي:

1. أداة مربع نص TextBox لإدخال المعهد الذي تخرج الطالب به.
2. أداة مربع نص TextBox لإدخال القسم.
3. أداة التاريخ والوقت DateTimePicker لإدخال تاريخ التخرج.
4. أداة مربع تحرير وسرد Combobox لاختيار الدورة الفصلية التي تخرج الطالب بها.
5. أداة مربع نص TextBox لإدخال معدل التخرج.
6. أداة مربع نص TextBox لإدخال التقدير.

أما في الجهة اليسرى من نافذة تسجيل الطالب وفي القسم الأعلى يوجد أداة button مع أداة pictureBox لإدخال صورة شخصية للطالب، وفي المنتصف يوجد أداة panel تحتوي على ثلاثة أزرار هي (زر إضافة ياللون الأخضر & زر تعديل بالأزرق & زر حذف بالأحمر):

زر **الإضافة**: ووظيفة هذا الزر هي إضافة بيانات طالب جديد إلى قاعدة البيانات جيث يتم الضغط على الزر بعد تعبئة بيانات الطالب في الأدوات التي تم تبيانها في الأعلى وعند الضغط عليه يتم تنفيذ الإجرائية add\_student والتي بدورها تأخذ البيانات من الأدوات الموجودة في نافذة تسجيل الطالب وتضعها في قاعدة البيانات عن طريق الإجرائيتين المخزنتين ([add\_student] & [add\_student\_detile]).

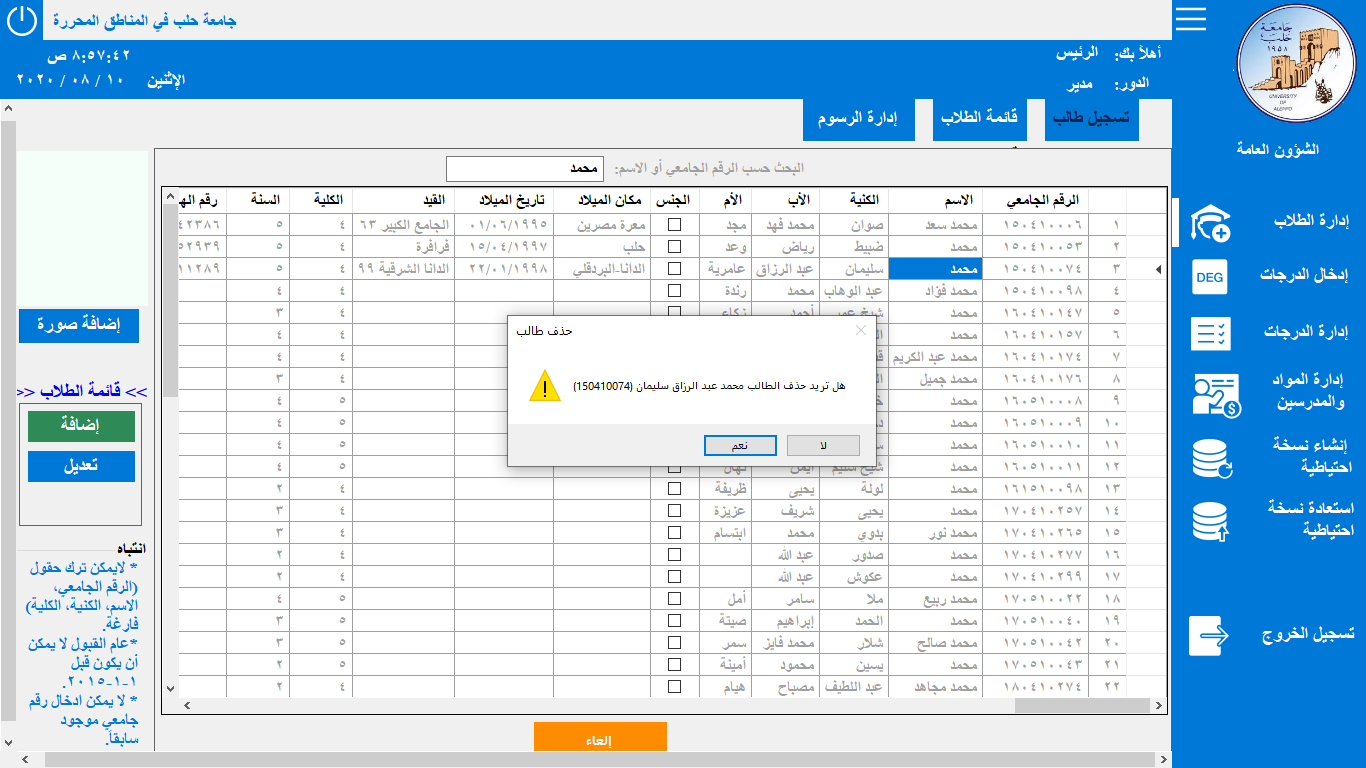
زر **تعديل** : لتعديل بيانات طالب موجود سابقاً في قاعدة البيانات، حيث عند النقر على هذا الزر يتم إظهار بيانات كافة الطلاب الذين تم إدخالهم في أداة DataGridView ضمن Panel مصغرة و مخفية موجودة بجانب زر التعديل حيث يتم جلب البيانات عن طريق الإجرائية المخزنة التي اسمها [Get\_All\_student] ، حيث عند الضغط على الزر يتم إظهار هذه الـ Panel مع تكبير حجمها، وتحتوي على أداة مربع نص TextBox للبحث عن الطالب الذي نريد تعديل بياناته ويكون البحث إما عن طريق الرقم الجامعي أو الاسم، وعند النقر على إحدى خلايا بيانات الطالب المراد تعديل بياناته يتم إخفاء الـ Panel وتعبئة البيانات الموجودة مسبقا ًللطالب الذي تم اختياره في الأدوات السابقة كما في الشكل (4-5):



الشكل (4-5) تعبئة البيانات آليا لأجل التعديل

بعد تعبئة البيانات تلقائياً كما في الشكل (4-5) نستطيع بعدها تعديل بيانات الطالب بكل سهولة وبساطة وعلى اليسار يوجد آداتين من نوع button ، الآداة الأولى هي زر **عدم الحفظ** باللون الأخضر ونضغط عليه إذا أردنا إلغاء العملية والتراجع عن التعديل فتعود واجه البرنامج كما كانت قبل الضغط على زر تعديل ولا يتم إجراء أي تغيرات على بيانات الطالب ، والآداة الثانية هي زر **حفظ** باللون الأزرق ونضغط عليه بعد تعديل البيانات التي نريد تعديلها فيتم حفظ البيانات الجديدة للطالب بعد التعديل ، حيث عند الضغط على زر حفظ تقوم الإجرائية البرمجية والتي اسمها edit\_student بنقل البيانات الجديدة من واجهة البرنامج إلى قاعدة البيانات وذلك عن طريق الإجرائية المخزنة في SQL والتي اسمها [edit\_student] .

زر **حذف** : لحذف طالب موجود سابقاً في قاعدة البيانات مع كافة بياناته ودرجاته، حيث عند النقر على هذا الزر يتم إظهار بيانات كافة الطلاب في أداة DataGridView ضمن الـ Panel السابقة نفسها عن طريق الإجرائية المخزنة [Get\_All\_student] حيث عند الضغط على الزر يتم إظهار هذه الـ Panel مع تكبير حجمها، وتحتوي على أداة مربع نص للبحث عن الطالب المُراد حذفه عن طريق الرقم الجامعي أو الاسم، وعند النقر المزدوج على إحدى خلايا بيانات الطالب المراد حذفه تظهر رسالة تحتوي اسم الطالب الثلاثي ورقمه الجامعي لتأكيد الحذف أو إلغاؤه كما في الشكل(4-6) فعند الإلغاء يتم إلغاء العملية أي تصغير الـ panel وإخفاءها ، أما عند الضغط على زر التأكيد فيتم حذف الطالب مع كافة بياناته ودرجاته ، حيث تقوم الإجرائية البرمجية والتي اسمها delet\_student بحذف بيانات الطالب من قاعدة البيانات وذلك عن طريق الإجرائية المخزنة في قاعدة البيانات [delete\_student] .



الشكل (4-6) ظهور رسالة لتأكيد عملية الحذف

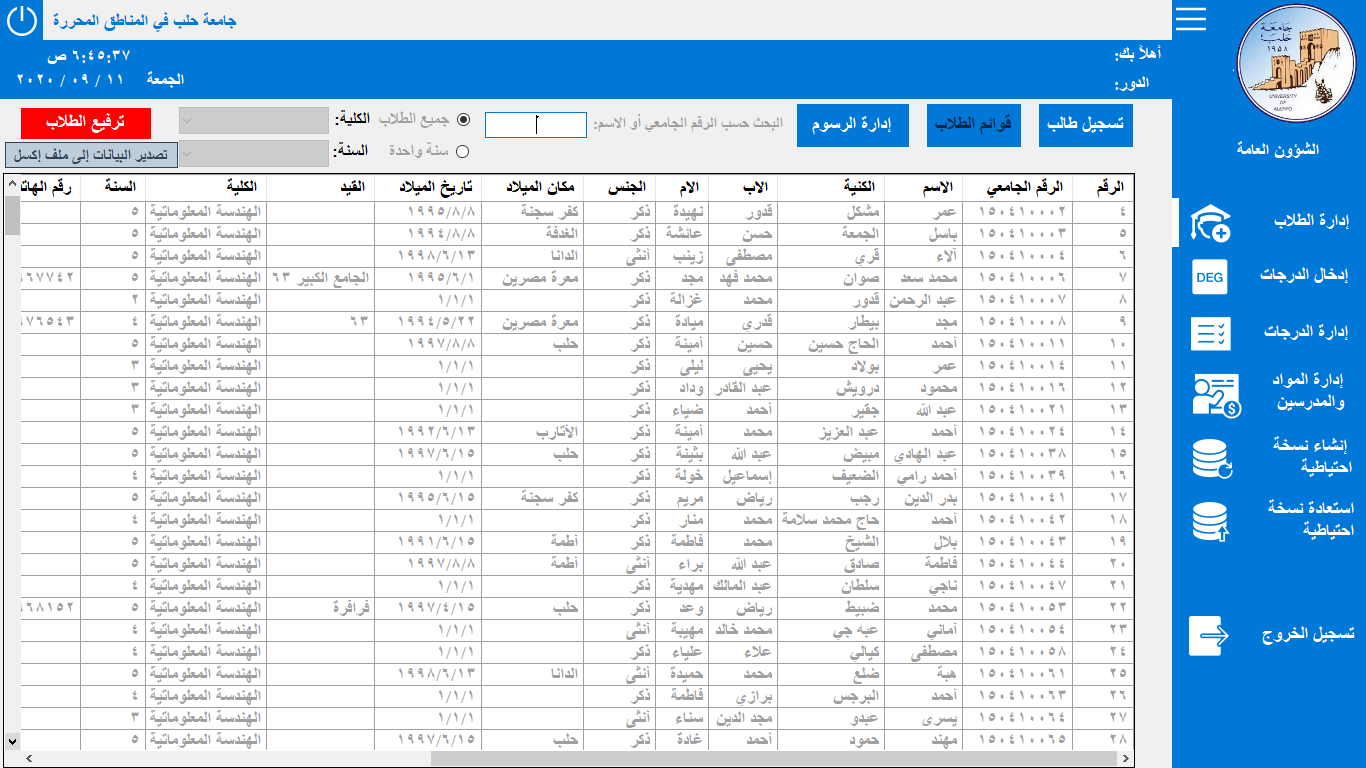
أما من الجهة اليسرى من نافذة تسجيل طالب وفي القسم الأسفل يوجد ملاحظة وهي عبارة عن أداة من نوع groupebox تحتوي بداخلها على أداة من نوع label والتي بدورها تحتوي على العبارة التالية:

* لايمكن ترك حقول (الرقم الجامعي، الاسم، الكنية، الكلية) فارغة.
* عام القبول لا يمكن أن يكون قبل 2015-1-1.
* لا يمكن ادخال رقم جامعي موجود سابقاً.

وهي ملاحظات وتنبيهات هامة عند إضافة بيانات طالب جديد لكي تتم عملية الإضافة بنجاح فمثلاً الجامعة تأسست في عام 2015 فلا يمكن أن يكون عام القبول قبل عام 2015.

النافذة الثانية **(قائمة الطلاب)**:

وهي نافذة خاصة للإطلاع على الطلاب وبياناتهم وتصدير قائمة بأسماء الطلاب إلى ملف إيكسل بالإضافة إلى ترفيعهم في نهاية العام الدراسي وتظهر هذه النافذة عند الضغط على زر قائمة الطلاب حيث تظهر قائمة فيها بيانات كافة الطلاب ضمن أداة DataGridView عن طريق الإجرائية المخزنة [Get\_All\_student] كما في الشكل (4-7):

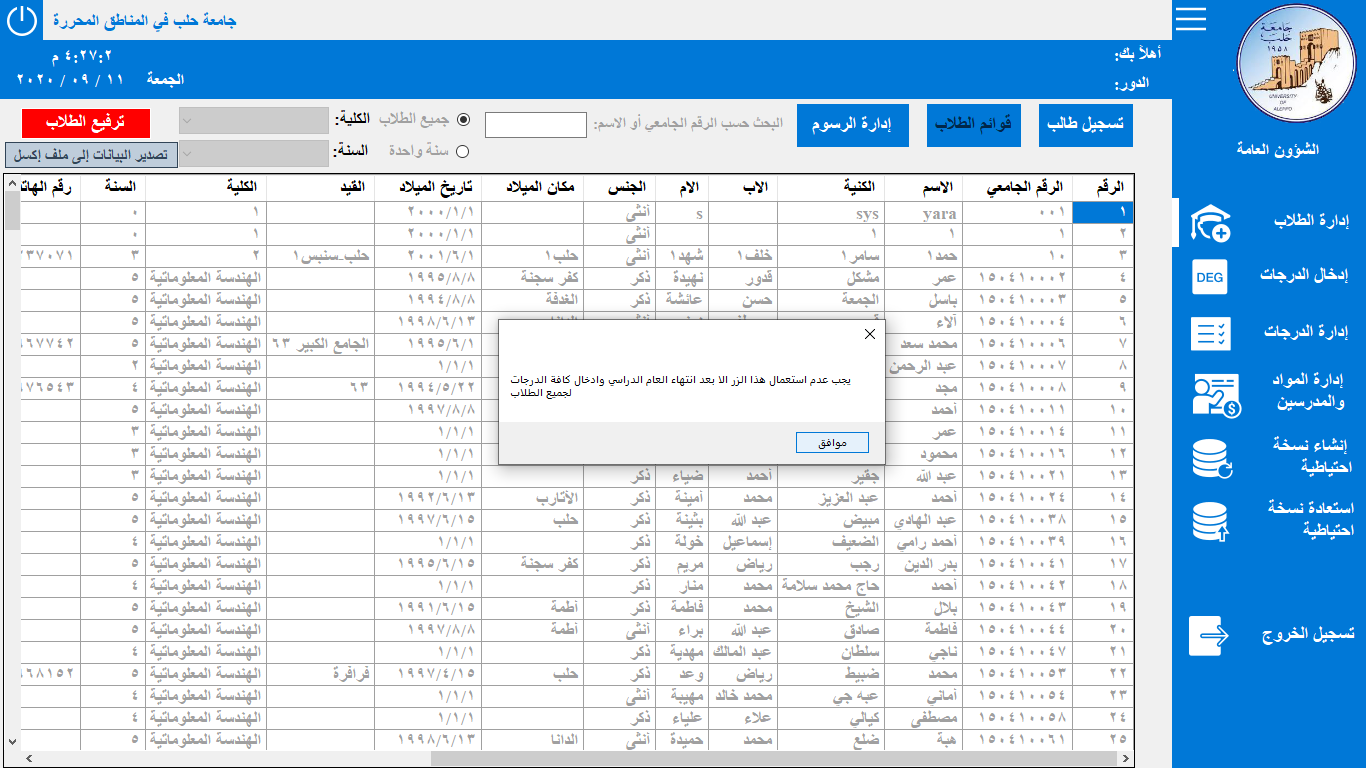


الشكل (4-7) قائمة ببيانات كافة الطلاب

كما يمكن أيضاً إظهار طلاب سنة واحدة فقط وذلك بتحديد الخيار ( سنة واحدة ) بدلاً من خيار (جميع الطلاب) ثم تحديد الكلية مثلا كلية الهندسة المعلوماتية ثم تحديد السنة مثلاً السنة الخامسة فتظهر لدينا قائمة بطلاب (كلية الهندسة المعلوماتية السنة الخامسة فقط) , وفي أعلى قائمة الطلاب يوجد أداة من نوع TextBox للبحث عن طالب معين حيث تقوم الإجرائية البرمجية التي اسمها Search\_Student\_name بالبحث عن طالب بواسطة الإجرائية المخزنة في قاعدة البيانات والتي اسمها [Search\_Student\_name] والتي تمنحنا حرية كبيرة في البحث ، حيث يمكننا البحث بواسطة الرقم الجامعي للطالب فهنا نتيجة البحث هي طالب واحد لأن الرقم الجامعي هو رقم وحيد وخاص بكل طالب والذي لا يتكرر مع أي طالب آخر يدرس في نفس الجامعة أو يمكننا البحث بواسطة الاسم الأول للطالب أو البحث بواسطة الاسم الأول للطالب مع اسم الأب أو البحث بواسطة الاسم الثلاثي للطالب أو البحث بواسطة الاسم الأول للطالب والكنية بغض النظر عن اسم الأب أو البحث بواسطة الكنية للطالب حيث عند كتابة الكنية في مربع البحث فسوف يظهر جميع الطلاب الذين لهم نفس الكنية , كما يوجد أيضاً زر من نوع Button اسمه (تصدير البيانات إلى ملف إيكسل) وعند الضغط على هذا الزر يتم تصدير بيانات الطلاب الموجودة في الـ DataGridView إلى ملف من نوع إيكسل مهما كانت بيانات الطلاب سواء كانت جميع الطلاب أو سواء كانت فقط طلاب سنة محددة.

ترفيع الطلاب:

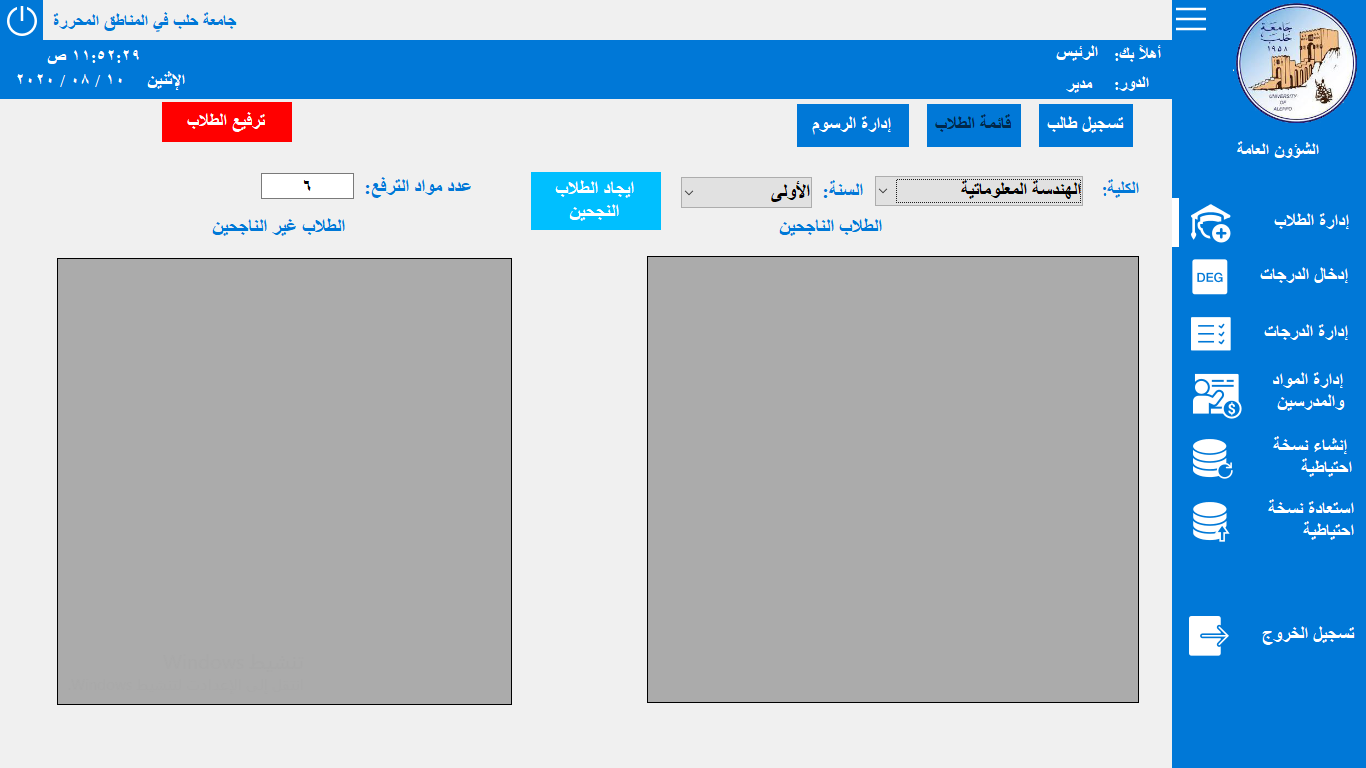
عند الضغط على زر ترفيع الطلاب ذو اللون الأحمر والموجود في نافذة قائمة الطلاب فتظهر رسالة تنبيهية للعملية محتوى الرسالة هو (يجب عدم استعمال هذا الزر إلا بعد انتهاء العام الدراسي وإدخال كافة الدرجات لجيع الطلاب) كما في الشكل (4-8):



الشكل (4-8) الرسالة التنبيهية الأولى لترفيع الطلاب

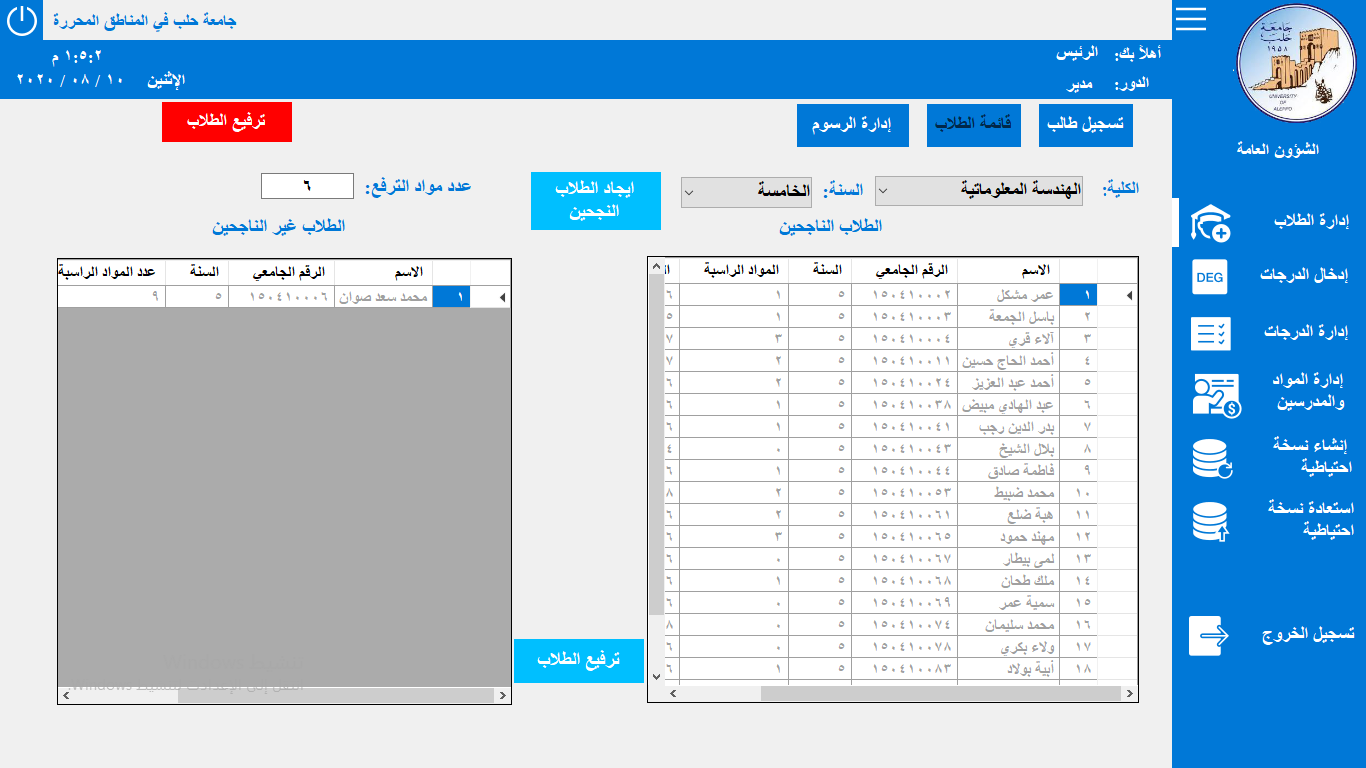
حيث تم وضع أكثر من رسالة تنبيهية وتأكيدية لحساسية الأمر، وبعد تأكيد الرسالة التنبيهة الأولى فسوف تظهر النافذة أو الواجهة الخاصة لعملية ترفيع الطلاب كما في الشكل (4-9) حيث محتوياتها هي:

1. أداة مربع تحرير وسرد Combobox لاختيار الكلية.
2. أداة مربع تحرير وسرد Combobox لاختيار السنة الدراسية (الأولى، الثانية، .....).
3. أداة TextBox لإدخال عدد مواد الترفع (الحد الأعلى لعدد المواد الراسبة لترفيع الطالب).
4. أداة DataGridView لعرض الطلاب الناجحين الذين سوف ينجحون أو يُنقلون للسنة التالية.
5. أداة DataGridView لعرض الطلاب الراسبين الذين عدد موادهم الراسبة أكبر من عدد مواد الترفع.
6. أداة button لإيجاد الطلاب الناجحين وكذلك الراسبين.



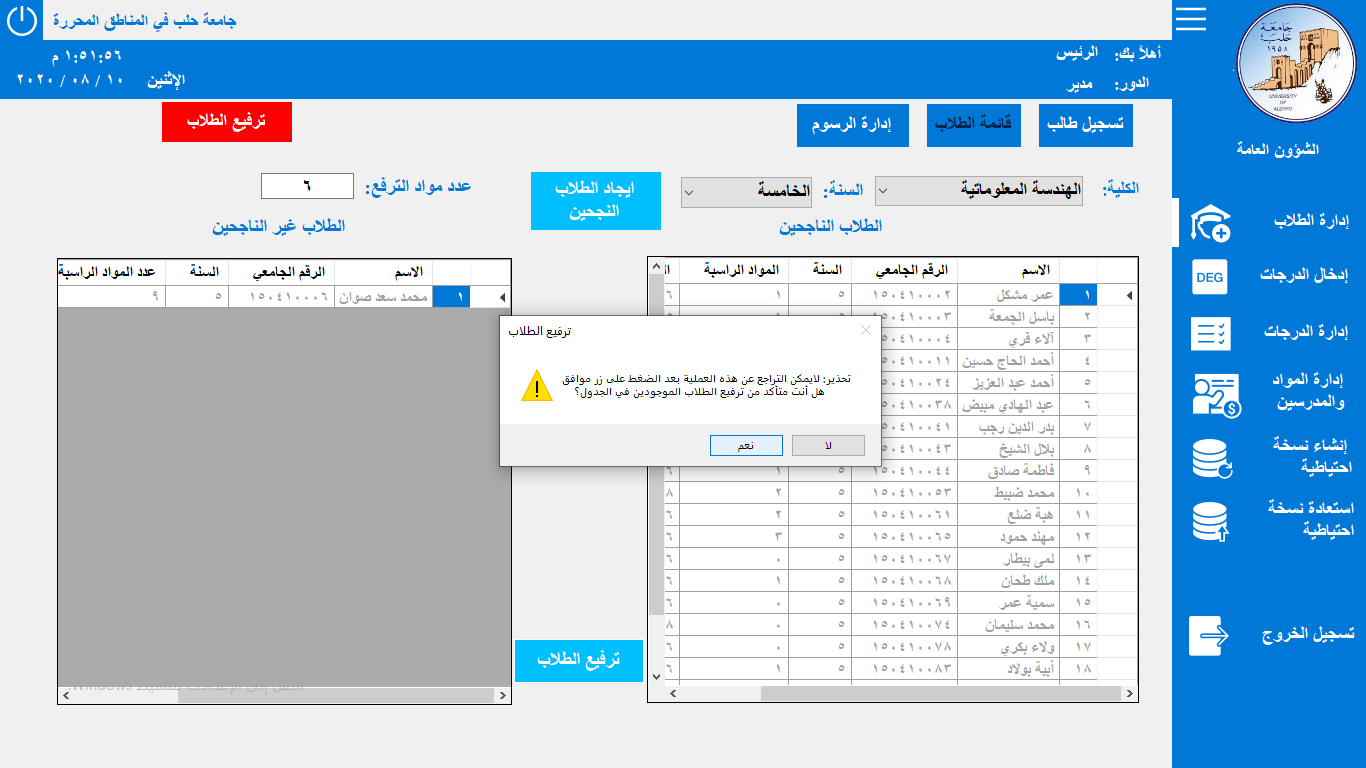
الشكل (4-9) واجهة ترفيع الطلاب

عملية ترفيع الطلاب تتم لكل كلية ولكل سنة على حِدى، فلأجل أن نقوم بعملية الترفيع نختار أولاً الكلية ثم نختار السنة الدراسية (أولى، ثانية، ثالثة ...) وبعدها نضع عدد مواد الترفع وهي أعلى عدد من المواد الراسبة التي يحملها الطالب ويكون ناجحاً (منقولاً) إلى السنة التالية، ثم نضغط على زر **إيجاد الطلاب الناجحين** باللون الأزرق الموجود في أعلى الواجهة فيحدث كما في الشكل (4-10):



الشكل (4-10) المعاينة الأولية للطلاب الناجحين والراسبين

إذ يتم جلب جميع طلاب الكلية المحددة والسنة المحددة ووضعهم في قائمتي الطلاب الناجحين والطلاب الراسبين فالطلاب الذين عدد موادهم الراسبة أصغر أو يساوي عدد مواد الترفع يتم وضعهم في قائمة الطلاب الناجحين عن طريق الإجرائية المخزنة والتي اسمها [success\_year] والطلاب الذين عدد موادهم الراسبة أكبر من عدد مواد الترفع يتم وضعهم في قائمة الطلاب الراسبين عن طريق الإجرائية المخزنة والتي اسمها [not\_success\_year] ، والغاية من ذلك هي المعاينة أولية للطلاب الذين سوف ينجحون قبل الضغط على زر الترفيع النهائي حيث يظهر هذا الزر **ترفيع الطلاب** بعد تعبئة قائمتي الطلاب الناجحين والطلاب الراسبين ، وهو أداة من نوع button باللون الأزرق موجود في أسفل واجهة ترفيع الطلاب وعند الضغط عليه تظهر رسالة تاكيدية تطلب تأكيد العملية ومحتوى الرسالة هو ( تحذير: لايمكن التراحع عن هذه العملية بعد الضغط على زر موافق، هل أنت متأكد من ترفيع الطلاب الموجودين في الجدول ؟) كما في الشكل (4-11):

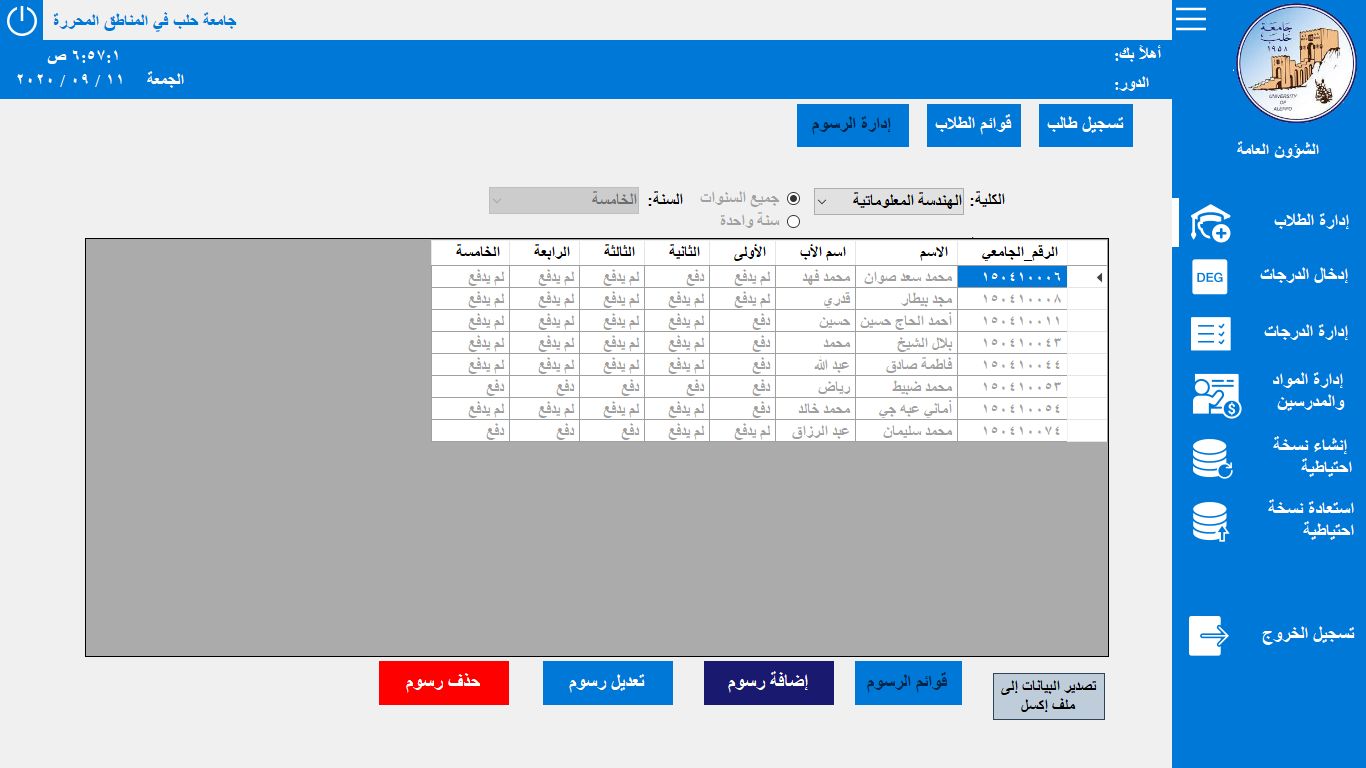


الشكل (4-11) الرسالة التأكيدية النهائية لترفيع الطلاب

عند عدم تأكيد الرسالة فسوف تُلغى عملية الترفيع أما عند تأكيد الرسالة واختيار نعم فسوف يتم ترفيع الطلاب الموجودين في جدول الطلاب الناجحين إلى السنة الدراسية الأعلى وذلك بواسطة الإجرائية المخزنة والتي اسمها [success\_year\_ok].

النافذة الثالثة (إدارة الرسوم):

وهي نافذة خاصة برسوم الطلاب للاستعلام عنها أو إضافتها أو تعديلها أو حذفها أو تصديرها إلى ملف إيكسل وتظهر هذه النافذة عند الضغط على زر **إدارة الرسوم** كما في الشكل (4-12) إذ يتم جلب بيانات كافة الطلاب لكلية محددة ووضعها في أداة DataGridView عن طريق الإجرائية المخزنة في قاعدة البيانات [get\_all\_fees] لتوضح أن الطالب دفع الرسوم أو لم يدفع وذلك لكل سنة دراسية على حِدى أي عرض قائمة بأسماء الطلاب وسنوات الدفع , كما يمكن أيضاً تحديد سنة واحدة فقط كالسنة الأولى أو الثانية أو الثالثة ..... :



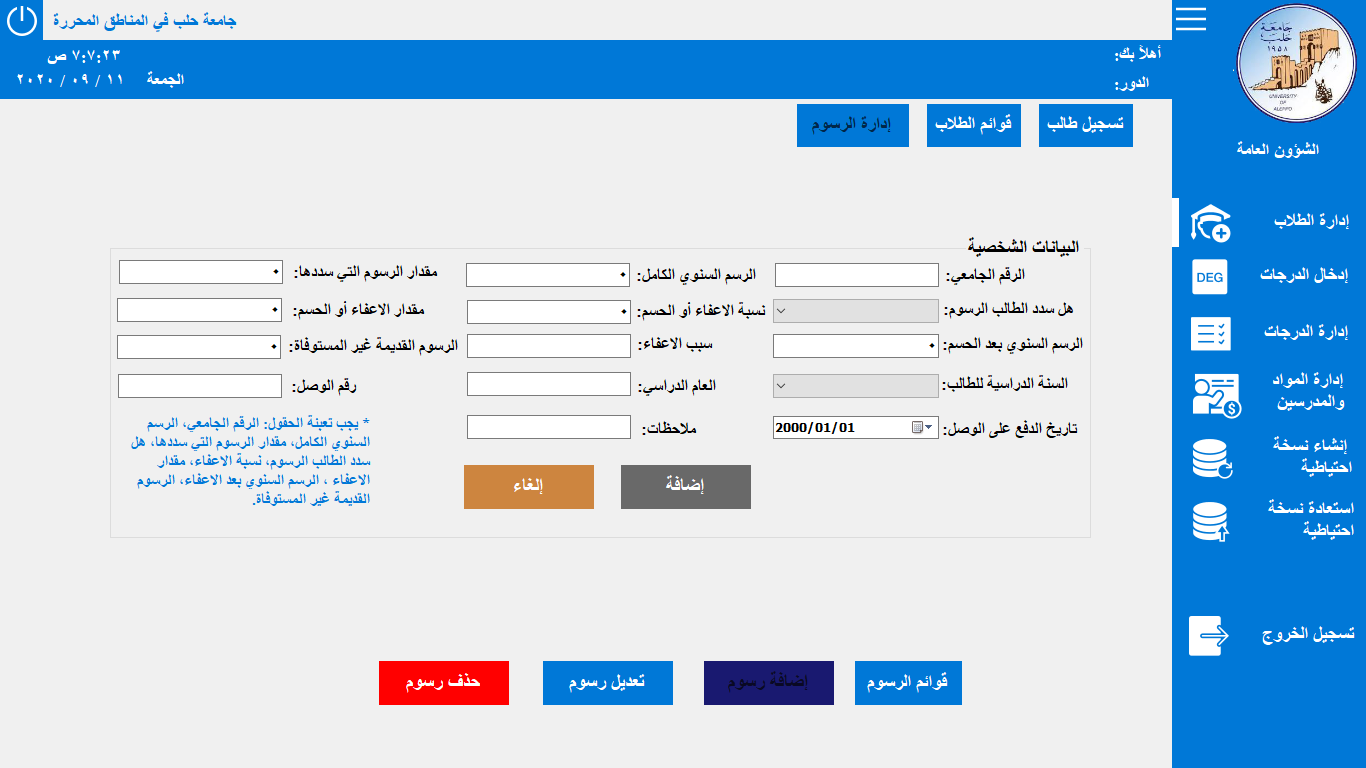
الشكل (4-12) نافذة إدارة الرسوم

تحتوي هذه النافذة على:

1. أداة DataGridView لعرضرسوم الطلاب فيها.
2. أداة button لإضافة رسم جديد.
3. أداة button لتعديل بيانات رسوم قديمة.
4. أداة button لحذف رسم محدد عند الحاجة لذلك.
5. أداة button لتصدير بيانات الرسوم إلى ملف إيكسل.
6. أداة button لعرض قوائم الرسوم.
7. أداة مربع تحرير وسرد Combobox لاختيار الكلية.
8. أداة مربع تحرير وسرد Combobox لاختيار السنة الدراسية.

عند الضغط على زر **إضافة رسوم** فسوف تظهر واجهة جديدة لتعبئة بيانات الرسم المُراد إدخاله كما في الشكل (4-13) حيث تحتوي على أداة grobebox والتي بدورها تحتوي على الأدوات التالية:

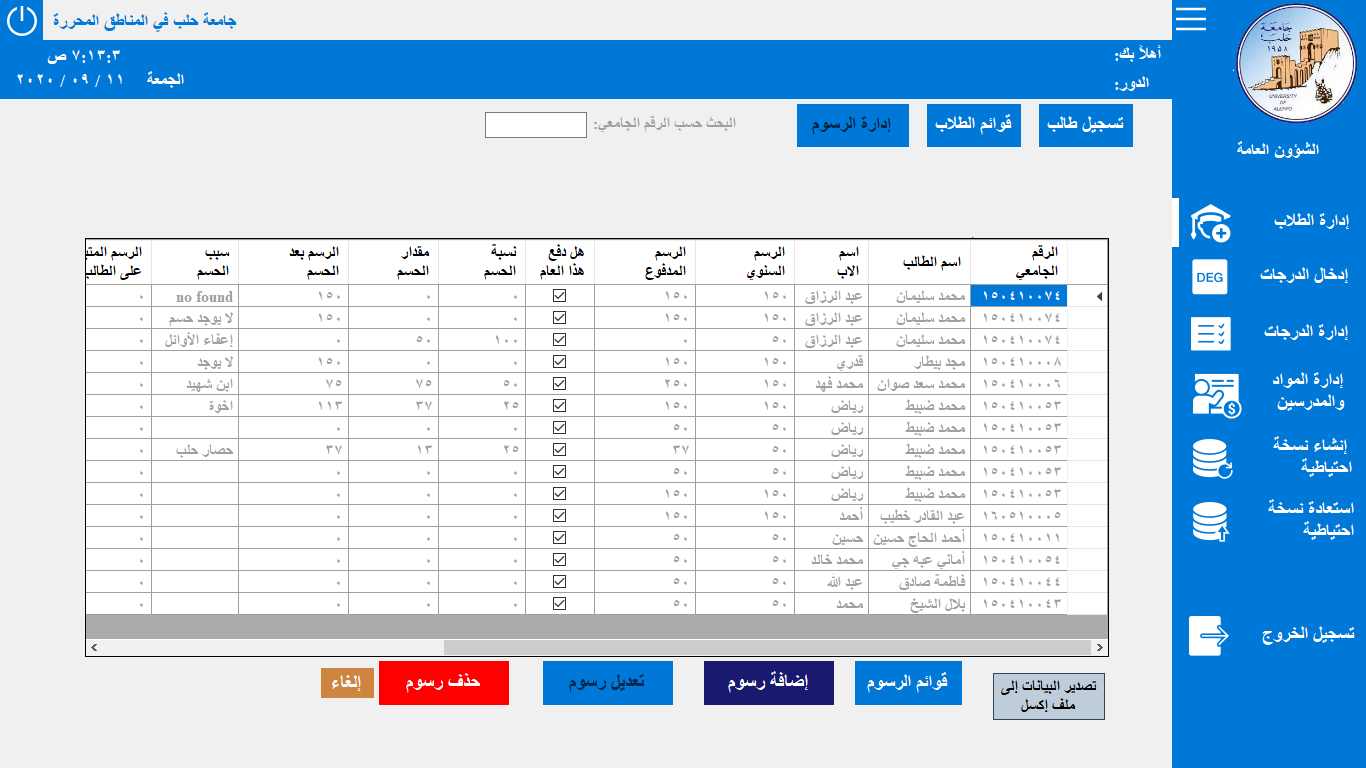
1. أداة مربع نص TextBox لإدخال الرقم الجامعي للطالب المراد إضافة رسم جديد له.
2. أداة مربع نص TextBox لإدخال الرسم السنوي الكامل له.
3. أداة مربع نص TextBox لإدخال مقدار الرسوم التي دفعها الطالب.
4. أداة مربع تحرير وسرد Combobox لاختيار هل سدد الطالب الرسوم أم لا.
5. أداة مربع نص TextBox لإدخال النسبة المئوية للحسم أو الإعفاء.
6. أداة مربع نص TextBox لإدخال مقدار الاعفاء أو الرسوم.
7. أداة مربع نص TextBox لإدخال الرسم بعد السنوي للحسم.
8. أداة مربع نص TextBox لإدخال سبب الإعفاء.
9. أداة مربع نص TextBox لإدخال الرسوم الغير مستوفاة.
10. أداة مربع تحرير وسرد Combobox لاختيار السنة الدراسية (أولى، ثانية .......).
11. أداة مربع نص TextBox لإدخال العام الدراسي.
12. أداة مربع نص TextBox لإدخال رقم وصل الرسوم.
13. أداة التاريخ والوقت DateTimePicker لإدخال تاريخ دفع الرسوم.
14. أداة مربع نص TextBox لإدخال أي ملاحظة نحتاجها.
15. أداة lable تحتوي على ملاحظات يجب التقيد بها عند إضافة رسم جديد.



الشكل (4-13) واجهة إضافة رسوم.

كما تحتوي على أداة button وهو زر **إلغاء** نستخدمه في حال أردنا التراجع عن عملية إضافة رسوم جديدة، وأداة ثانية button وهو زر **إضافة** ونضغط عليه بعد تعبئة بيانات الرسوم فتتم عملية إضافة رسم جديد وتخزينه في قاعدة البيانات بواسطة الإجرائية المخزنة [add\_fees].

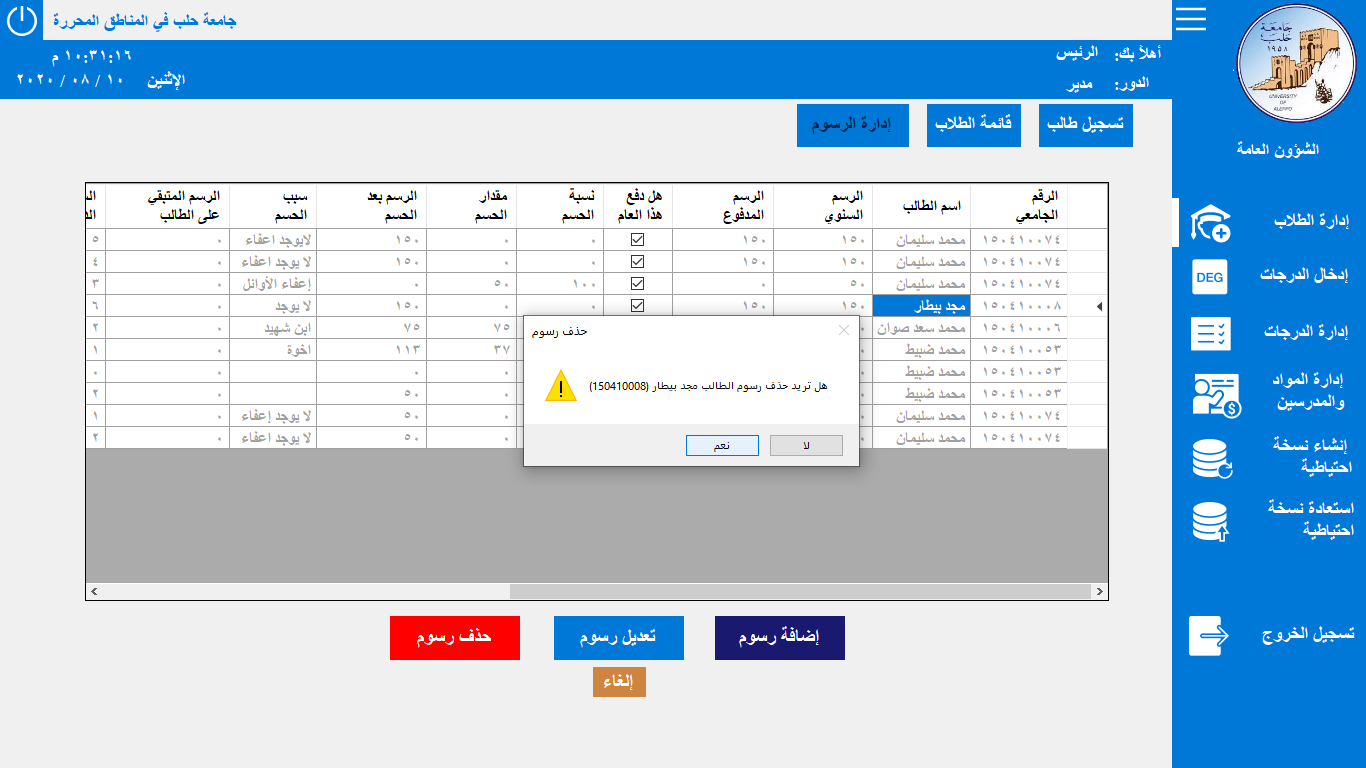
العملية الثانية في نافذة إدارة الرسوم وهي عملية تعديل رسوم طالب مُدخلة مُسبقاً وتكون عند الضغط على زر **تعديل رسوم** حيث عند النقر عليه يتم إظهار بيانات رسوم كافة الطلاب الذين تم إدخالهم في أداة DataGridView ضمن Panel مصغرة و مخفية موجودة بجانب زر التعديل حيث يتم جلب البيانات عن طريق الإجرائية المخزنة التي اسمها [get\_all\_fees] كما في الشكل(4-14)، حيث عند الضغط على الزر يتم إظهار هذه الـ Panel مع تكبير حجمها، وتحتوي على أداة مربع نص TextBox للبحث عن الطالب الذي نريد تعديل رسومه ، وعند النقر المزدوج على إحدى خلايا بيانات رسوم الطالب المراد تعديل بيانات رسومه يتم إخفاء الـ Panel وتعبئة البيانات تلقائياً للطالب الذي تم اختياره في الأدوات والحقول البرمجية في نافذة تشابه نافذة إضافة الرسوم , فعندها يمكن تعديلها بكل بساطة وسهولة:



الشكل (4-14) واجهة قائمة الرسوم من أجل التعديل.

فبعد تعديل بيانات الرسوم يوجد لدينا زرين الأول زر **إلغاء** للتراجع عن عملية تعديل بيانات الرسوم الجامعية ولا يتم تغيير أي شيء، والزر الثاني هو زر **حفظ** ويستخدم لتأكيد عملية التعديل حيث يتم حفظ البيانات الجديدة إلى قاعدة البيانات بواسطة الإجرائية المخزنة [edit\_fess].

العملية الثالثة في نافذة إدارة الرسوم هي عملية حذف رسوم سنوية لطالب ما وتتم العملية عند الضغط على زر **حذف رسوم** في نافذة إدارة الرسوم حيث عند النقر عليه يتم إظهار بيانات رسوم كافة الطلاب الذين تم إدخالهم في أداة DataGridView ضمن Panel مصغرة و مخفية موجودة بجانب زر الحذف حيث يتم جلب البيانات عن طريق الإجرائية المخزنة التي اسمها [get\_all\_fees]، حيث عند الضغط على الزر يتم إظهار هذه الـ Panel مع تكبير حجمها، وتحتوي على أداة مربع نص TextBox للبحث عن الطالب الذي نريد حذف رسومه في سنة جامعية محددة ، وعند النقر المزدوج على إحدى خلايا بيانات رسوم الطالب المراد حذف بيانات رسومه يتم إخفاء الـ Panel وظهور رسالة تأكيدية تطلب ـتأكيد عملية الحذف كما في الشكل (4-15):



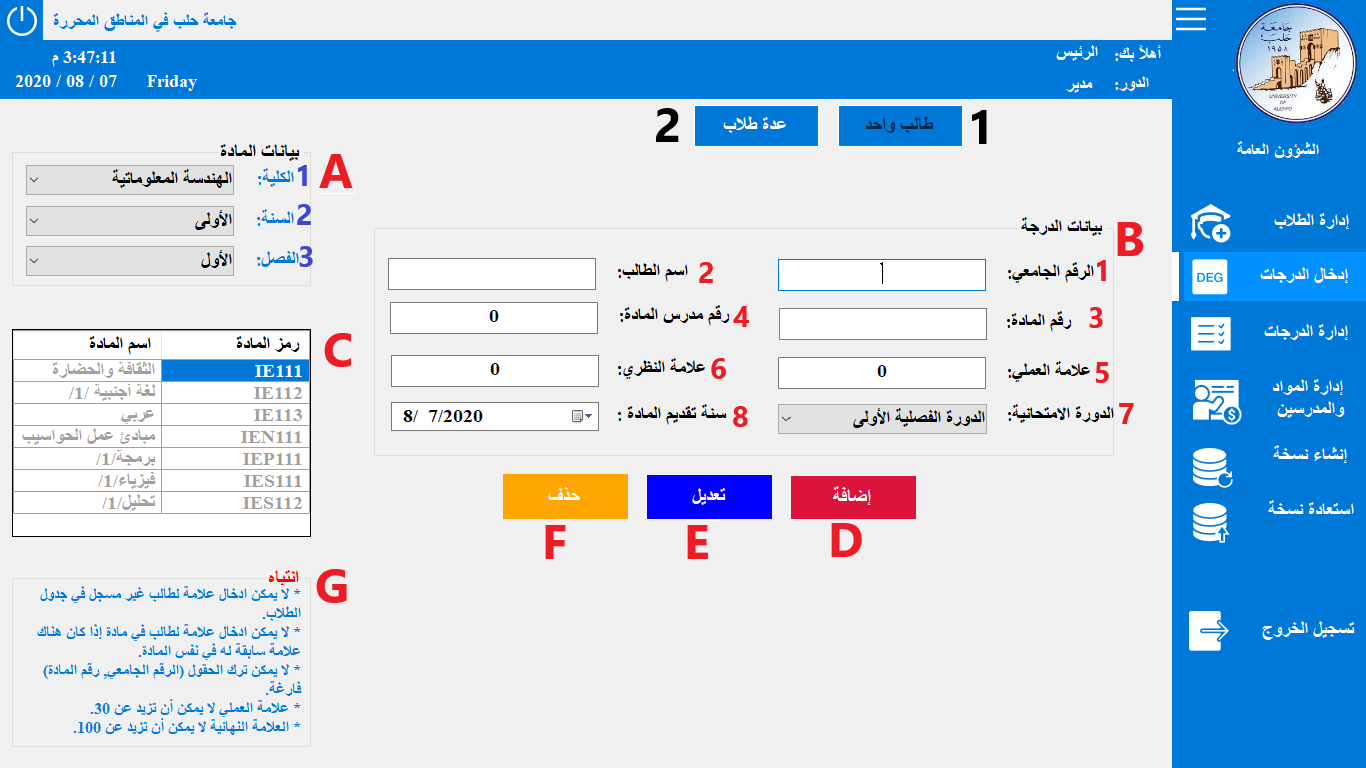
الشكل (4-15) ظهور رسالة تأكيد لحذف رسوم طالب.

بعد ظهور الرسالة التأكيدية إذا أردنا التراجع عن عملية الحذف فنضغط على زر (**لا**) ضمن الرسالة، أما إذا كنا نريد حذف الرسوم المحددة فنضغط على زر (**نعم**) وعندها يتم حذف الرسوم المححد من قاعدة البيانات بواسطة الإجرائية المخزنة [delete\_fees].

**ثانياً: إدخال الدرجات**

**طالب واحد:**

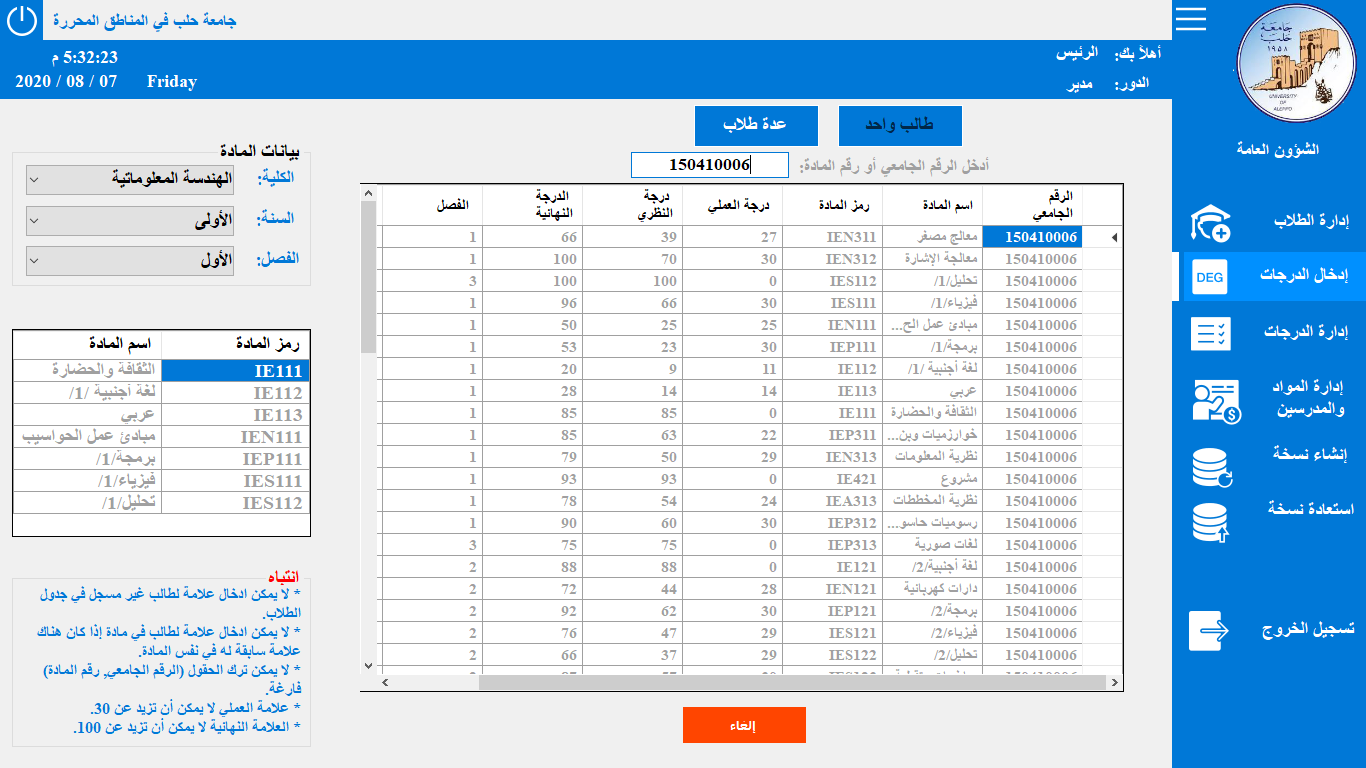
يتم من خلال هذه الواجهة إدخال معلومات الدرجة لطالب واحد أو لعدة طلاب معاً إلى جدول الدرجات في قاعدة البيانات أو تعديل بيانات درجة معينة أو حذف درجة مع بياناتها.

تحوي الواجهة على أداتي Button وأداتي groupbox (وهي أداة تستخدم لتجميع العناصر ضمنها بحيث تصبح هذه الأدوات كقالب واحد)، حيث يبين الشكل (4-16) تصميم الواجهة والأدوات التي تحتويها مرقمة على الشكل كالتالي:

الشكل (4-16) تصميم الـ User Control الخاص بإدخال الدرجات لطالب واحد.

1) أداة Button عند النقر عليها تقوم بعرض الواجهة الخاصة بإدخال الدرجة وتعديلها وحذفها لطالب واحد، وتضم هذه الواجهة الأدوات التالية:

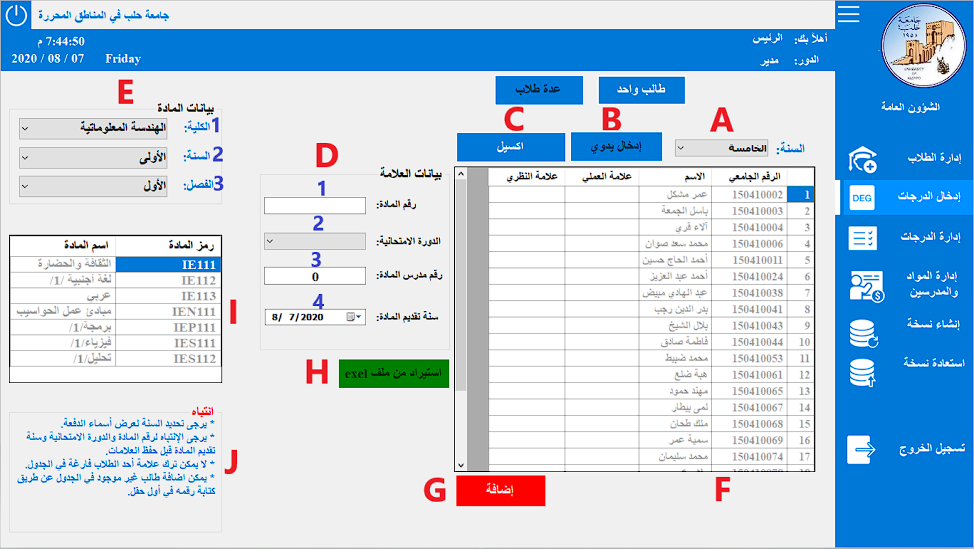
1. أداة groupbox الأولى "الخاصة ببيانات المادة" وتحتوي:
2. أداة مربع تحرير وسرد Combobox لاختيار الكلية.
3. أداة مربع تحرير وسرد Combobox لاختيار السنة.
4. أداة مربع تحرير وسرد Combobox لاختيار الفصل الدراسي.
5. أداة groupbox الثانية "الخاصة ببيانات الدرجة" وتحتوي:
6. أداة مربع نص TextBox لإدخال الرقم الجامعي للطالب.
7. أداة مربع نص TextBox لإدخال اسم للطالب.
8. أداة مربع نص TextBox لإدخال رقم المادة.
9. أداة مربع نص TextBox لإدخال رقم مدرس المادة.
10. أداة مربع نص TextBox لإدخال الدرجة العملية.
11. أداة مربع نص TextBox لإدخال الدرجة النظرية.
12. أداة مربع تحرير وسرد Combobox لاختيار الدورة الامتحانية.
13. أداة التاريخ والوقت DateTimePicker لاختيار عام تقديم المادة.
14. أداة dataGridView (تستخدم لعرض بيانات مجدولة مع أنواع مختلفة من مصادر البيانات وهي مزودة بالعديد من الخصائص والأحداث لإمكانية تخصيص طريقة عرض البيانات) لعرض اسم المادة والرقم المقابل لها.
15. زر Button لإضافة الدرجة والبيانات الخاصة بها إلى جدول الدرجات في قاعدة البيانات.
16. زر Button لتعديل بيانات الدرجة المدخلة، حيث عند النقر على هذا الزر يتم إظهار كافة الدرجات وبياناتها لكافة الطلاب الذي تم إدخالهم وذلك عن طريق Panel مصغرة و مخفية موجودة بجانب زر التعديل، حيث عند الضغط على الزر يتم إظهار هذه الـ Panel مع تكبير حجمها، وتحتوي على أداة مربع نص للبحث عن الدرجة عن طريق الرقم الجامعي كما في الشكل (4-17) أو رقم المادة، وأداة DataGridView يتم وضع البيانات بها و إظهارها وعند النقر على إحدى خلايا بيانات الدرجة المراد تعديل بياناتها يتم إخفاء الـ Panel وتعبئة بيانات الدرجة الذي تم اختيارها في الأدوات السابقة، ويوجد زر إلغاء Button لإلغاء العملية أي يتم تصغير الـPanel و إخفاءها.



الشكل (4-17) البحث عن درجات طالب عن طريق رقمه الجامعي.

1. زر Button لحذف درجة موجودة سابقاً في قاعدة البيانات (في جدول الدرجات) مع كافة بياناتها، حيث عند النقر على هذا الزر يتم إظهار بيانات كافة الدرجات في أداة DataGridView ضمن الـ Panel السابقة نفسها حيث عند الضغط على الزر يتم إظهار هذه الـ Panel مع تكبير حجمها، وتحتوي على أداة مربع نص للبحث عن الدرجة عن طريق الرقم الجامعي أو رقم المادة، عند النقر المزدوج على إحدى خلايا بيانات الدرجة المراد حذفها تظهر رسالة لتأكيد الحذف أو إلغاؤه فعند التأكيد يتم الحذف، أما عند الإلغاء يتم إلغاء العملية أي تصغير الـ Panel وإخفاءها.
2. أداة groupbox تحوي ضمنها أداة مربع نص TextBox لعرض بعض الملاحظات الهامة للمستخدم.

**عدة طلاب:**

2) أداة Button عند النقر عليها تقوم بعرض الواجهة الخاصة بإدخال الدرجة وتعديلها وحذفها لعدة طلاب، وتضم هذه الواجهة عدة أدوات مرقمة كما في الشكل (4-18):

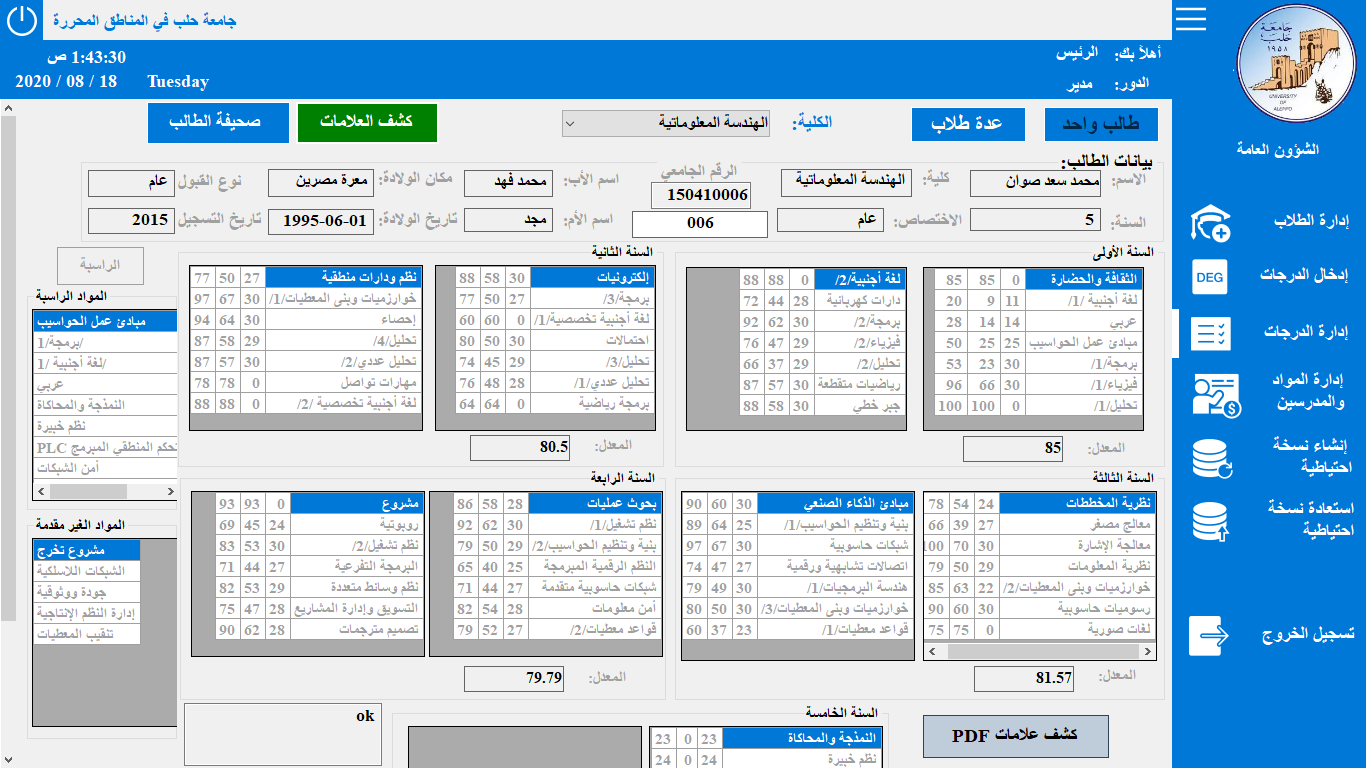
الشكل (4-18) تصميم الـ User Control الخاص بإدخال الدرجات لعدة طلاب.

1. أداة مربع تحرير وسرد Combobox لاختيار السنة.
2. أداة button لإدخال درجات عدة طلاب يدوياً.
3. أداة button لتفعيل عملية استيراد الدرجات من ملف إكسل.
4. أداة groupbox خاصة ببيانات الدرجة وتحتوي:
5. أداة مربع نص TextBox لإدخال رقم المادة.
6. أداة مربع تحرير وسرد Combobox لاختيار الدورة الامتحانية.
7. أداة مربع نص TextBox لإدخال رقم مدرس المادة.
8. أداة التاريخ والوقت DateTimePicker لاختيار عام تقديم المادة.
9. أداة groupbox خاصة ببيانات المادة وتحتوي:
10. أداة مربع تحرير وسرد Combobox لاختيار الكلية.
11. أداة مربع تحرير وسرد Combobox لاختيار السنة.
12. أداة مربع تحرير وسرد Combobox لاختيار الفصل الدراسي.
13. أداة datagridview لإدخال الدرجات لعدة طلاب إما يدوياً أو باستيرادها من ملف إكسل.
14. أداة button لإضافة الدرجات إلى جدول الدرجات بعد كتابتها يدوياً أو استيرادها.
15. أداة button لاستيراد الدرجات من ملف إكسل.
16. أداة dataGridView (تستخدم لعرض بيانات مجدولة مع أنواع مختلفة من مصادر البيانات وهي مزودة بالعديد من الخصائص والأحداث لإمكانية تخصيص طريقة عرض البيانات) لعرض اسم المادة والرقم المقابل لها.
17. أداة groupbox تحوي ضمنها أداة مربع نص TextBox لعرض بعض الملاحظات الهامة للمستخدم.

**آلية العمل:** عند النقر على زر "طالب واحد" تظهر واجهة إدخال الدرجة لطالب واحد، تتم عملية الإضافة بإدخال اسم الطالب أو رقمه ودرجة النظري والعملي له ثم النقر على زر إضافة، أما عند النقر على زر "عدة طلاب" تظهر واجهة إدخال الدرجة لعدة طلاب ويصبح زر الإدخال اليدوي مفعلّاً في البداية، وتتم عملية الإضافة هنا باختيار السنة الدراسية ليتم جلب أسماء وأرقام جميع طلاب هذه السنة تلقائياً، ثم يتم إدخال درجات الطلاب يدوياً وبيانات المادة المراد إدخال درجاتها، وبالنقر على زر إضافة يتم إضافة الدرجات إلى قاعدة البيانات، أما إذا تم النقر على زر "اكسيل" يتفعل زر "استيراد من ملف excel" وبالنقر عليه تظهر نافذة استعراض الملفات لاختيار ملف اكسل يحوي درجات مادة ما، بعد ذلك يتم اختيار بيانات المادة المراد إدخال درجاتها، وبالنقر على زر إضافة تتم عملية الإضافة.

**ثالثاً: إدارة الدرجات**

هذه الواجهة مخصصة لإدارة الدرجات التي يتم إدخالها بواسطة الواجهة السابقة (إدخال الدرجات)، وهي تنقسم إلى قسمين، القسم الأول لإدارة الدرجات لطالب واحد، والقسم الثاني لإدارة الدرجات لعدة طلاب، ويبين الشكل (4-19) القسم الأول من هذه الواجهة:



الشكل (4-19) إدارة الدرجات لطالب واحد (كشف العلامات).

وبدوره يتفرع هذا القسم أيضاً إلى قسمين وهما "كشف العلامات" و"صحيفة الطالب"، حيث يوجد زر لكل قسم بمجرد الضغط عليه تظهر الواجهة الخاصة به.

كشف العلامات:

من خلال هذه الواجهة يتم إصدار كشف درجات للطالب ذو الرقم الجامعي المُدخل وذلك عن طريق جلب جميع درجات هذا الطالب وبياناته من قاعدة البيانات (من جدول الطلاب وجدول الدرجات).

1. أداة مربع تحرير وسرد Combobox لاختيار كليّة الطالب.
2. أداة groupbox الأولى (الخاصة ببيانات الطالب):

يوجد ضمنها أداة مربع نص TextBox لإدخال الرقم الجامعي، حيث عند إدخال الرقم الجامعي (إما كاملاً أو بإدخال آخر ثلاثة رموز من الرقم) يتم ملأ باقي الأدوات ضمن الـ Groupboxبيانات الطالب الموافق للرقم الجامعي المدخل (البيانات الشخصية وبيانات التسجيل).

1. أداة groupbox الثانية (الخاصة بدرجات السنة الأولى) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول، والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

1. أداة groupbox الثالثة (الخاصة بدرجات السنة الثانية) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول، والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

1. أداة groupbox الرابعة (الخاصة بالمواد الراسبة) وتحتوي:

أداة DataGridView لإظهار أسماء المواد الراسبة للطالب ذو الرقم الجامعي المُدخل.

1. أداة groupbox الخامسة (الخاصة بدرجات السنة الثالثة) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

1. أداة groupbox السادسة (الخاصة بدرجات السنة الرابعة) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول، والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

1. أداة groupbox السابعة (الخاصة بالمواد غير المقدمة) وتحتوي:

أداة DataGridView لإظهار أسماء المواد التي لم يقدمها الطالب ذو الرقم الجامعي المُدخل.

1. أداة groupbox الثامنة (الخاصة بدرجات السنة الخامسة) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

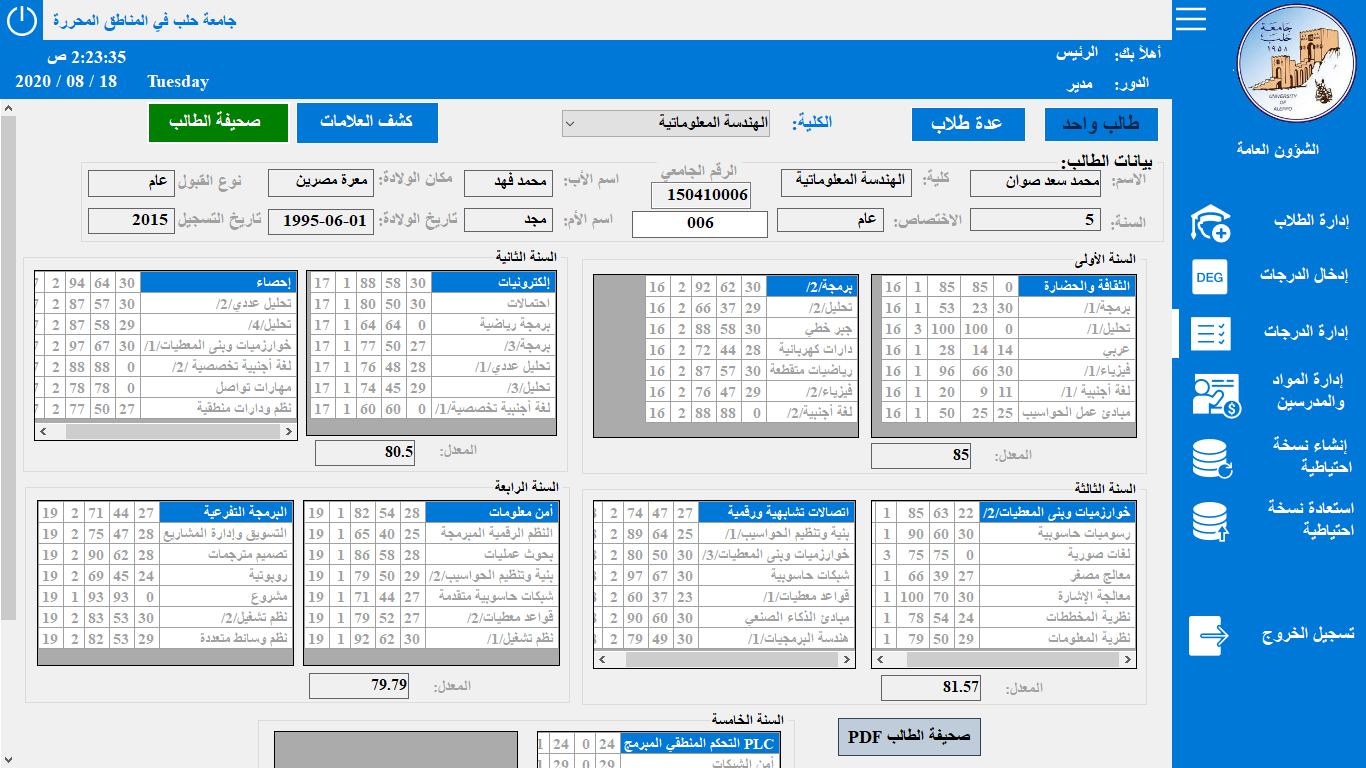
10) زر Button " كشف علامات PDF، عند الضغط على هذا الزر يتم تصدير كشف الدرجات

للطالب بصيغة PDF وحفظه في المجلد "أرشيف كشف العلامات" في المستندات.

1. أداة DataGridView وهذه الأداة تكون مخفية قمنا باستخدامها من أجل جلب بيانات الطالب إليها ثم تحويل البيانات إلى الأدوات في الـ groupbox الأولى، وذلك لأنه لا يمكن استيراد بيانات الطالب من جدول الطلاب إلى الأدوات داخل الـ groupbox بشكل مباشر.

صحيفة الطالب:

من خلال هذه الواجهة يتم إصدار صحيفة الطالب ذو الرقم الجامعي المُدخل وذلك عن طريق جلب جميع درجات هذا الطالب وبياناته من قاعدة البيانات (من جدول الطلاب وجدول الدرجات)، ويبين الشكل (4-20) هذه الواجهة:



الشكل (4-20) إدارة الدرجات لطالب واحد (صحيفة الطالب).

تتكون الواجهة السابقة من الأدوات التالية:

(1أداة مربع تحرير وسرد Combobox لاختيار كليّة الطالب.

2) أداة groupbox الأولى (الخاصة ببيانات الطالب):

يوجد ضمنها أداة مربع نص TextBox لإدخال الرقم الجامعي، حيث عند إدخال الرقم الجامعي (إما كاملاً أو بإدخال آخر ثلاثة رمو.) يتم ملأ باقي الأدوات ضمن الـ Groupboxبيانات الطالب الموافق للرقم الجامعي المدخل (البيانات الشخصية وبيانات التسجيل).

3) أداة groupbox الثانية (الخاصة بدرجات السنة الأولى) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول، والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

4) أداة groupbox الثالثة (الخاصة بدرجات السنة الثانية) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول، والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

أداة groupbox الرابعة (الخاصة بدرجات السنة الثالثة) وتحتوي:

1. أداتي DataGridView، الأولى تحتوي درجات الفصل الأول والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.
2. أداة groupbox الخامسة (الخاصة بدرجات السنة الرابعة) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول، والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

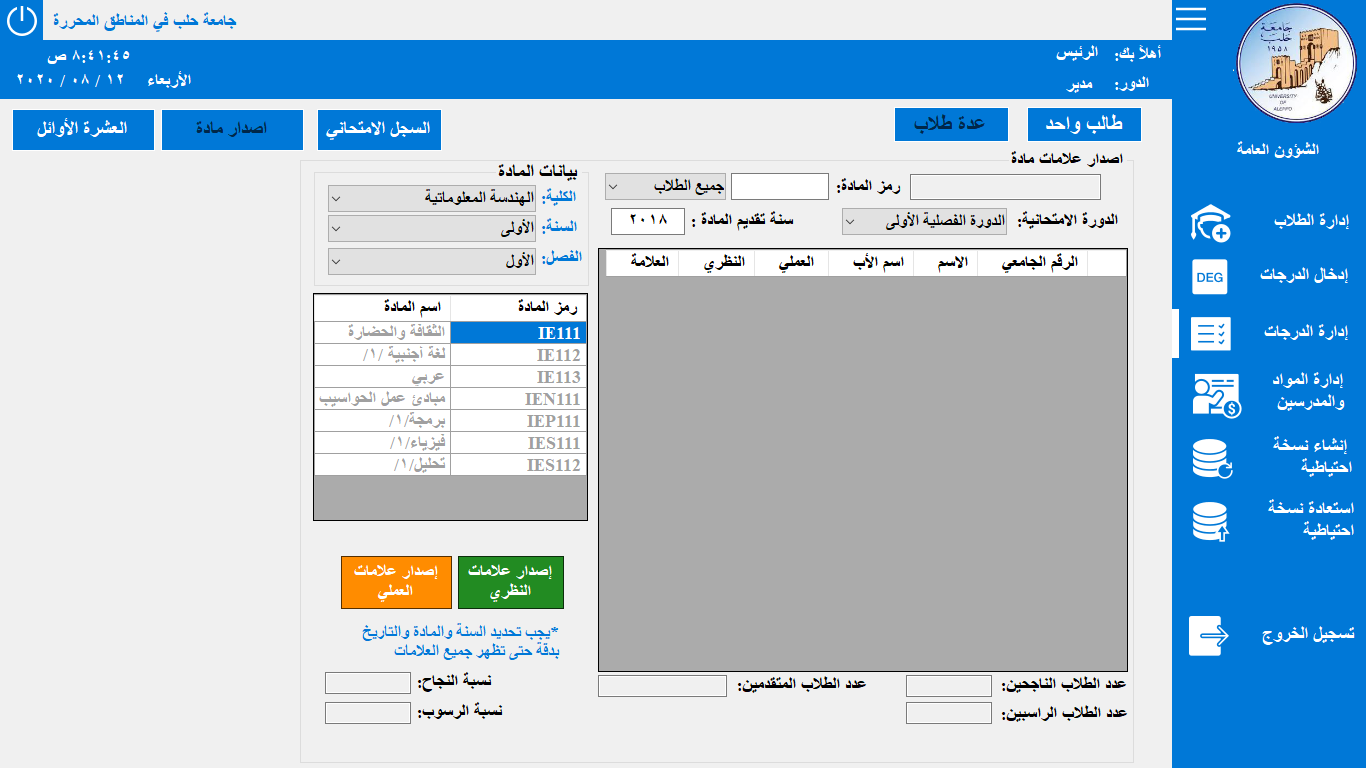
1. أداة groupbox السادسة (الخاصة بدرجات السنة الخامسة) وتحتوي:

أداتي DataGridView، الأولى تحتوي درجات الفصل الأول والثانية تحتوي درجات الفصل الثاني، وأداة مربع نص TextBox تظهر معدل الطالب في السنة.

1. زر Button " صحيفة الطالب PDF، عند الضغط على هذا الزر يتم تصدير صحيفة الطالب بصيغة PDF وحفظه في المجلد "أرشيف كشف العلامات" في المستندات.
2. أداة DataGridView وهذه الأداة تكون مخفية قمنا باستخدامها من أجل جلب بيانات الطالب إليها ثم تحويل البيانات إلى الأدوات في الـ groupbox الأولى، وذلك لأنه لا يمكن استيراد بيانات الطالب من جدول الطلاب إلى الأدوات داخل الـ groupbox بشكل مباشر.

أما القسم الثاني من واجهة إدارة الدرجات لعدة طلاب فهي تتكون من ثلاث نوافذ (إصدار مادة، السجل الامتحاني، العشرة الأوائل)، وسنتكلم عن هذه النوافذ الثلاثة:

النافذة الأولى إصدار مادة:

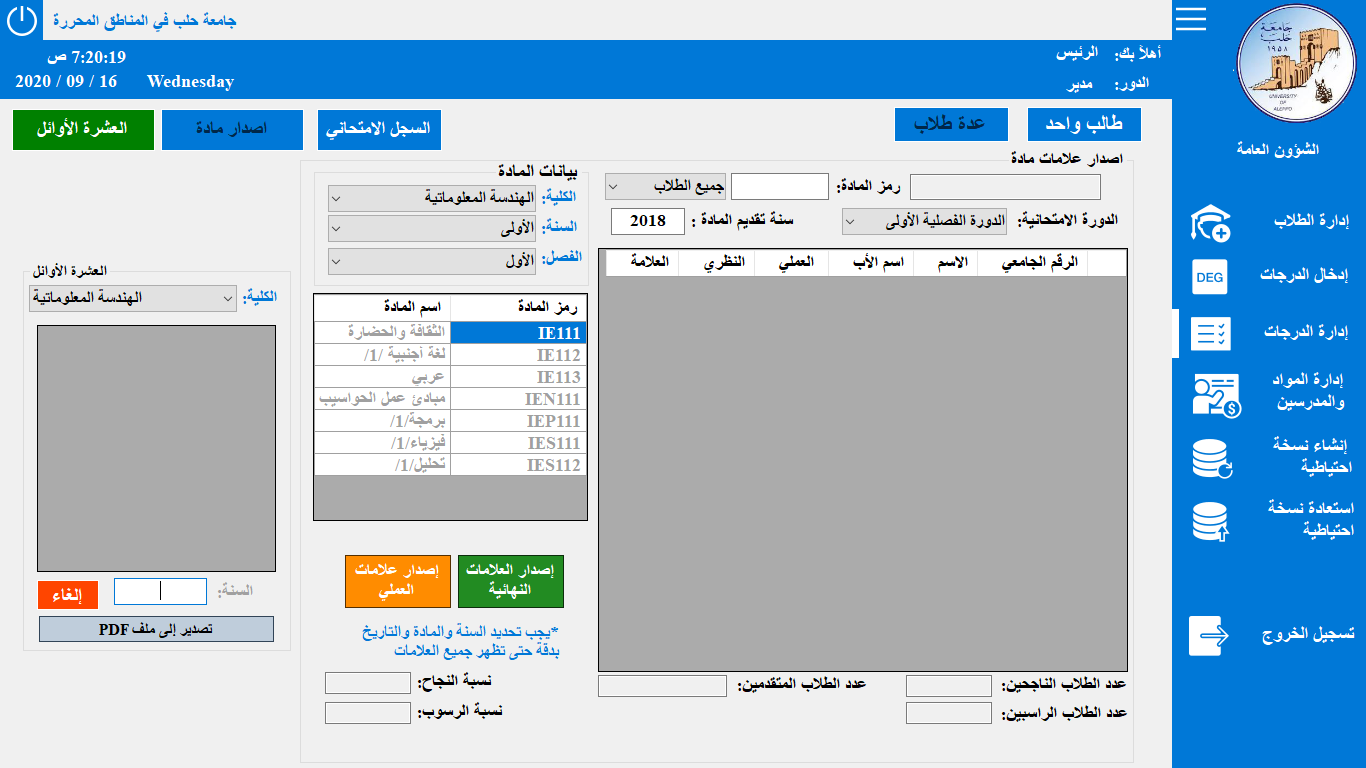
وهي للاطلاع على العلامات في مادة معينة وتصديرها إلى ملف pdf وتظهر بعد الضغط على زر **عدة طلاب** في إدارة الدرجات ومن ثم الضغط على زر **إصدار مادة** فتظهر كما في الشكل (4-21):

الشكل (4-21) واجهة إصدار مادة.

وتحتوي على الأدوات البرمجية التالية:

1. أداة مربع نص TextBox لإظهار اسم المادة المراد اصدار علاماتها.
2. أداة مربع نص TextBox لإدخال رمز المادة المراد اصدار علاماتها.
3. أداة مربع تحرير وسرد Combobox لاختيار الطلاب الناجحين أو الراسبين أو كليهما معاً.
4. أداة مربع تحرير وسرد Combobox لاختيار الدورة الامتحانية التي تم تقديم المادة فيها.
5. أداة مربع نص TextBox لإدخال سنة تقديم المادة.
6. أداة datagridview لعرض علامات الطلاب مع أسمائهم وأرقامهم الجامعية.
7. أداة مربع نص TextBox لإظهار عدد الطلاب المتقدمين للمادة.
8. أداة مربع نص TextBox لإظهار عدد الطلاب الناجحين.
9. أداة مربع نص TextBox لإظهار عدد الطلاب الراسبين.
10. أداة groubbox تتضمن ثلاثة أدوات مربع تحرير وسرد Combobox لاختيار الكلية والسنة الدراسية والفصل.
11. أداة datagridview لعرض أسماء المواد في الفصل الواحد مع رموزها.
12. أداة button لإصدار العلامات النهائية كملف pdf.
13. أداة button لإصدار علامات العملي كملف pdf.
14. أداة مربع نص TextBox لإظهار النسبة المئوية للنجاح.
15. أداة مربع نص TextBox لإظهار النسبة المئوية للرسوب.

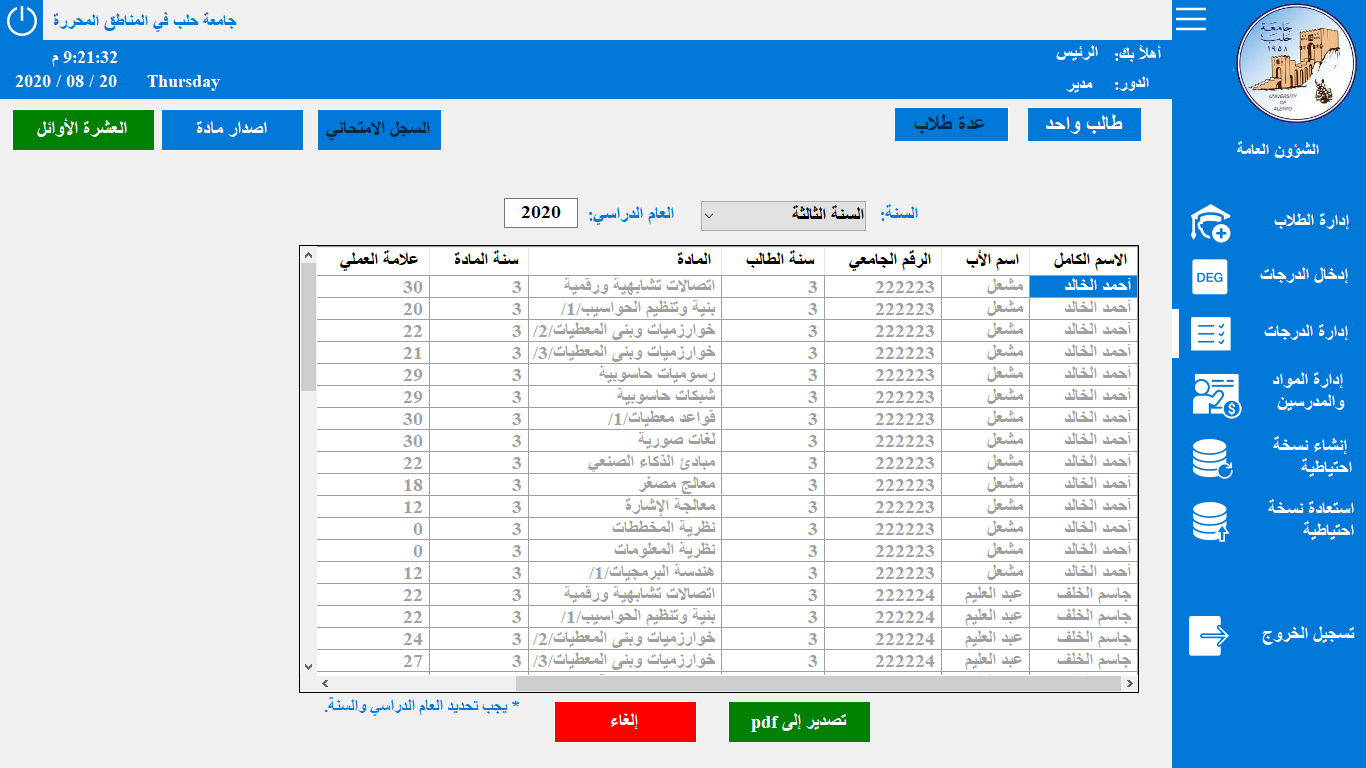
ولإصدار علامات مادة معينة يجب تحديد الدورة الامتحانية وسنة تقديم المادة بالإضافة إلى إدخال رمز المادة في الحقل المخصص لها إما إدخال يدوي أو يمكن إدخال الرمز آلياً عن طريق قائمة المواد التي على اليسار وذلك بعد تحديد الكلية والسنة الدراسية والفصل الدراسي فتظهر جميع مواد الفصل المحدد في القائمة، وبالضغط مرة واحدة على اسم المادة يتم إدخال رمزها تلقائيا في حقل رمز المادة وبالتالي يتم عرض علامات جميع الطلاب الذين تقدموا للمادة في الدورة الامتحانية المحددة كما في الشكل (4-22):

**الشكل (4-22) إصدار علامات مادة معينة.**

حيث يقوم بذلك الإجرائية البرمجية export\_mat\_lot وبواسطة الإجرائيات المخزنة [get\_mat\_year]) & [get\_mat\_year2] & [get\_mat\_year3] ).

ولتصدير العلامات إلى ملف pdf فنضغط على زر **إصدار علامات النظري** لإصدار علامات النظري أو نضغط على زر **إصدار علامات العملي** إذا أردنا إصدار علامات العملي فقط وتم استخدام البرمجة التفرعية لتسريع عملية التصدير، كما يتم تخزين الملف تلقائياً ضمن مجلد اسمه أرشيف علامات المواد إذ يتم إنشاء هذا المجلد داخل مجلد المستدات في الحاسوب لأول عملية تصدير، وبعدها تُخزن باقي ملفات العلامات التي تم تصديرها فيه.

النافذة الثانية السجل الامتحاني:

****بالنسبة لهذه النافذة فهي مسؤولة عن إصدار السجل الامتحاني السنوي للطلاب في العام الدراسي المحدد، وذلك بهدف حفظ الدرجات الامتحانية وأرشفتها ضمن ملفات للرجوع لها عند الحاجة، وتفيد أيضاً في إصدار تقارير عن العملية الامتحانية، ومن فوائدها أيضاً الاحتفاظ بدرجات المواد الراسبة قبل تعديلها واستبدالها بالدرجة الجديدة (بعد نجاح المواد)، تظهر هذه النافذة بعد الضغط على زر السجلي السنوي"، ويوضح الشكل (4-23) هذه النافذة:

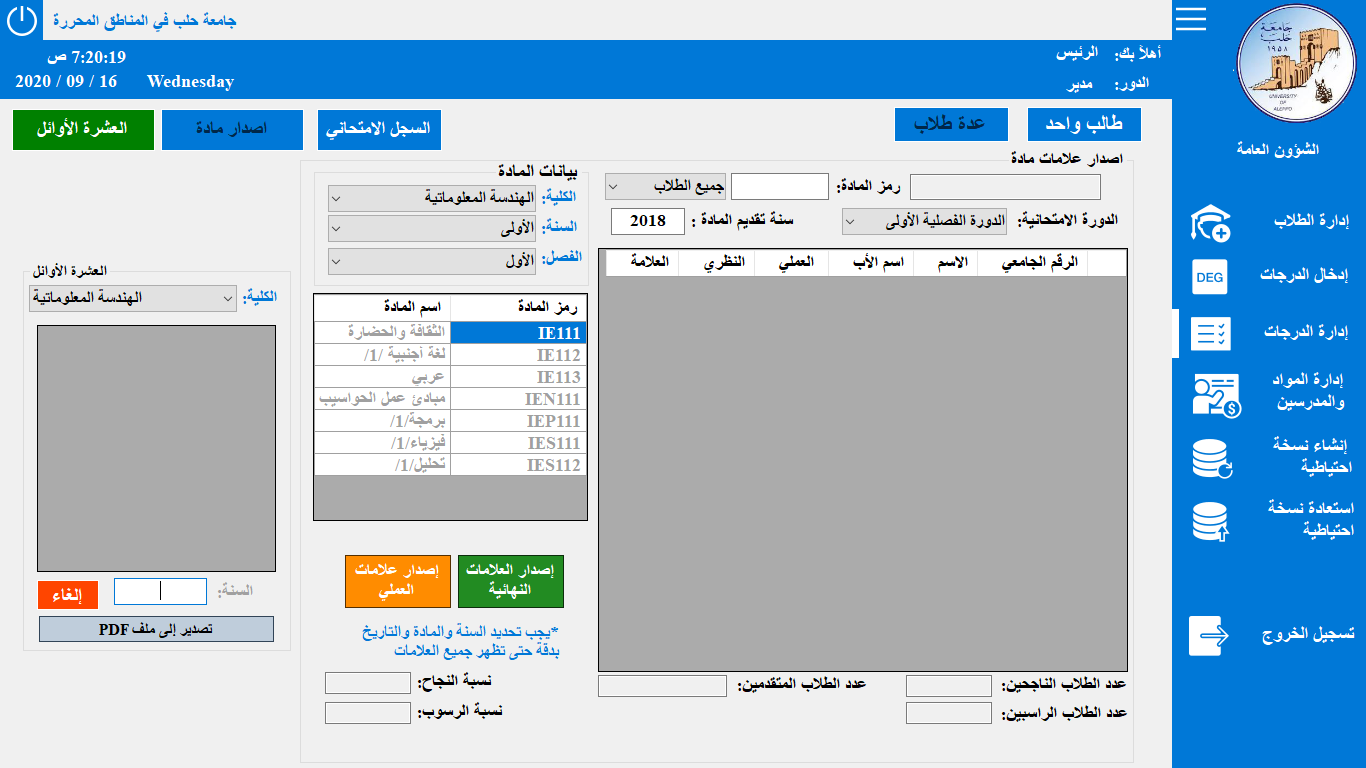
**الشكل (4-23) نافذة السجل الامتحاني السنوي.**

تحتوي النافذة السابقة على الأدوات التالية:

* أداة مربع تحرير وسرد Combobox لاختيار السنة الدراسية للطلاب.
* أداة مربع نص TextBox لإدخال عام تقديم الامتحان.
* أداة datagridview لعرض علامات الطلاب في العام المحدد مع أسمائهم وأرقامهم الجامعية.
* أداة زر Button لتصدير السجل السنوي إلى ملف بصيغة PDF.
* أداة زر Button لإلغاء العملية وإخفاء هذه النافذة.

النافذة الثالثة العشرة الأوائل:

عند الضغط على زر "العشرة الأوائل" تظهر هذه النافذة بجانب إحدى النافذتين (إصدار مادة أو السجل السنوي) وهي مخصصة لجلب الطلاب العشرة الأوائل بين طلاب سنة ما وإذا كان الطالب راسب بمادة وحدة على الأقل فلا يتم حساب معدله ووضعه مع العشرة الأوائل، ولها الشكل التالي:

**الشكل (4-24) نافذة العشرة الأوائل.**

وتحتوي على الأدوات التالية:

* أداة مربع نص TextBox لإدخال السنة الدراسية الطلاب.
* أداة datagridview لعرض أسماء الطلاب العشرة الأوائل مع أرقامهم الجامعية ومعدلاتهم.
* أداة زر Button لإلغاء العملية وإخفاء هذه النافذة.
* أداة زر Button لتصدير أسماء الطلاب العشرة الأوائل إلى ملف إيكسل.

**رابعاً: إدارة المواد والمدرسين**

تتكون هذه الواجهة من نافذتين:

أولاً: نافذة إدارة المواد:

في هذه النافذة يتم إدارة المواد وذلك بإجراء عمليات مختلفة على جدول المواد من إضافة وتعديل وحذف وبحث، وتظهر هذه النافذة بالضغط على زر "إدارة المواد" ولها الشكل التالي الشكل (4-25):

**الشكل (4-25) نافذة** إدارة المواد.

تتكون هذه النافذة من الأدوات التالية:

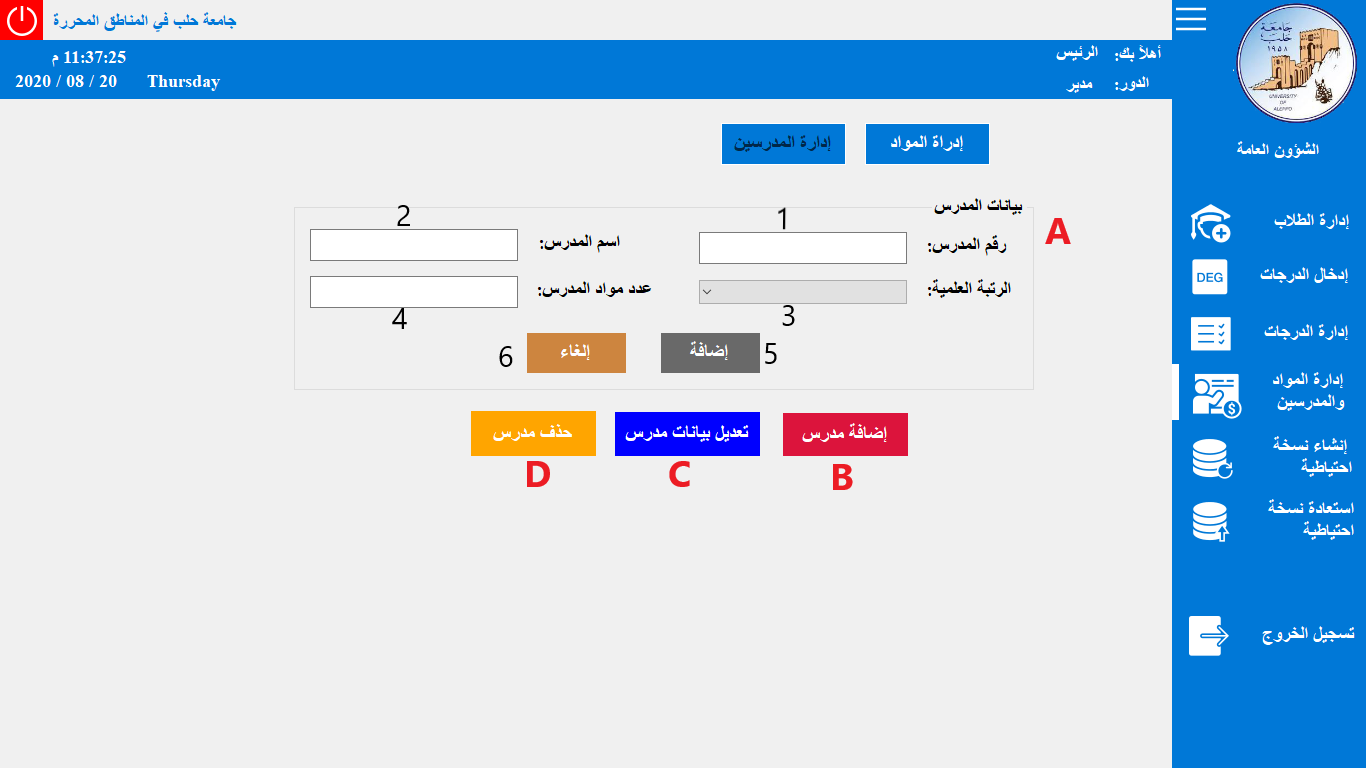
1. أداة groupbox خاصة ببيانات المادة وتحتوي:
2. أداة مربع نص TextBox لإدخال رمز المادة.
3. أداة مربع نص TextBox لإدخال اسم المادة.
4. أداة مربع تحرير وسرد Combobox لاختيار الكلية.
5. أداة مربع تحرير وسرد Combobox لاختيار سنة المادة (أولى، ثانية، ......).
6. أداة مربع تحرير وسرد Combobox لاختيار الفصل الدراسي (أول، ثاني).
7. أداة زر Button اسمه **إضافة مادة** لإضافة مادة جديدة حيث عند النقر على هذا الزر يتم إظهار أداة الـ groupbox الخاصة ببيانات المادة كما في الشكل (4-25)، فبعد أن تظهر هذه النافذة تقوم بتعبئة الحقول ببيانات المادة الجديدة المراد إدخالها وبعد ذلك نضغط على الزر الذي اسمه **إضافة** والموجود داخل الـ groupbox فيتم إضافة المادة إلى جدول المواد وذلك عن طريق الإجرائية المخزنة [add\_material].
8. أداة زر Button اسمه **تعديل بيانات مادة** نستخدمه لتعديل بيانات مادة موجوةد سابقاً في جدول المواد، حيث عند النقر على هذا الزر تظهر أداة datagridview تحوي بيانات جميع المواد، حيث يمكن البحث عن أي مادة (عن طريق مربع نص للبحث) وعند العثور على المادة المطلوبة يتم النقر عليها مرتين متتاليتين فيتم جلب بياناتها ووضعها تلقائياً في الأدوات الخاصة ببيانات المادة ليتم التعديل بكل سهولة, بعد ذلك بالضغط على زر **حفظ** من أجل حفظ البيانات بعد التعديل في جدول المواد في قاعدة البيانات بواسطة الإجرايئة المخزنة [edit\_material]، ويوضح الشكل (26-4) نافذة قائمة المواد من أجل اختيار مادة:

 **الشكل (4-26) نافذة قائمة المواد.**

1. أداة زر Button سمه **حذف مادة** نستخدمه لحذف مادة موجودة سابقاً في جدول المواد، حيث عند النقر على هذا الزر تظهر أداة datagridview تحوي بيانات جميع المواد، حيث يمكن البحث عن أي مادة (عن طريق مربع نص للبحث) وعند العثور على المادة المطلوبة يتم النقر عليها مرتين فتظهر رسالة تأكيد الحذف وبالنقر على زر "نعم" يتم حذف المادة من جدول المواد وذلك بواسطة الإجرائية المخزنة [delete\_material].

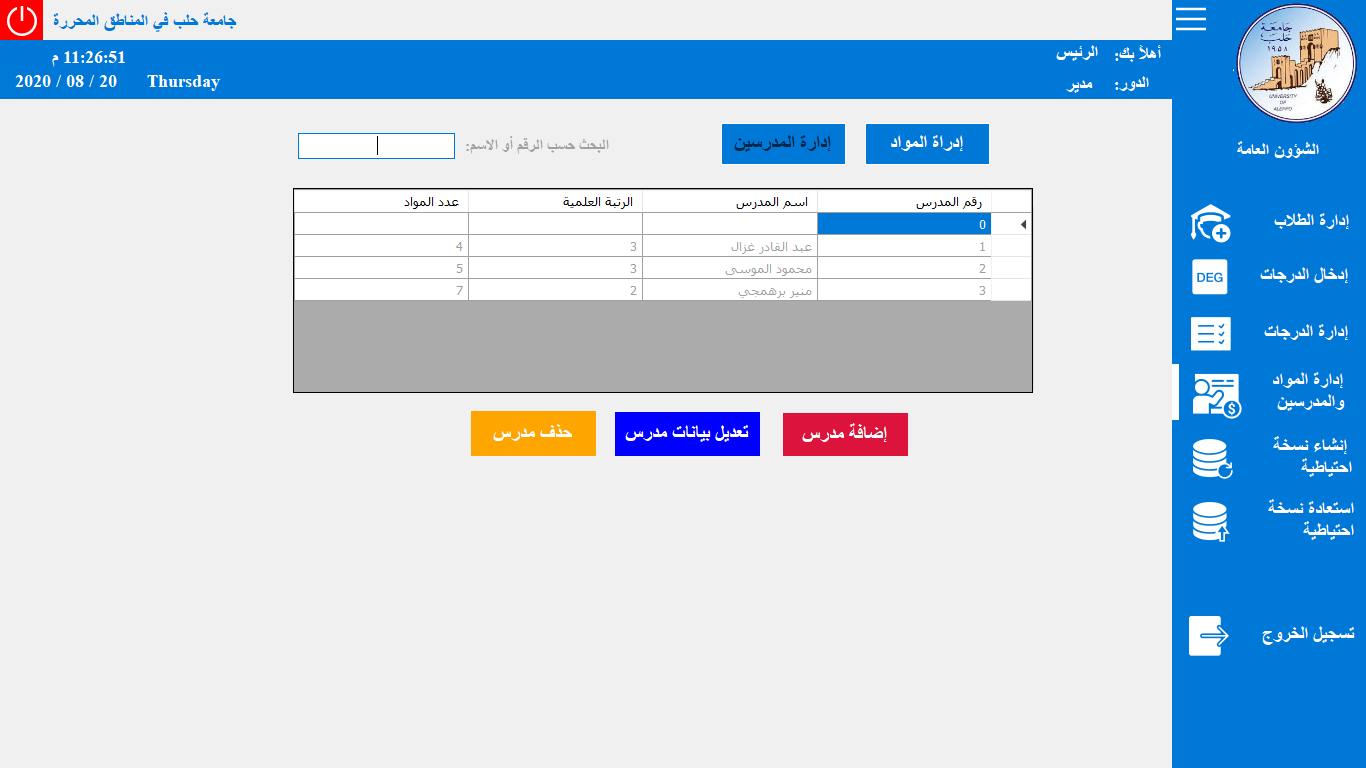
ثانياً: إدارة المدرسين:

في هذه النافذة يتم إدارة المدرسين وذلك بإجراء عمليات مختلفة على جدول المدرسين من إضافة وتعديل وحذف وبحث، وتظهر هذه النافذة بالضغط على زر "إدارة المدرسين" ولها الشكل التالي:

**الشكل (4-27) نافذة إدارة المدرسين.**

تتكون هذه النافذة من الأدوات التالية:

1. أداة groupbox خاصة ببيانات المدرس وتحتوي:
2. أداة مربع نص TextBox لإدخال رقم المدرس.
3. أداة مربع نص TextBox لإدخال اسم المدرس.
4. أداة مربع تحرير وسرد Combobox لاختيار الرتبة العلمية للمدرس (مدرس، معيد، ماجسيتر، دكتوراة، أستاذ دكتور).
5. أداة مربع نص TextBox لإدخال عدد المواد التي يدرسها المدرس.
6. أداة زر Button لإضافة مدرس جديد إلى جدول المدرسين.
7. أداة زر Button لإلغاء عملية الإضافة وحذف البيانات.
8. أداة زر Button لإضافة مدرس جديد حيث عند النقر على هذا الزر يتم إظهار أداة ال group box الخاصة ببيانات المدرس.
9. أداة زر Button لتعديل بيانات مدرس موجود سابقاً إلى جدول المدرسين، حيث عند النقر على هذا الزر تظهر أداة datagridview تحوي بيانات جميع المدرسين، حيث يمكن البحث عن أي مدرس (عن طريق مربع نص للبحث) وعند العثور على المدرس المطلوب يتم النقر عليه مرتين فيتم جلب بياناته ووضعها تلقائياً في الأدوات الخاصة ببيانات المدرس ليتم التعديل بعد ذلك بالضغط على زر حفظ، ويوضح الشكل (28-4) نافذة التعديل:

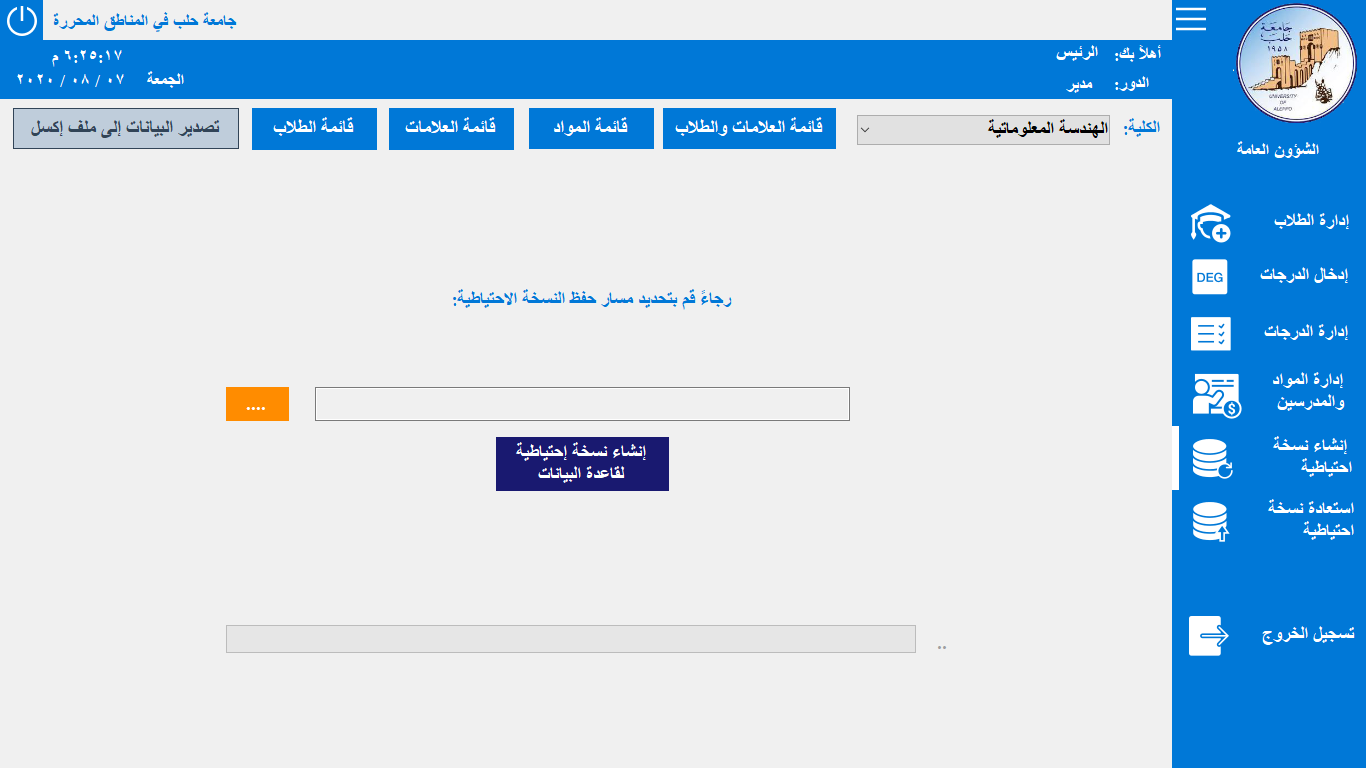
**الشكل (4-28) نافذة إدارة المدرسين (تعديل بيانات المدرس).**

1. أداة زر Button لحذف مدرس موجود سابقاً في جدول المدرسين، حيث عند النقر على هذا الزر تظهر أداة datagridview تحوي بيانات جميع المدرسين، حيث يمكن البحث عن أي مدرس (عن طريق مربع نص للبحث) وعند العثور على المدرس المطلوب يتم النقر عليه مرتين فتظهر رسالة تأكيد الحذف وبالنقر على زر "نعم" يتم الحذف.

**خامساً: إنشاء نسخة احتياطية**

يبين الشكل (4-29) تصميمه، وتحتوي على الأدوات التالية:

الواجهة الأساسية وهي واجهة إنشاء نسخة احتياطية تحوي:

* زر Button "...." لتحديد المسار المراد تصدير نسخة من قاعدة البيانات إليه.
* أداة مربع نص TextBox يتم فيها وضع المسار بعد أن قام المستخدم بتحديده.
* زر Button "إنشاء" يقوم بتصدير نسخة من قاعدة البيانات على شكل ملف بصيغة bak إلى المسار الذي تم تحديده.
* أداة folderBrowserDialog (مستعرض المجلدات)، لفتح نافذة يقوم المستخدم من خلالها باختيار المكان الذي يرغب بحفظ النسخة فيه، حيث تظهر هذه النافذة بعد الضغط على زر الاستعراض.

الشكل (4-29) تصميم الـ User Control الخاص بإنشاء نسخة احتياطية.

آلية العمل: عند تحديد المسار والضغط على زر إنشاء نسخة يتم تخزين المسار وإرساله إلى التابع (Backup\_database) ضمن الكلاس (ClsLogin) الذي يستدعي الإجرائية المخزنة Backup\_database)) فتقوم بحفظ ملف لقاعدة البيانات ضمن المسار المحدد.

في حال عدم تحديد مسار يتم حفظ الملف في المسار الافتراضي ضمن المستندات\أرشيف\ملفات قاعدة البيانات الاحتياطية.

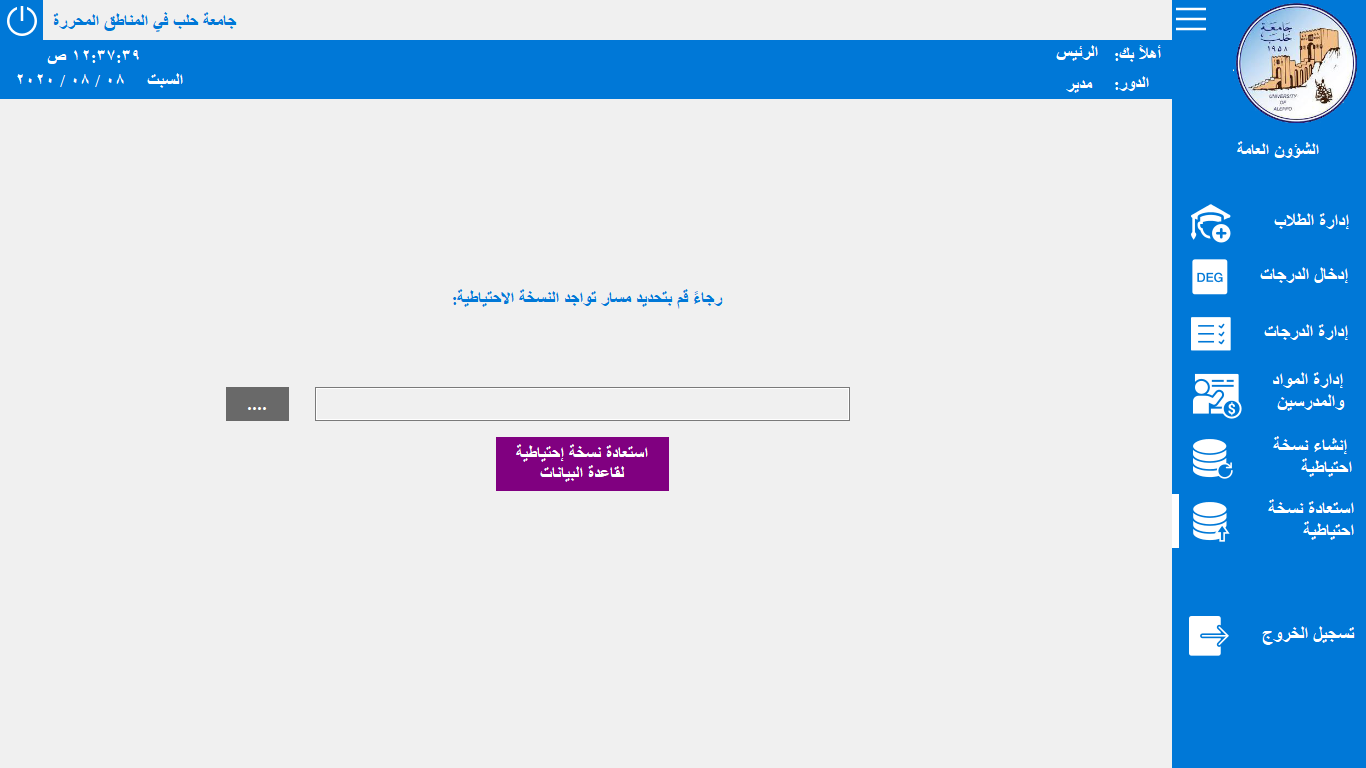
تحوي النافذة أيضا على أزرار:

* زر Button "قائمة الطلاب": يقوم بعرض قائمة بجميع الطلاب مع تفاصيل كل طالب ضمن أداة "DataGridViwe"، حيث يتم استدعاء التابع (GET\_All\_Student) ضمن الكلاس (Cls\_student) الذي يستدعي الإجرائية المخزنة (Get\_All\_student) والتي يعيد جدول ببيانات جميع الطلاب.
* زر Button "قائمة العلامات": يقوم بعرض قائمة بجميع العلامات مع تفاصيل كل علامة ضمن أداة "DataGridViwe"، حيث يتم استدعاء التابع(backup\_degree) ضمن الكلاس (Cls\_degree) الذي يستدعي الإجرائية المخزنة(backup\_degree) والتي يعيد جدول ببيانات جميع العلامات.
* أداة ComboBox "الكلية": يتم تحديد الكلية قبل الضغط على زر قائمة المواد أو زر قائمة العلامات والطلاب.
* زر Button "قائمة المواد": يقوم بعرض قائمة بجميع المواد مع تفاصيل كل مادة ضمن أداة "DataGridViwe"، حيث يتم إرسال رقم الكلية من أداة ComboBox للتابع(all\_mat) ضمن الكلاس (Cls\_degree) الذي يستدعي الإجرائية المخزنة(all\_mat) والتي يعيد جدول ببيانات جميع مواد الكلية المحددة.
* زر Button "قائمة العلامات والطلاب": يقوم بعرض قائمة الأعمدة فيها مواد الكلية والأسطر هي طلاب الكلية وكل سطر يحوي علامات الطالب في جميع مواده ضمن أداة "DataGridViwe"، حيث يتم إرسال رقم الكلية من أداة ComboBox للتابع(all\_degree\_with\_student) ضمن الكلاس (Cls\_degree) الذي يستدعي الإجرائية المخزنة(all\_degree\_with\_student).
* زر Button "تصدير البيانات إلى ملف إكسل": يتم تفعيل هذا الزر بعد الضغط على أحد الأزرار (قائمة الطلاب، قائمة العلامات، قائمة المواد، قائمة علامات والطلاب) حيث يتم تعبئة الأداة "DataGridViwe" بالبيانات المطلوبة، عند الضغط على هذا الزر يتم استدعاء الأداة "backgroundWorker".
* أداة "backgroundWorker": هي أداة العمل في الخلفية بحيث تنفذ التعليمات داخلها دون إيقاف البرنامج عن الاستجابة، تحوي على ثلاث توابع (تابع العمل "DoWork"، تابع عند حصول تغيير"ProgressChanged"، تابع الانتهاء"RunWorkerCompleted")، يقوم تابع العمل باستدعاء التابع المناسب للبيانات المعبئة ضمن ""DataGridViwe حيث يتم استعمال ملف الاكسل الموافق للبيانات وتعبئته، تم استعمال المعالجة التفرعية لزيادة سرعة التنفيذ، يتم عرض نسبة انهاء العمل ضمن أداة "progressBar" و "Lable" واللتين تتغيران عند حصول تقدم بالعمل عن طريق تابع حصول التغيير، عند انتهاء العملية يتم تخزين ملف الاكسل ضمن المسار: المستندات\أرشيف.

**سادساً: استعادة نسخة احتياطية**

يبين الشكل (4-30) تصميمه، وتحتوي على الأدوات التالية:

* زر Button "...." لتحديد الملف المراد استيراده والذي يعد نسخة من قاعدة البيانات، حيث يتم وضع المسار عند تحديده في الـ TextBox بجانب الزر.
* أداة مربع نص TextBox يتم فيها وضع مسار الملف بعد أن قام المستخدم بتحديده.
* زر Button "استعادة" يقوم باستيراد قاعدة البيانات من الملف (ذو الصيغة bak) من المسار الذي تم تحديده.
* أداة openFileDialog (مستعرض الملفات)، لفتح نافذة يقوم المستخدم من خلالها باختيار ملف النسخة الاحتياطية المراد استعادتها، حيث تظهر هذه النافذة بعد الضغط على زر الاستعراض.

آلية العمل: عند الضغط على زر استعادة نسخة احتياطية يتم إرسال مسار النسخة الاحتياطية إلى التابع (Restore\_database) ضمن الكلاس (ClsLogin) الذي يستدعي الإجرائية المخزنة (restore\_db) في قاعدة البيانات فيتم استعادة النسخة الاحتياطية.

الشكل (4-30) تصميم الـ User Control الخاص باستعادة نسخة احتياطية.

**القسم البرمجي**

**البرمجة الكائنية (OOP) Object Oriented Programming**

وهي عبارة عن نمط برمجة متقدمة، وفيه يقسم البرنامج إلى وحدات تسمى الكائنات Objects، كل كائن عبارة عن حزمة من البيانات والمتغيرات والثوابت والدوال ووحدات التنظيم وواجهات الاستخدام.

ويتم بناء البرنامج بواسطة استخدام الكائنات وربطها مع بعضها البعض وواجهة البرنامج الخارجية باستخدام هيكلية البرنامج وواجهات الاستخدام الخاصة بكل كائن.

**مجلدات البرنامج**

في البداية قمنا بتنظيم العمل عن طريق إنشاء مجلدات ضمن مستعرض مكونات المشروع Solution Explorer كما في الشكل (3-14)، والبالغ عددها أربع مجلدات:

* **BL (Business Layer)**

يمثل هذا المجلد طبقة العمل، حيث قمنا بوضع الصفوف Classes الثلاثة الرئيسية التي يعتمد عليها مشروعنا وهي:

* صف الدرجة (Cls\_degree).
* صف الطالب (Cls\_student).
* صف تسجيل الدخول (Cls\_Login).
* صف المدرسين والمواد (Cls\_TeacherandMaterial).
* **DAL (Data Access Layer)**

يمثل هذا المجلد طبقة الاتصال بقاعدة البيانات حيث يحتوي صف DataAccessLayer الخاص بنص الاتصال الذي يربط البرنامج بقاعدة البيانات.

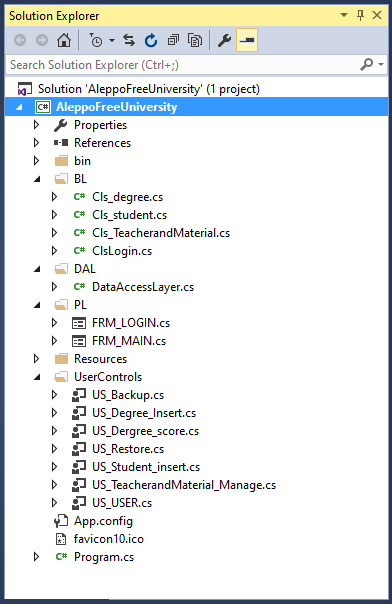
* **PL (Physical Layer)**

يمثل هذا المجلد الطبقة الفيزيائية ويتضمن الواجهات:

* واجهة تسجيل الدخول (FRM\_LOGIN).
* الواجهة الرئيسية (FRM\_MAIN).
* **User Controls**

قمنا بتجميع كافة أدوات User Control ضمن هذا المجلد**،**لدينا ست واجهات من النوع "User Control" وهي:

* واجهة انشاء نسخة احتياطية.
* واجهة إدخال الدرجات.
* واجهة إدارة الدرجات.
* واجهة استعادة نسخة احتياطية.
* واجهة إدارة الطلاب.
* واجهة إدارة المواد والمدرسين.

الشكل (4-31) المجلدات ضمن مستعرض مكونات المشروع.

**صفوف BL وDAL**

الأن سنقوم بشرح كل صف من صفوف الـ BL وDAL وفقاً لترتيب أهميته في المشروع:

**أولاً: صف (DataAccessLayer)**

هذا الصف هو الصف الأساسي في المشروع حيث من خلاله نستطيع ربط البرنامج مع قاعدة البيانات، في البداية يتم إنشاء كائن باسم sqlcon من صف الاتصال SqlConnection، يتكون هذا الصف من عدة توابع وإجرائيات:

* **الباني:** هو تابع يحمل نفس اسم الكلاس (DataAccessLayer) ويجب أن يكون من نوع Public يتم ضمنه إنشاء نص الاتصال بقاعدة البيانات عن طريق تمرير البارامترات التالية (اسم السيرفر، اسم قاعدة البيانات، اسم الدخول، كلمة المرور) للكائن sqlcon، يتم الحصول على البارامترات من الملف النصي الذي ينشأ عند تسجيل الدخول للبرنامج في واجهة تسجيل الدخول.
* **إجرائية فتح الاتصال (open):** يتم من خلال هذه الإجرائية فحص الكائن الخاص بالاتصال (sqlcon) إذا كان مغلق يتم فتحه.
* **إجرائية إغلاق الاتصال (close):** يتم من خلال هذه الإجرائية فحص الكائن الخاص بالاتصال (sqlcon) إذا كان مفتوح يتم إغلاقه.
* **تابع جلب البيانات (SelecteData):** يستخدم هذا التابع لجلب البيانات من قاعدة البيانات، حيث نمرر له بارامترين (الأول اسم الإجرائية المخزنة (Stored\_Procedure) المطلوب تنفيذها في قاعدة البيانات، البارامتر الثاني هو مصفوفة البارامترات التابعة للإجرائية المخزنة).
* **إجرائية تنفيذ الأوامر (ExecuteCommand):** تستخدم هذه الإجرائية لإرسال الأوامر إلى قاعدة البيانات من أجل تنفيذها، ونمرر لها أيضاً نفس بارامترات التابع السابق.

**ثانياً: صف تسجيل الدخول (Cls\_Login)**

هذا الصف يحوي على التوابع والإجرائيات الخاصة بتسجيل الدخول والنسخ الاحتياطي والاستعادة:

* **تابع تسجيل الدخول (LOGIN):** نمرر له بارامترين، الأول اسم المستخدم، والثاني كلمة المرور حيث يتم تنفيذ الإجرائية المخزنة الخاصة بتسجيل الدخول"Sp\_Login" باستخدام التابع SelecteData ضمن تابع تسجيل الدخول.
* **تابع النسخ الاحتياطي (Backup\_database):** نمرر له المسار الذي سيتم تخزين ملف قاعدة البيانات فيه، عن طريق التابع Select Data يقوم باستدعاء الإجرائية المخزنة "Backup\_database" لتقوم بعمل نسخة احتياطية.
* **تابع استعادة نسخة احتياطية (Restore\_database):** نمرر له المسار الذي سيتم استعادة ملف قاعدة البيانات منه، يقوم هذا التابع بالاتصال بقاعدة البيانات "Master" حتى يتمكن من استدعاء الإجرائية المخزنة "restore\_db" لتقوم باستعادة نسخة احتياطية.

**ثالثاً: صف الطالب (Cls\_student)**

ويوجد ضمنه التوابع والإجرائيات المخصصة لإدارة الطلاب:

* **تابع إضافة الطالب (add\_student):** نمرر لها البارامترات الخاصة ببيانات الطالب البالغ عددها ثمانية عشر بارامتر، فيتم ضمن الإجرائية تعبئة هذه البارامترات ضمن مصفوفة ثم يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل البيانات إلى الإجرائية المخزنة ""add\_user ليتم تنفيذ عملية الإضافة.
* **تابع إضافة تفاصيل الطالب (add\_student\_detile):** نمرر لها البارامترات الخاصة ببيانات تفاصيل الطالب البالغ عددها ست وثلاثون بارامتر، فيتم ضمن الإجرائية تعبئة هذه البارامترات ضمن مصفوفة ثم يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل البيانات إلى الإجرائية المخزنة "add\_student\_detile" ليتم تنفيذ عملية الإضافة.
* **تابع تعديل بيانات الطالب (edit\_student):** نمرر له البارامترات الخاصة ببيانات الطالب وتفاصيل الطالب البالغ عددها واحد وخمسون بارامتر، فيتم ضمن التابع تعبئة هذه البارامترات ضمن مصفوفة ثم يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل البيانات إلى الإجرائية المخزنة " edit\_student" ليتم تنفيذ عملية التعديل.
* **تابع تفاصيل الطالب (chek\_student\_detile):** نمرر له الرقم الجامعي للطالب المراد معرفة تفاصيله، يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل الرقم الجامعي إلى الإجرائية المخزنة "chek\_student\_detile" ليتم إعادة النتيجة.
* **تابع حذف طالب (delet\_student):** نمرر له الرقم الجامعي للطالب المراد حذفه، فيتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل الرقم الجامعي إلى الإجرائية المخزنة "delete\_student" لتتم عملية الحذف.
* **تابع البحث عن طالب بالرقم الجامعي (Search\_Student):** ضمن هذا التابع يتم استدعاء التابع SelectData ونمرر له اسم الإجرائية المخزنة الخاصة بالبحث عن بيانات الطالب "Search\_Student" مع تمرير بارامتر واحد يمثل الرقم الجامعي للطالب المراد البحث عنه، ويتم تخزين نتيجة البحث ضمن التابع Search\_Student.
* **تابع البحث عن طالب بالاسم (Search\_Student\_name):** ضمن هذا التابع يتم استدعاء التابع SelectData ونمرر له اسم الإجرائية المخزنة الخاصة بالبحث عن بيانات الطالب بالاسم "Search\_Student\_name" مع تمرير بارامتر واحد يمثل الاسم الأول للطالب المراد البحث عنه، ويتم تخزين نتيجة البحث ضمن التابع Search\_Student\_name.
* **تابع جلب بيانات جميع الطلاب (GET\_All\_Student):** ضمن هذا التابع يتم استدعاء التابع SelectData ونمرر له اسم الإجرائية المخزنة الخاصة بجلب بيانات الطلاب"Get\_All\_student"، فيتم اعادة النتيجة.
* **تابع جلب بيانات الطالب (GET\_DETILE\_STUD):** ضمن هذا التابع يتم استدعاء التابع SelectData ونمرر له اسم الإجرائية المخزنة الخاصة بجلب بيانات الطالب (المعلومات الشخصية للطالب) مع تمرير بارامتر واحد يمثل الرقم الجامعي للطالب المراد جلب تفاصيله، ويتم تخزين نتيجة البحث ضمن التابع " Student\_detiles".
* **تابع جلب رقم الطالب (GET\_stdnum):** ضمن هذا التابع يتم استدعاء التابع SelectData ونمرر له اسم الطالب الكامل إلى الإجرائية "Get\_stdnum" فيتم إعادة رقم الطالب الجامعي.
* **تابع جلب اسم الطالب (GET\_stdname):** ضمن هذا التابع يتم استدعاء التابع SelectData ونمرر له رقم الطالب إلى الإجرائية "Get\_stdname" فيتم إعادة اسم الطالب الجامعي.
* **تابع البحث عن طريق آخر ثلاث أرقام (Search\_by3id):** يتم استدعاء التابع SelectData ويتم إرسال آخر ثلاث أرقام من الرقم الجامعي مع رقم الكلية إلى الإجرائية "Search\_by3id" فيتم إعادة الرقم الجامعي كاملا.
* **تابع جلب رسوم الطلاب (get\_all\_fees):** ضمن هذا التابع يتم استدعاء التابع SelectData ونمرر له الإجرائية المخزنة "get\_all\_fees" فتقوم بإعادة بيانات الرسوم لجميع الطلاب.
* **تابع إضافة رسوم طالب (add\_fees):** نمرر لها البارامترات الخاصة ببيانات رسوم الطالب البالغ عددها أربعة عشر بارامتر، فيتم ضمن الإجرائية تعبئة هذه البارامترات ضمن مصفوفة ثم يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل البيانات إلى الإجرائية المخزنة "add\_fees " ليتم تنفيذ عملية الإضافة.
* **تابع تعديل رسوم طالب (edit\_fees):** نمرر له البارامترات الخاصة ببيانات رسوم الطالب البالغ عددها أربعة عشر بارامتر، فيتم ضمن التابع تعبئة هذه البارامترات ضمن مصفوفة ثم يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل البيانات إلى الإجرائية المخزنة "edit\_fess" ليتم تنفيذ عملية التعديل.
* **تابع حذف طالب (delete\_fees):** نمرر له الرقم الجامعي والسنة الدراسية التي سدد الطالب رسومها والعام الدراسي الموافق لها، فيتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل الرقم الجامعي إلى الإجرائية المخزنة "delete\_fees" لتتم عملية الحذف.
* **تابع الحصول على بيانات الرسوم لطالب (get\_std\_fees):** نمرر له الرقم الجامعي فيتم إرساله إلى الإجرائية "get\_std\_fees" فتعيد بيانات جميع رسوم الطالب.
* **تابع الحصول على الطلاب الناجحين (success\_year):** يتم استدعاء التابع SelectData ونمرر له عدد مواد الترفع خلال هذا العام ورقم الكلية والسنة إلى الإجرائية المخزنة "success\_year" فتعيد قائمة بأسماء الطلاب الناجحينوأرقامهم والسنة الدراسية وعدد المواد الراسبة وعدد المواد الكلية المقدمة.
* **تابع الحصول على الطلاب الراسبين (not\_success\_year):** يتم استدعاء التابع SelectData ونمرر له عدد مواد الترفع خلال هذا العام ورقم الكلية والسنة إلى الإجرائية المخزنة "not\_success\_year" فتعيد قائمة بأسماء الطلاب الراسبينوأرقامهم والسنة الدراسية وعدد المواد الراسبة وعدد المواد الكلية المقدمة.
* **تابع ترفيع الطلاب الناجحين (success\_year\_ok):** يتم استدعاء التابع SelectData ونمرر له عدد مواد الترفع خلال هذا العام ورقم الكلية والسنة إلى الإجرائية المخزنة "success\_year\_ok" فتعيد قائمة بأسماء الطلاب الناجحينوأرقامهم والسنة الدراسية وعدد المواد الراسبة وعدد المواد الكلية المقدمة.

**رابعاً: صف الدرجة (Cls\_degree)**

ويوجد ضمنه التوابع والإجرائيات المخصصة لإدارة الدرجة:

* **تابع إدخال الدرجة (insert\_degree):** نمرر له البارامترات الخاصة ببيانات الدرجة البالغ عددها ستة بارامترات، فيتم ضمن التابع تعبئة هذه البارامترات ضمن مصفوفة ثم يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل البيانات إلى الإجرائية المخزنة "insert\_degree" ليتم تنفيذ عملية الإدخال.
* **تابع تعديل درجة (edit\_degree):** يتم من خلال هذا التابع تعديل بيانات الدرجة، حيث نمرر له البارامترات الخاصة بالدرجة (الرقم الجامعي للطالب، رمز المادة، الدرجة العملية، الدرجة النظرية، الدورة الامتحانية، عام تقديم المادة)، فتقوم الإجرائية ExecuteCommand بإرسال هذه البارامترات إلى الإجرائية المخزنة "edit\_degree" في قاعدة البيانات لتتم عملية التعديل.
* **تابع حذف درجة (delete\_degree):** نمرر لهذا التابع الرقم الجامعي للطالب المُراد حذف درجته في مادة ما فتقوم الإجرائية ExecuteCommand بإرسال هذه البارامتر إلى الإجرائية المخزنة delete\_degree"" في قاعدة البيانات لتتم عملية الحذف.
* **تابع جلب علامات مواد فصل واحد (get\_degreesWithSeson):** نمرر لهذا التابع رقم الطالب والسنة الدراسية والفصل، فيقوم التابع SelectData بإرسال البارامترات إلى الإجرائية المخزنة "get\_degreesWithSeson" في قاعدة البينات فتعيد علامات الطالب للسنة والفصل المحددين.
* **تابع البحث عن درجة معينة (Search\_Degree):** يقوم بالبحث عن بيانات درجة معينة إما عن طريق الرقم الجامعي للطالب أو رقم المادة (رمزها)، حيث نمرر هذين البارامترين إلى هذا التابع فيقوم التابع SelectData بتنفيذ الإجرائية المخزنة "Search\_Degree" وإرجاع نتيجة البحث إلى التابع.
* **تابع جلب الدرجة (GET\_DEGREE):** ضمن هذا التابع يتم استدعاء التابع SelectData ونمرر له اسم الإجرائية المخزنة الخاصة بجلب بيانات الدرجة مع تمرير ثلاث بارامترات (رقم المادة وعام صدور الدرجة والدورة الامتحانية)، ويتم تخزين النتيجة ضمن التابع DEGREE\_SCORE"".
* **تابع جلب كل الدرجات (GET\_All\_DEGREE):** من خلال هذا التابع يتم جلب كافة الدرجات مع البيانات الخاصة بكل درجة وذلك بتنفيذ الإجرائية المخزنة "Get\_All\_Degree" عن طريق التابع SelectData، ويتم تخزين النتيجة ضمن التابع.
* **تابع حساب معدل الطالب (GET\_AVG\_STUDENT):** نمرر له الرقم الجامعي والسنة الدراسية للطالب المراد حساب معدل درجاته، فيقوم التابع SelectData بإرسال هذه البارامترات إلى الإجرائية المخزنة GET\_AVG\_DEGREE"" فتقوم هذه الإجرائية المخزنة بحساب معدل درجات الطالب في السنة المحددة وذلك بجمع الدرجات النهائية وتقسيم المجموع على عدد هذه المواد، ويتم إعادة النتيجة إلى التابع.
* **تابع جلب كل المواد (Get\_All\_Material):** نحدد لهذا التابع رقم الكلية والسنة الدراسية والفصل، فيقوم التابع SelectData بإرسال هذه القيم إلى الإجرائية المخزنة Get\_All\_Material"" التي تُرجع أسماء المواد حسب القيم المحددة.
* **تابع إيجاد العشرة الأوائل (Get\_Top10):** نحدد لهذا التابع السنة الدراسية، فيقوم التابع SelectData بإرسالها إلى الإجرائية المخزنة get\_top10"" التي تعيد أسماء العشرة الأوائل على الدفعة مع رقم ومعدل كل طالب.
* **تابع جلب المواد غير المقدمة للطالب (null\_matirial):** نحدد لهذا التابع رقم الطالب، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة null\_matirial"" التي تعيد أسماء المواد التي لم يقدمها الطالب.
* **تابع جلب المواد الراسبة للطالب (fail\_degree):** نحدد لهذا التابع رقم الطالب، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة fail\_degree"" التي تعيد أسماء المواد الراسبة للطالب.
* **تابع جلب أسماء طلاب السنة (get\_all\_name\_in\_year):** نحدد لهذا التابع رمز المادة والعام لتاريخ تقديم المادة، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة get\_all\_name\_in\_year "" التي تعيد أسماء جميع طلاب السنة وأرقامهم.
* **تابع جلب علامات الدفعة (get\_mat\_year1):** نحدد لهذا التابع رقم الكلية والسنة، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة get\_mat\_year"" التي تعيد علامات العملي والنظري لجميع طلاب السنة.
* **تابع جلب علامات الدفعة (get\_mat\_year2):** نحدد لهذا التابع رقم الكلية والسنة، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة get\_mat\_year2"" التي تعيد علامات العملي والنظري للطلاب الناجحين في السنة.
* **تابع جلب علامات الدفعة (get\_mat\_year3):** نحدد لهذا التابع رقم الكلية والسنة، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة get\_mat\_year3"" التي تعيد علامات العملي والنظري لجميع للطلاب الراسبين في السنة.
* **تابع جلب السجل الامتحاني (exam\_record):** نحدد لهذا التابع السنة الدراسية والعام الدراسي، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة "exam\_record" التي تعيد السجل السنوي الذي يحوي على علامات جميع طلاب الدفعة في جميع المواد التي قدمت في العام الدراسي المحدد.
* **تابع نسخ الاحتياطي للعلامات (backup\_degree):** يقوم التابع SelectData باستدعاء الإجرائية المخزنة "backup\_degree" التي تعيد بيانات جميع العلامات في جدول العلامات.
* **تابع نسخ الاحتياطي لمواد الكلية(all\_mat):** نحدد لهذا التابع الكلية، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة "all\_mat" التي تعيد بيانات جميع المواد في جدول المواد.
* **تابع نسخ الاحتياطي لقائمة الطلاب والعلامات(all\_mat):** نحدد لهذا التابع الكلية، فيقوم التابع SelectData بإرساله إلى الإجرائية المخزنة "all\_degree\_with\_student" التي تعيد جدول الأعمدة فيه هي مواد الكلية والأسطر هي علامات كل طالب في جميع المواد.

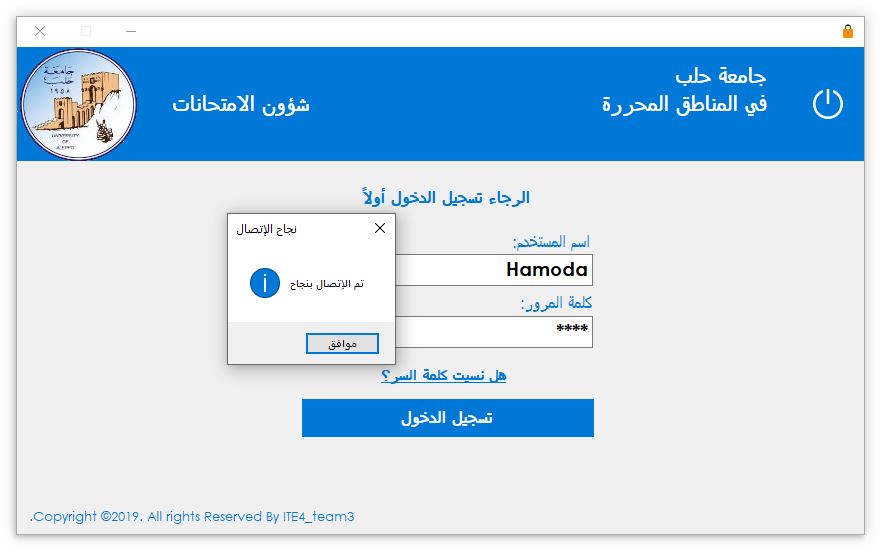
**خامسا: صف المدرسين والمواد (Cls\_TeacherandMaterial)**يحتوي على التوابع الخاصة بالمدرسين والمواد:

* **تابع جلب جميع المدرسين (Get\_all\_teacher):** يقوم التابع SelectData باستدعاء الإجرائية المخزنة "Get\_all\_teacher" التي تعيد بيانات جميع المدرسين في جدول المدرسين.
* **تابع إضافة مدرس(add\_teacher):** نمرر له الرقم الخاص بالمدرس واسم المدرس والرتبة العلمية وعدد مواد التدريس ثم يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل البيانات إلى الإجرائية المخزنة "add\_teacher" ليتم تنفيذ عملية الاضافة.
* **تابع تعديل بيانات المدرس (edit\_teacher):** يتم من خلال هذا التابع تعديل بيانات الدرجة، حيث نمرر له البارامترات الخاصة بالمدرس (الرقم الخاص بالمدرس، اسم المدرس، الرتبة العلمية وعدد مواد التدريس)، فتقوم الإجرائية ExecuteCommand بإرسال هذه البارامترات إلى الإجرائية المخزنة "edit\_teacher" في قاعدة البيانات لتتم عملية التعديل.
* **تابع حذف مدرس (delete\_teacher):** نمرر لهذا التابع الرقم الخاص بالمدرس المُراد حذفه فتقوم الإجرائية ExecuteCommand بإرسال هذه البارامتر إلى الإجرائية المخزنةdelete\_teacher"" في قاعدة البيانات لتتم عملية الحذف.
* **تابع البحث عن مدرس عبر الرقم (search\_teacher\_by\_id):** نمرر لهذا التابع الرقم الخاص بالمدرس المُراد البحث عنه فيقوم التابع SelectData باستدعاء الإجرائية المخزنة search\_teacher\_by\_id"" في قاعدة البيانات لتتم إعادة بيانات المدرس المحدد.
* **تابع البحث عن مدرس عبر الاسم (search\_teacher\_by\_name):** نمرر لهذا التابع اسم المدرس المُراد البحث عنه فيقوم التابع SelectData باستدعاء الإجرائية المخزنة search\_teacher\_by\_name"" في قاعدة البيانات لتتم إعادة بيانات المدرس المحدد.
* **تابع جلب جميع المواد (Get\_material):** يقوم التابع SelectData باستدعاء الإجرائية المخزنة "Get\_Material" التي تعيد بيانات جميع المواد في جدول المواد.
* **تابع إضافة مادة(add\_material):** نمرر له كود المادة واسمها والكلية والسنة ثم يتم استدعاء الإجرائية ExecuteCommand والتي بدورها سوف ترسل البيانات إلى الإجرائية المخزنة "add\_material" ليتم تنفيذ عملية الاضافة.
* **تابع تعديل بيانات المادة (edit\_material):** يتم من خلال هذا التابع تعديل بيانات المادة، حيث نمرر له البارامترات الخاصة بالمادة (كود المادة، اسم المادة، الكلية والسنة)، فتقوم الإجرائية ExecuteCommand بإرسال هذه البارامترات إلى الإجرائية المخزنة "edit\_material " في قاعدة البيانات لتتم عملية التعديل.
* **تابع حذف مادة (delete\_material):** نمرر لهذا التابع كود المدة المُراد حذفها فتقوم الإجرائية ExecuteCommand بإرسال هذه البارامتر إلى الإجرائية المخزنة delete\_material"" في قاعدة البيانات لتتم عملية الحذف.
* **تابع البحث عن مادة عبر الرقم (search\_material\_by\_id):** نمرر لهذا التابع بالمادة المُراد البحث عنها فيقوم التابع SelectData باستدعاء الإجرائية المخزنة search\_material\_by\_id"" في قاعدة البيانات لتتم إعادة بيانات المادة المحددة.
* **تابع البحث عن مادة عبر الاسم (search\_material\_by\_name):** نمرر لهذا التابع اسم المادة المُراد البحث عنها فيقوم التابع SelectData باستدعاء الإجرائية المخزنة search\_material\_by\_name"" في قاعدة البيانات لتتم إعادة بيانات المادة المحددة.

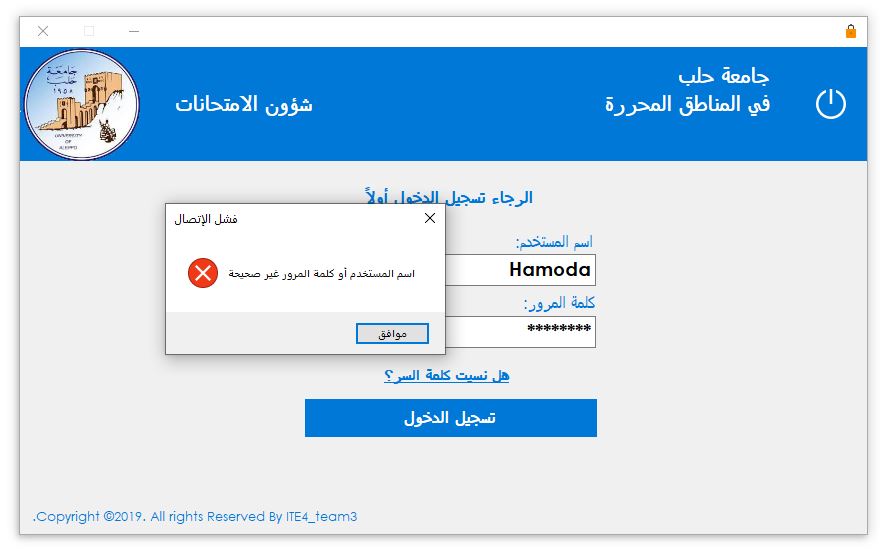
**صفوف PL وUser Controls**

**أولاً: صف واجهة تسجيل الدخول (frmlogin)**

في البداية قمنا بإنشاء كائن من صف تسجيل الدخول، وكائن من صف الواجهة الرئيسية.

* يحوي هذا الصف على إجرائية باسم login يتم بداخلها إنشاء كائن من الكلاس DataTable باسم dt ونقوم باستدعاء التابع LOGIN من صف تسجيل الدخول مع تمرير القيم المدخلة في مربعات النص التي تمثل اسم المستخدم وكلمة المرور، حيث يتم ضمن التابع LOGIN تنفيذ الإجرائية المخزنة Sp\_Login التي تقوم بجلب البيانات من جدول المستخدمين وذلك عند تطابق البيانات المدخلة في مربعات النصوص مع البيانات الموجودة في الجدول، ثم يتم إسناد نتيجة التابع LOGIN إلى الكائن dt، بعدها يتم الفحص إذا كان عدد أسطر الكائن dt الذي يمثل (جدول البيانات) أكبر من الصفر (أي حصل تطابق بين البيانات المُدخلة والبيانات في الجدول) يتم إظهار رسالة تدل على نجاح عملية الاتصال كما في الشكل (4-32)، وإلا تظهر رسالة خطأ تشير أن كلمة المرور أو اسم المستخدم غير صحيح كما في الشكل (4-33).

الشكل (4-32) ظهور رسالة تدل على نجاح عملية تسجيل الدخول.

الشكل (4-33) ظهور رسالة تدل على فشل عملية تسجيل الدخول.

* يتم ضمن الإجرائية login التحقق إذا كان المستخدم المسجّل للدخول دوره مدير يتم تفعيل الزر الخاص بإدارة المستخدمين وإلا سيتم إلغاء تفعيله، أي لا يمكن إدارة المستخدمين إلا عن طريق المدراء.
* يتم أيضاً في الإجرائية السابقة جلب اسم الشخص الذي سجّل الدخول ودوره عن طريق الكائن dt وذلك لعرضهم في أدوات اللافتة على الواجهة الرئيسية.
* نقوم باستدعاء الإجرائية login ضمن حدث الضغط على زر تسجيل الدخول وضمن حدث مغادرة مربع النص الخاص بكلمة المرور.

**ثانياً: صف الواجهة الرئيسية (FRM\_MAIN)**

الإجرائيات والتوابع والأحداث التي يحتويها هذا الصف هي:

* **الإجرائية (addControlToPanel):** تستخدم هذه الإجرائية لإضافة أدوات الـ user control إلى الحاوية (panelcontrols) الموجودة في الواجهة الرئيسية، حيث نمرر لها بارامتر من نوع أداة تحكم (في مشروعنا هي أداة التحكم user control)، فتقوم أولاً بتحديد موضع وأبعاد الأداة بشكلٍ يتناسب مع حجم الحاوية (ضمنها)، ثم تحذف كل الأدوات الموجودة على الحاوية (panelcontrols) ومن بعدها تضيف الأداة المحددة.
* **الإجرائية (movesidepanel):** تقوم هذه الإجرائية بالتحكم بحركة الحاوية الصغيرة الموجودة بجانب الأزرار في القسم الأيمن من الواجهة الرئيسية، حيث نمرر لها أداة تحكم (زر)، فتقوم هذه الإجرائية أولاً بإظهار الحاوية وجعل ارتفاع وطول هذه الحاوية مثل ارتفاع وطول الزر المحدد.
* **حدث العد الخاص بالمؤقت timer1 (timer1\_Tick):** يتم في البداية فحص قيمة المتحول المنطقي iscollapsed إذا كانت صحيحة (true) يتم زيادة عرض الحاوية اليسارية panelleft بمقدار 10، وتتم الزيادة إلى أن يصبح عرض الحاوية panelleft أكبر أو يساوي عرضها الكلي، فعندها يتم إيقاف المؤقت وجعل قيمة المتحول iscollapsed تساوي الـ false، عندما تكون قيمة المتحول iscollapsed تساوي false يتم إنقاص عرض الحاوية اليسارية panelleft بمقدار10 أيضاً إلى أن تعود إلى موضعها الأصلي، بعدها يتم إرجاع قيمة المتحول iscollapsed إلى true.
* **حدث العدّ الخاص بالمؤقت timertime (timertime\_Tick):** ضمن هذا الحدث يتم إنشاء كائن من الصف DateTime باسم dt، ثم يتم إسناد قيمة الوقت الحالي لهذا الكائن ليتم بعدها عرض قيمة هذا الكائن في أداة اللافتة الموجودة في الواجهة الرئيسية.
* **حدث النقر على زر تسجيل الخروج (btn\_logout\_Click):** عند النقر على هذا الزر يتم إخفاء الواجهة الرئيسية وإظهار واجهة تسجيل الدخول.
* **حدث النقر على زر الإغلاق (btn\_exit\_Click):** عند النقر على هذا الزر يتم إغلاق البرنامج.
* **الباني:** يتم ضمن هذا التابع بدء عمل المؤقت الخاص بالوقت، بالإضافة لإسناد قيمة عرض أداة الحاوية panelleft إلى المتحول panelwidth، وإخفاء الحاوية الصغيرة panelside ومن ثم جعل قيمة المتحول iscollapsed تساوي false، حيث يعمل هذا التابع عند تشغيل البرنامج.
* نقوم باستدعاء الإجرائية movesidepanel ضمن حدث النقر على الزر مع تمرير اسم هذا الزر للإجرائية، ونقوم باستدعاء الإجرائية addControlToPanel أيضاً ضمن هذا الحدث بعد أن نقوم بإنشاء كائن من أداة الـ user control الخاصة بالزر المحدد، حيث نمرر اسم هذا الكائن للإجرائية addControlToPanel، نقوم بعملية الاستدعاء للإجرائيتين داخل أحداث جميع الأزرار الموجودة في القسم الأيمن من الشاشة.

**ثالثاً: صف الـ User Controls الخاص بإدارة الطلاب (US\_Student\_Managament)**

في البداية يتم إنشاء كائن من صف الطالب باسم std، وكائن من صف الاتصال بقاعدة البيانات باسم da لنتمكن من استخدام كافة إجرائياتهم ضمن هذا الصف، وتعريف المتحول المنطقي theDelete والمتحول theDel وإعطائهما القيمة false، يحوي هذا الصف على الإجرائيات والتوابع والأحداث التالية:

* **الباني:** يتم بداخله اختيار القيمة الأولى من أداة الـ combobox الخاصة بنوع الشهادة.
* **حدث تحميل أداة الـ UserControls (US\_Student\_Managment\_Load):** يتم من خلاله وضع مؤشر الكتابة داخل مربع النص الأول "الاسم الأول".
* **إجرائية مسح المحتويات (clear\_std):** تستخدم هذه الإجرائية لمسح محتويات جميع أدوات مربع النص ضمن واجهة إدارة الطلاب، واستبدال قيم أدوات التاريخ بالتاريخ الحالي للنظام، واختيار القيم الأولى لأدوات الـ combobox، ومسح الصورة من أداة Picture، وذلك بهدف التحديث بعد كل عملية إضافة أو تعديل.
* **حدث تغيير الكتابة في مربع النص الخاص بالبحث (txt\_search\_TextChanged):** في البداية يتم فحص إذا كان مربع النص فارغ (أي لم يتم الكتابة ضمنه للبحث) فيتم جلب بيانات جميع الطلاب باستدعاء التابع GET\_All\_Student، أما إذا كان مربع النص غير فارغ (أي تم كتابة اسم الطالب) يتم استدعاء التابع Search\_Student\_name مع تمرير له القيمة المدخلة في مربع النص "اسم الطالب" ليقوم هذا التابع بإرجاع نتيجة البحث أيضاً.
* **حدث النقر المزدوج على أداة dataGridView الخاصة ببيانات الطلاب (dataGridViewAll\_Std\_CellDoubleClick):** بعد النقر المزدوج على خلية معينة يتم فحص قيمة المتحول المنطقي theDelete إذا كانت قيمته false هذا يعني أن العملية ليست حذف وإنما تعديل أي عند النقر المزدوج على أي خلية من خلايا الـ dataGridView يتم جلب جميع البيانات الموجودة في سطر الخلية إلى كافة أدوات مربع النص وأدوات مربع التحرير والسرد وأدوات التاريخ وأداة الصورة (إن لم تكن قيمتها فارغة)، ثم يتم إخفاء الحاوية الخاصة ببيانات الطلاب واستبدال عنوان زر التعديل بالكلمة حفظ، وعنوان زر الإضافة بالكلمة عدم الحفظ، أما إذا كانت قيمة المتحول theDelete تساوي true يتم أولاً تخزين الرقم الجامعي في المتحول id\_stud واسم الطالب الكامل في المتحول fullName ثم تظهر رسالة لتأكيد عملية حذف الطالب المُحدد فعند النقر على زر "نعم" يتم استدعاء إجرائية حذف الطالب delet\_student مع تمرير قيمة المتحول id\_stud لهذه الإجرائية ثم يتم تحديث بيانات الطلاب في أداة الـ dataGridView باستدعاء التابع GET\_All\_Student مجدداً، وأخيراً تظهر رسالة تُخبر المستخدم بنجاح عملية الحذف.
* **حدث الضغط على مفتاح من لوحة المفاتيح (txt\_id\_student\_KeyPress):**

هنا يتم فحص ماهو المفتاح المضغوط، إذا لم يكن رقماً أو مفتاح الحذف يتم تعطيل الكتابة داخل مربع النص الخاص بالرقم الجامعي.

* **حدث النقر الخاص بزر الإضافة (btnadd\_Click):** يتم أولاً فحص إذا كان عنوان الزر هو إضافة أو عدم الحفظ، فإذا كان إضافة يتم استدعاء الإجرائية add\_student مع تمرير القيم المدخلة في مربعات النص وأدوات التاريخ وأدوات مربع التحرير والسرد (التي تمثل بيانات الطالب الشخصية والتفاصيل الجامعية) لها، ويتم استدعاء الإجرائية add\_student\_detile مع تمرير القيم المدخلة في مربعات النص وأدوات التاريخ وأدوات مربع التحرير والسرد (التي تمثل بيانات الطالب التفصيلية ومعلومات التواصل ومعلومات الشهادة الثانوية ومعلومات الطلاب المسجلين تسجيل مباشر) لها، لتقوم الإجرائية add\_student بإرسال قيمها إلى الإجرائية المخزنة add\_student وتقوم الإجرائية add\_student\_detile بإرسال قيمها إلى الإجرائية المخزنة add\_student\_detile لتتم بذلك عملية الإضافة واستدعاء الإجرائية clear\_std لتحديث القيم وتظهر رسالة تخبر المستخدم بنجاح العملية، أما إذا كان عنوان الزر هو عدم الحفظ فيتم وضع عنوان زر التعديل بالكلمة تعديل، وعنوان زر الإضافة بالكلمة إضافة، ويتم إظهار زر الحذف واستدعاء الإجرائية clear\_std لتحديث القيم.
* **حدث النقر الخاص بزر التعديل (btnedit\_Click):** هنا يتم الفحص أيضاً إذا كان عنوان الزر هو تعديل أو حفظ، فإذا كان تعديل يتم إظهار الحاوية الخاصة بالتعديل وإعطاءها حجم وموقع جديدين لتظهر في منتصف الشاشة، ويتم جلب بيانات جميع الطلاب باستدعاء التابع GET\_All\_Student ووضع مؤشر الكتابة ضمن مربع النص الخاص بالبحث ومن ثم إخفاء زر الحذف، أما إذا كان عنوان الزر هو حفظ يتم أولاً تحديد المسار للصورة الجديدة ومن ثم استدعاء التابع chek\_student\_detile والذي يقوم بالتأكد من وجود بيانات تفاصيل الطالب فإذا لم تكن موجودة يتم استدعاء التابع add\_student\_detile لإضافة تفاصيل الطالب، بعد ذلك يتم استدعاء الإجرائية edit\_student مع تمرير لها القيم الجديدة المدخلة في مربعات النص وأدوات التاريخ وأدوات مربع التحرير والسرد (التي تمثل جميع البيانات الخاصة بالطالب بعد تعديلها) لتقوم الإجرائية edit\_student بإرسال هذه القيم إلى الإجرائية المخزنة edit\_student لتتم بذلك عملية التعديل، وبعدها يتم استدعاء الإجرائية clear\_std لتحديث القيم ثم تظهر رسالة تخبر المستخدم بنجاح العملية، ثم يتم وضع عنوان زر التعديل بالكلمة تعديل، وعنوان زر الإضافة بالكلمة إضافة، ويتم إظهار زر الحذف.
* **حدث النقر على زر الحذف (btndel\_Click):** يتم فيه جعل قيمة المتحول المنطقي theDelete تساوي true، وإخفاء زر الحذف، وإظهار الحاوية الخاصة ببيانات الطلاب وإعطائها حجم وموقع جديدين للحاوية لتظهر في منتصف الشاشة وبعد ذلك يتم جلب بيانات جميع الطلاب مجدداً باستدعاء التابع GET\_All\_Student ثم وضع مؤشر الماوس ضمن مربع نص البحث.
* **حدث النقر على زر إلغاء (btn\_cancel\_Click):** عند إلغاء عملية الحذف يتم إخفاء الحاوية وإظهار زر الحذف ومسح محتويات مربع نص البحث وجعل قيمة المتحول المنطقي theDelete تساوي false.
* **حدث النقر على زر قوائم الطلاب (btn\_list\_Click):** عند النقر على هذا الزر يتم إظهار زر إدارة الرسوم وزر تسجيل طالب وزر ترفيع الطلاب الناجحين، يتم أيضاً إخفاء البانل الخاصة بإدارة الرسوم وبانل ترفيع الطلاب الناجحين وبانل بيانات الطلاب وجميع أدوات البانل الظاهرة الخاصة بإدارة بيانات الطلاب، بعد ذلك يتم إظهار أداة datagridview الخاصة ببيانات جميع الطلاب وتعبئتها بالبيانات عن طريق استدعاء التابع GET\_All\_Student، ثم يتم استدعاء الإجرائية list()، وأخيراً يتم إظهار البانل الخاصة بالقوائم والبانل الخاصة بالإضافة.
* **إجرائية قوائم الطلاب التفصيلية (list):** في هذه الإجرائية يتماستنساخ أسطر جديدة من أداة datagridview الخاصة بقوائم الطلاب، ومن ثم تخزين بيانات الطلاب الموجودين في أداة datagridview الخاصة ببيانات جميع الطلاب في هذه الأسطر الجديدة، مع تفصيل بيانات كل طالب (اسم الكلية الخاصة به، اسم السنة الدراسية، نوع الشهادة الثانوية).
* **حدث النقر على زر تسجيل طالب (btn\_cncl\_Click):** عند النقر على هذا الزر يتم إخفاء هذا الزر وإخفاء البانل الخاصة بترفيع الطلاب الناجحين وزر ترفيع الطلاب الناجحين، ويتم إظهار جميع أدوات البانل الخاصة بإدارة بيانات الطلاب وأداة الصورة أيضاً، بعد ذلك يتم إخفاء أداة datagridview الخاصة بإدارة الرسوم، وإظهار زر قائمة الطلاب وبانل الإضافة.
* **حدث تغيير الكتابة الخاص بمربع نص البحث (txt\_serch\_list\_TextChanged):** أولاً يتم فحص مربع نص البحث إذا لم يكن فارغاً يبدأ العمل حيث يتم جلب بيانات جميع الطلاب ووضعها في أداة dataGridViewAll\_Std وذلك بواسطة الإجرائية المخزنة GET\_ALL\_Student التي يتم استدعائها بواسطة التابع GET\_All\_Student، ويتم أيضاً جلب بيانات الرسوم ووضعها في أداة dt\_fees عن طريق التابع get\_all\_fees، أما إذا كان مربع النص فارغاً يتم إستدعاء تابع البحث عن طالب عن طريق الاسم Search\_Student\_name مع تمرير اسم الطالب له، ويتم أيضاً إستدعاء تابع جلب رسوم طالب ما get\_std\_fees مع تمرير اسم الطالب له، وأخيراً يتم استدعاء الإجرائية list().
* **حدث النقر على أداة الصورة (btn\_pic\_Click):** عند النقر يتم إنشاء كائن من الكلاس OpenFileDialog باسم of وهو الكلاس المسؤول عن فتح مستعرض الملفات، ثم يتم بعدها تحديد اللواحق الخاصة (jpg, png, pmb) بالصورة لتظهر الصور فقط، يتم الفحص إن تم الضغط على زر ok يتم تخزين الصورة في أداة الصورة.
* **حدث تغيير اختيار أداة checkbox (checkBox1\_CheckedChanged): هنا يتم اختيار أداة الـ** checkboxإن كانت مختارة يتم زيادة ارتفاع بانل الإضافة لتظهر أداة groupbox الخاصة ببيانات الطلاب المسجلين تسجيلاً مباشراً، وإلا يبقى ارتفاع البانل على حاله.
* **حدث النقر على زر إضافة صورة الشهادة (btn\_openduc\_Click):** يتم أولاً اختيار جميع اللواحق لتظهر جميع الملفات، تم يتم التأكد إن تم الضغط على زر ok يتم تخزين مسار الصورة في المتحول النصي path\_file.
* **حدث النقر على زر إدارة الرسوم (btn\_fees\_Click):** بمجرد النقر على هذا الزر يتم إخفاء datagridview الخاصة بالرسوم وبانل البحث، ثم يتم إظهار البانل الخاصة بإدارة الرسوم، وإخفاء الأزرار وأدوات البانل الأخرى وأداة الصورة، ثم يتم جلب بيانات الرسوم للطلاب باستخدام التابع all\_fees\_with\_student وذلك فقاً للكلية والسنة.
* **حدث النقر على زر إضافة رسوم طالب (btn\_add\_fees\_Click):** بمجرد النقر على هذا الزر يتم إظهار أداة الـ group box الخاصة بإضافة رسوم، وجعل عنوان الزر btn\_add\_f بالعنوان إضافة، وإخفاء زر الإضافة وزر تصدير الرسوم، وإظهار زر التعديل وزر الحذف.
* **حدث النقر على زر تعديل رسوم طالب (btn\_edit\_fees\_Click):** يتم أولاً إظهار dt\_fees ثم جلب رسوم جميع الطلاب باستخدام التابع get\_all\_fees ووضعهم في dt\_fees، ثم يتم إظهار أداة الـ group box الخاصة بإضافة الرسوم، وجعل قيمة المتحول theDel تساوي false، وإخفاء زر التعديل وإظهار زر الإضافة وزر الحذف وزر تصدير الرسوم.
* **حدث النقر على زر حذف رسوم طالب (btn\_del\_fees\_Click):** يتم أولاً إظهار dt\_fees ثم جلب رسوم جميع الطلاب باستخدام التابع get\_all\_fees ووضعهم في dt\_fees، ثم يتم جعل قيمة المتحول theDel تساوي true، بعد ذلك يتم إظهار البانل الخاصة بعرض قوائم الرسوم، ثم يتم إخفاء زر الحذف وزر تصدير الرسوم، وإظهار زر الإضافة وزر التعديل.
* **حدث النقر المزدوج على أداة dataGridView الخاصة ببيانات الطلاب (dataGridViewAll\_Std\_CellDoubleClick):** بعد النقر المزدوج على خلية معينة يتم فحص قيمة المتحول المنطقي theDel إذا كانت قيمته false هذا يعني أن العملية ليست حذف وإنما تعديل أي عند النقر المزدوج على أي خلية من خلايا الـ dataGridView يتم جلب جميع البيانات الموجودة في سطر الخلية إلى كافة أدوات مربع النص وأدوات مربع التحرير والسرد وأداة التاريخ، ثم يتم استبدال عنوان زر إضافة بالكلمة حفظ، أما إذا كانت قيمة المتحول theDel تساوي true يتم أولاً تخزين الرقم الجامعي في المتحول stud\_id واسم الطالب الكامل في المتحول std\_name وسنة الطالب في المتحول stud\_year، ثم تظهر رسالة لتأكيد عملية حذف الطالب المُحدد فعند النقر على زر "نعم" يتم استدعاء إجرائية حذف رسوم طالب delete\_fees مع تمرير رقم الطالب والسنة وسنة الطالب لهذه الإجرائية ثم يتم تحديث بيانات رسوم الطلاب في أداة الـ dataGridView باستدعاء التابع get\_all\_fees مجدداً، وأخيراً تظهر رسالة تُخبر المستخدم بنجاح عملية الحذف.
* **حدث النقر الخاص بزر إضافة رسوم (btn\_add\_f\_Click):** يتم في البداية التأكد من أن مربعات النص الخاصة برقم الطالب الجامعي وبيانات الرسوم غير فارغة، ثم يتم فحص عنوان الزر هل هو إضافة أو حفظ، فإذا كان إضافة يتم استدعاء الإجرائية add\_fees مع تمرير القيم المدخلة في مربعات النص وأدوات التاريخ وأدوات مربع التحرير والسرد (التي تمثل رقم الطالب وبيانات الرسوم) لها، لتقوم الإجرائية add\_fees بإرسال قيمها إلى الإجرائية المخزنة add\_fees لتتم بذلك عملية الإضافة، بعد ذلك يتم استدعاء الإجرائية clear\_fees لتحديث القيم وتظهر رسالة تخبر المستخدم بنجاح العملية، أما إذا كان عنوان الزر هو حفظ فهذا يعني أن العملية هي تعديل فيتم استدعاء إجرائية التعديل edit\_fees مع تمرير القيم لها، ثم يتم استدعاء الإجرائية clear\_fees لتحديث القيم وتظهر رسالة تخبر المستخدم بنجاح العملية .
* **إجرائية مسح محتويات أدوات الرسوم (clear\_fees):** هذه الإجرائية مسؤولة عن مسح محتويات جميع أدوات مربع النص وأدوات السرد وأداة التاريخ وذلك من أجل التحديث وإجراء عمليات جديدة.
* **حدث النقر على زر ترفيع الطلاب** **(btn\_succes\_Click):** عند النقر على هذا الزر تظهر رسالة للمستخدم لتنبهه على استخدام هذا الزر بعد انتهاء العام الدراسي وإدخال كافة الدرجات لجميع الطلاب، بعد ذلك يتم إظهار الأدوات الخاصة بترفيع الطلاب.
* **حدث النقر على زر إيجاد الطلاب الناجحين** **(button2\_Click):** هذا الزر يقوم بإيجاد الطلاب الناجحين في سنة ما وذلك عن طريق التابع success\_year حيث نمرر له (عدد مواد الترفع، سنة الطالب، الكلية)، ويقوم هذا الزر أيضاً بإيجاد الطلاب الراسبين في سنة ما وذلك عن طريق التابع not\_success\_year حيث نمرر له (عدد مواد الترفع، سنة الطالب، الكلية)، تظهر رسالة للمستخدم لتنبهه على استخدام هذا الزر بعد انتهاء العام الدراسي وإدخال كافة الدرجات لجميع الطلاب، بعد ذلك يتم إظهار الأدوات الخاصة بترفيع الطلاب.
* **حدث النقر على زر ترفيع (button3\_Click):** عند النقر على هذا الزر تظهر رسالة للمستخدم لتأكيد العملية وأنه لا يمكن التراجع بعد الضغط على زر yes، عند الضغط على زر yes يتم ترفيع الطلاب عن طريق استدعاء التابع success\_year\_ok مع تمرير القيم (عدد مواد الترفع، سنة الطالب، الكلية) له، ثم تظهر رسالة تخبر المستخدم بنجاح العملية أو فشلها.
* **إجرائية جلب طلاب سنة ما (list\_year):** تقوم هذه الإجرائية باستدعاء التابع Get\_All\_student\_inYear مع تمرير الكلية والسنة له، ليقوم بإرجاع بيانات الطلاب الموافقين لهذه السنة.
* **إجرائية تصدير بيانات الطلاب (export\_student):** تقوم هذه الإجرائية بتصدير بيانات جميع الطلاب التفصيلية الموجودة في أداة dataGridViewAll\_Std إلى ملف إكسل.
* **إجرائية تصدير بيانات الرسوم (all\_fees):** تقوم هذه الإجرائية بتصدير بيانات رسوم جميع الطلاب الموجودة في أداة dt\_fees إلى ملف إكسل.
* **إجرائية تصدير بيانات رسوم طالب (all\_fees\_with\_student):** تقوم هذه الإجرائية بتصدير بيانات رسوم طالب واحد في جميع السنوات إلى ملف إكسل.

**رابعاً: صف الـ User Controls الخاص بإدخال الدرجات (US\_Degree\_Insert)**

في البداية يتم إنشاء كائن من صف الدرجة باسم deg، لنتمكن من استخدام كافة إجرائياته ضمن هذا الصف، ويحوي هذا الصف على الإجرائيات والتوابع والأحداث التالية:

* **الباني:** يتم داخله إعطاء قيم ابتدائية (كلية المعلوماتية- السنة الأولى- الفصل الأول) لأدوات مربع التحرير والسرد Combobox الخاصة ببيانات المادة، ثم يتم استدعاء التابع Get\_All\_Material مع تمرير له قيم أدوات الـ Combobox لجلب مواد الكلية والسنة والفصل المحددات، ثم يتم وضع مؤشر الكتابة داخل مربع النص الخاص بالرقم الجامعي للطالب.
* **حدث النقر على زر الإضافة (btnadd\_Click):** يتم أولاً فحص إذا كان عنوان الزر هو إضافة أو عدم الحفظ، فإذا كان إضافة يتم استدعاء الإجرائية insert\_degree مع تمرير لها القيم المدخلة في مربعات النص وأدوات التاريخ وأدوات مربع التحرير والسرد وأداة التاريخ (التي تمثل بيانات الدرجة)، لتقوم الإجرائية insert\_degree بإرسال هذه القيم إلى الإجرائية المخزنة insert\_degree لتتم بذلك عملية الإضافة وظهور رسالة تخبر المستخدم بنجاح العملية، ويتم تحديث قيم مربعات النص وذلك بمسح محتوى مربع النص الخاص بالرقم الجامعي وجعل قيمة مربعي النص الخاصين بدرجة العملي ودرجة النظري تساويان الصفر، أما إذا كان عنوان الزر هو عدم الحفظ فيتم وضع عنوان زر التعديل بالكلمة تعديل، وعنوان زر الإضافة بالكلمة إضافة، وإظهار زر الحذف ومن ثم استدعاء الإجرائية clear\_deg لتحديث القيم.
* **إجرائية مسح المحتويات (clear\_deg):** تقوم بتحديث القيم وذلك بمسح محتوى مربع النص الخاص بالرقم الجامعي وجعل قيمة مربعي النص الخاصين بدرجة العملي ودرجة النظري تساويان الصفر، ومسح محتوى مربع النص الخاص برقم المادة، وإسناد قيمة التاريخ الحالي لنظام التشغيل إلى أداة التاريخ الخاصة بتقديم المادة ومسح محتوى مربع النص الخاص برقم المدرس ثم إسناد القيمة (1) إلى هذا المربع.
* **حدث النقر المزدوج على أداة dataGridView الخاصة ببيانات الدرجة (dataGridViewAll\_Deg\_CellDoubleClick):**

بعد النقر المزدوج على خلية معينة يتم فحص قيمة المتحول المنطقي theDelete إذا كانت قيمته false معناها أن العملية ليست حذف وإنما تعديل أي عند النقر المزدوج على أي خلية من خلايا الـ dataGridView، يتم جلب جميع البيانات الموجودة في سطر الخلية إلى أدوات مربع النص وأدوات مربع التحرير والسرد وأداة التاريخ ثم يتم إخفاء الحاوية الخاصة ببيانات الدرجة واستبدال عنوان زر التعديل بالكلمة حفظ وعنوان زر الإضافة بالكلمة عدم الحفظ، أما إذا كانت قيمة المتحول theDelete تساوي true يتم أولاً تخزين الرقم الجامعي في المتحول id\_stud وتخزين رمز المادة المراد حذقف درجتها في المتحول id\_mat ثم تظهر رسالة لتأكيد عملية حذف الدرجة فعند النقر على زر "نعم" يتم استدعاء إجرائية حذف الدرجة delet\_degree مع تمرير قيمة المتحول id\_stud والمتحول id\_mat لهذه الإجرائية والتي بدورها تحذف الدرجة من قاعدة البيانات بولسطة الإجرائية المخزنة delete\_degree ثم يتم تحديث البيانات في أداة الـ dataGridView باستدعاء التابع GET\_All\_DEGREE مجدداً وذلك بواسطة الإجرائية المخزنة Get\_All\_Degree فتظهر بعد ذلك رسالة تُخبر المستخدم بنجاح عملية الحذف، ثم يتم مسح محتوى مربع نص البحث ووضع مؤشر الماوس فيه، أما عند الضغط على زر "لا" يتم إخفاء الحاوية وجعل قيمة المتحول theDelete تساوي false وإظهار زر الحذف ومسح محتوى مربع نص البحث.

* **حدث النقر على أداة dataGridView الخاصة ببيانات المادة (dataGridView\_mat\_CellClick):** عند النقر على أي خلية من خلايا الـ dataGridView يتم جلب رقم المادة إلى مربع النص الخاص بها.
* **حدث تغيير الكتابة في مربع النص الخاص بالبحث (txt\_search\_TextChanged):** في البداية يتم الفحص إذا كان مربع النص فارغ (أي لم يتم الكتابة ضمنه للبحث)، فيتم جلب بيانات جميع الدرجات باستدعاء التابع GET\_All\_DEGREE، أما إذا كان مربع النص غير فارغ (أي تم كتابة رقم جامعي أو رقم المادة)، فيتم استدعاء التابع Search\_Degree وتمرير النص المكتوب داخل مربع النص لهذا التابع فيقوم بإرجاع نتيجة البحث وذلك بواسطة الإجرائية المخزنة Search\_Degree.
* **حدث تغيير القيمة المختارة لأداة الـ Combobox الخاصة بالكلية (cmbfaculty\_SelectedIndexChanged):** عند تغيير القيمة المختارة (أي تغيير الكلية)، يتم جلب أسماء المواد وأرقامها للكلية المحددة وتخزينها في أداة الـ dataGridView الخاصة بالمواد وذلك بواسطة الإجرائية المخزنة [Get\_All\_Material]، يتم تنفيذ هذه التعليمة داخل الحدث ذاته لكل أدوات الـمربع السرد والتحرير Combobox الخاصة بمعلومات المادة (السنة، الفصل).
* **حدث الضغط على مفتاح من لوحة المفاتيح txtid\_KeyPress)):** هذا الحدث يقوم بالتحقق إذا كانت القيم في مربع النص (الخاص بالرقم الجامعي) المقابلة للمفاتيح التي تم الضغط عليها ليست أرقام أو ليست قيمة مفتاح الحذف (Backspace) فيقوم بمنع عملية الكتابة ضمن مربع النص لأن الرقم الجامعي لا يمكن أن يكون حرف أو رمز.
* **حدث تحميل أداة الـ User Controls (US\_Degree\_Insert\_Load):** يتم من خلاله وضع مؤشر الكتابة داخل مربع النص الأول (الرقم الجامعي).
* **حدث النقر الخاص بزر التعديل (btnedit\_Click):** هنا يتم الفحص أيضاً إذا كان عنوان الزر هو تعديل أو حفظ، فإذا كان تعديل يتم إظهار الحاوية الخاصة بالتعديل وإعطاءها حجم وموقع جديدين لتظهر في منتصف الشاشة، ومن ثم وضع مؤشر الكتابة في مربع نص البحث، ويتم جلب بيانات جميع الدرجات باستدعاء التابع GET\_All\_DEGREE وذلك بواسطة الإجرائية المخزنة Get\_All\_Degree وثم إخفاء زر الحذف، أما إذا كان عنوان الزر هو حفظ يتم استدعاء الإجرائية edit\_degree مع تمرير لها القيم الجديدة المدخلة في مربعات النص وأداة التاريخ وأداة مربع التحرير والسرد (التي تمثل بيانات الدرجة بعد تعديلها) لتقوم الإجرائية edit\_degree بإرسال هذه القيم إلى الإجرائية المخزنة edit\_degree لتتم بذلك عملية التعديل، وبعدها يتم استدعاء الإجرائية clear\_deg لتحديث القيم ثم تظهر رسالة تخبر المستخدم بنجاح العملية، ثم يتم وضع عنوان زر التعديل بالكلمة تعديل، وعنوان زر الإضافة بالكلمة إضافة وإظهار زر الحذف.
* **إجرائية message\_error()** : وهي إجرائية خاصة برسائل الأخطاء والتوجيه التي تظهر للمستخدم أثناء استخدام البرنامج فتظهر رسالة واضحة تشرح ما هو الخطأ أو التحذير أو توجيه معين لمساعدة المستخدم في العمل على البرنامج بسهولة .
* **حدث النقر على زر الحذف (btndel\_Click):** يتم فيه جعل قيمة المتحول المنطقي theDelete تساوي true، وإخفاء زر الحذف، وإظهار الحاوية الخاصة بالدرجات وإعطائها حجم وموقع جديدين للحاوية لتظهر في منتصف الشاشة وبعد ذلك يتم جلب بيانات جميع الدرجات مجدداً باستدعاء التابع GET\_All\_DEGREE ثم وضع مؤشر الماوس ضمن مربع نص البحث.
* **حدث النقر على زر إلغاء (btncancel\_Click):** عند النقر على زر إلغاء عملية التعديل أو عملية الحذف يتم إخفاء الحاوية وإظهار زر الحذف ومسح محتويات مربع نص البحث وجعل قيمة المتحول theDelete تساوي false.
* **حدث تغيير الكتابة في مربع النص الخاص باسم الطالب(txtstdname\_TextChanged)** أولاً يتم فحص مربع النص الخاص باسم الطالب إذا لم يكن فارغ يبدأ العمل حيث يقوم بجلب الرقم الجامعي للطالب ووضعه في dataGridView1 وذلك بواسطة الإجرائية المخزنة Get\_stdnum وبعد ذلك يتم وضع الرقم الجامعي للطالب في حقل مربع النص المخصص له، أي جلب الرقم الجامعي للطالب عن طريق اسمه.
* **حدث تغيير الكتابة لمربع النص الخاص بالرقم الجامعي (txtid\_TextChanged):** يقوم بجلب اسم الطالب عند تغير الكتابة بمربع النص الخاص برقم الطالب الجامعي حيث يجلب الاسم عند كتابة آخر ثلاثة أرقام من الرقم الجامعي بواسطة الإجرائية المخزنة Search\_by3id ثم وضع اسم الطالب في حقل مربع النص المخصص له.
* **إجرائية جلب أسماء الطلاب في الكلية والسنة المحددتين get\_names():** تعتمد هذه الإجرائية على كل من المتحولين faculty , year في تحديد الكلية والسنة لجلب أسماء الطلاب فيها وذلك بواسط الإجرائية المخزنة get\_all\_name\_in\_year ووضع الأسماء مع أرقامهم الجامعية في أداة dataGridView\_lot .
* **حدث تغيير القيمة المختارة لأداة الـ Combobox الخاصة بالكلية (cmb\_year\_students\_SelectedIndexChanged):** عند تغيير القيمة المختارة للسنة يتم استدعاء الإجرائية **get\_names()** التي تقوم بجلب أسماء الطلاب مع أرقامهم الجامعية وفق الكلية والسنة المحددتين .
* **حدث النقر على زر طالب واحد (btn\_one\_Click):** عند النقر على زر طالب واحد يتم إخفاء panel\_lot الخاصة بإدخال الدرجات لعدة طلاب ومن ثم إظهار panel\_one الخاصة بإدخال الدرجات لطالب واحد بعد إعطاؤها حجم وموقع جديدين.
* **حدث النقر على زر عدة طلاب (btn\_lot\_Click):** عند النقر على زر عدة طلاب يتم إخفاء panel\_one الخاصة بإدخال الدرجات لطالب واحدوثم إظهار panel\_lot الخاصة بإدخال الدرجات لعدة طلاببعد إعطاؤها حجم وموقع جديدين.
* **حدث تغيير القيمة المختارة لأداة الـ Combobox الخاصة بالكلية (cmb\_year\_students\_SelectedIndexChanged\_1):** عند تغيير القيمة المختارة للسنة يتم استدعاء الإجرائية **get\_names()** التي تقوم بجلب أسماء الطلاب مع أرقامهم الجامعية وفق الكلية والسنة المحددتين.
* **حدث النقر على زر إضافة (btn\_insertlot\_Click):** عند النقر على زر إضافة يتم فحص هل الزر btn\_exel مفعل أم لا فإذا كان غير مفعل يتم إضافة الدرجات من dataGridView\_lot إلى قاعدة البيانات عن طريق التابع insert\_degree وذلك بواسطة الإجرائية المخزنة DEGREE\_SCORE أما اذا كان مفعل فيتك إضافة الدرجات من ملف إيكسل إلى قاعدة البيانات، ويظهر رسالة تفيد بنجاح عملية الإدخال.
* **إجرائية ConvertExcelToDataTable():** يتم استخدام هذه الإجرائية من أجل تحويل البيانات من ملف إيكسل إلى بيانات ضمن جدول خاص بقاعدة البيانات من نوعDataTable
* **حدث النقر على زر إدخال يدوي (btn\_manule\_Click):** عند الضغط **زر** إدخال يدوي يتم إلغاء تفعيل النقر على الزر ذاته ويتم تفعيل الزر btn\_exelوإظهار dataGridView\_lot\_e.
* **حدث النقر على زر استيراد ملف من exel (btn\_exel\_Click):** عند الضغط على هذا الزر فسوف تظهر قائمة استعراض ملفات الكمبيوتر وذلك بهدف اختيار ملف الإكسل المراد استيراد الدرجات منه وبعد اختيار الملف وتحديده يتم نقل البيانات من ملف الإكسل الى dataGridView\_lot\_e**.**
* **حدث النقر على زر ايكسل (button1\_Click\_1):** عند الضغط على الزر إكسل يتم إلغاء تفعيل الزر ذاته وتفعيل زر إدخال يدوي وإظهار dataGridView\_lot\_e بعد إعطاؤها حجم وموقع جديدين.
* **حدث تغيير الكتابة لمربع النص رقم مدرس المادة** (**txt\_teach\_TextChanged):** في نافذة إدخال علامات لطالب واحد وعند تغيير الكتابة الموجودة في مربع نص رقم مدرس المادة يقوم التابع get\_techer\_name\_idبجلب اسم المدرس وذلك بواسطة الإجرائية المخزنة get\_techer\_name\_idوبعد ذلك يتم وضع اسم المدرس في مربع النص المخصص له.
* **حدث تغيير الكتابة لمربع النص رقم مدرس المادة (txt\_teacher\_TextChanged):** في نافذة إدخال علامات عدة طلاب وعند تغيير الكتابة الموجودة في مربع نص رقم مدرس المادة يقوم التابع get\_techer\_name\_idبجلب اسم المدرس وذلك بواسطة الإجرائية المخزنة get\_techer\_name\_idوبعد ذلك يتم وضع اسم المدرس في مربع النص المخصص له.
* **حدث تغيير الكتابة لمربع النص (txt\_teach\_namel\_TextChanged) اسم مدرس المادة:** في نافذة إدخال علامات لطالب واحد وعند تغيير الكتابة الموجودة في مربع نص اسم مدرس المادة يقوم التابع get\_techer\_name\_idبجلب رقم المدرس وذلك بواسطة الإجرائية المخزنة get\_techer\_name\_idوبعد ذلك يتم وضع رقم المدرس في مربع النص المخصص له.
* **حدث تغيير الكتابة لمربع النص (txt\_teacher\_namel\_TextChanged) اسم مدرس المادة:** في نافذة إدخال علامات لعدة طلاب وعند تغيير الكتابة الموجودة في مربع نص اسم مدرس المادة يقوم التابع get\_techer\_name\_idبجلب رقم المدرس وذلك بواسطة الإجرائية المخزنة get\_techer\_name\_idوبعد ذلك يتم وضع رقم المدرس في مربع النص المخصص له.

**خامساً: صف الـ User Controls الخاص بإدارة الدرجات (US\_Dergree\_Management):**

في البداية يتم إنشاء كائن من صف الدرجة باسم cost، وكائن من صف الطالب باسم std، لنتمكن من استخدام كافة إجرائياتهم ضمن هذا الصف، يحوي هذا الصف على الأحداث التالية:

* **حدث تغيير الكتابة لمربع النص الخاص بالرقم الجامعي (txtid\_TextChanged):** عند التغيير في حقل مربع النص للرقم الجامعي فيتم جلب اسم الطالب عن طريق آخر ثلاثة أرقام من رقمه الجامعي , بعد ذلك يتم فحص مربع النص txt\_idfull فإذا كان فارغ يتم إعطاء حجم جديد لكل من panel\_score & panel\_newspaperواستدعاء الإجرائية clear()، ثم يتم استعاء كل من الإجرائيتين score\_n() & score() ويتم فحص مربع النص الرقم الجامعي إذا كان فارغ يتم إلغاء تفعيل الأزرار ( زر كشف العلامات button9 وزر صحيفة طالب button8)، وأيضا يتم فحص مربع النص txt\_idfull إذا لم يكن فارغ يتم تفعيل الزرين السابقين أما ذا كان فارغ فيتم إلغاء تفعيلهما.
* **إجرائية score():** أولاً يتم فحص مربع النص الخاص بالرقم الجامعي إذا لم يكن فارغاً يبدأ العمل حيث يتم في البداية إخفاء أدوات الـ groupbox الخمسة الخاصة بالسنوات الدراسية ثم يتم استدعاء التابع GET\_DETILE\_STUD الذي يقوم بجلب تفاصيل الطالب (بياناته الشخصية) بعد أن نمرر له الرقم الجامعي للطالب بإدخاله في مربع النص، ثم يتم إسناد هذه البيانات إلى أداة الـ dataGridView الخاصة بتفاصيل الطالب (dataGridView\_detils)، بعد ذلك يتم فحص أداة dataGridView\_detils إذا كانت لا تحوي أسطر يتم إظهار رسالة للمستخدم تقول أنه لا يوجد طالب بهذا الرقم، أما إذا كانت تحوي أسطر فيتم جلب قيمة الخلية الخامسة منها التي تمثل السنة الدراسية للطالب (أولى، ثانية، ثالثة، ....) وتخزين هذه القيمة في المتحول الرقمي x الذي يتم فحصه إذا كانت قيمته أكبر أو تساوي واحد (أي أن الطالب هو سنة أولى) يتم إظهار الـ groupbox الخاصة بالسنة الأولى، يتم مقارنة قيمة المتحول x مع القيم 2 و3 و4 و5 إذا كانت قيمة x أكبر أو تساوي هذه الأرقام يتم إظهار الـ groupbox المقابلة لهذه الأرقام، بعد ذلك يتم إسناد قيم خلايا أداة dataGridView\_detils التسعة إلى مربعات النص الخاصة بتفاصيل الطالب، يتم بعدها جلب درجات كل سنة من السنوات باستدعاء التابع GET\_DEGREE مع تمرير له الرقم الجامعي للطالب المراد جلب درجاته والسنة الدراسية له والفصل الدراسي بعد جلب الدرجات يتم تخزينها في أداة الـ dataGridView المقابلة لكل فصل ويتم جلب معدل الطالب في السنة الموجود في الخلية الأولى من أداة dataGridView\_detils بعد التأكد من أن هذه الأداة تحوي بيانات (الدرجات) فيتم تخزين هذا المعدل في مربع النص الخاص بالمعدل، أما إذا كانت أداة dataGridView\_detils لا تحوي درجات يتم وضع القيمة صفر في مربع النص الخاص بالمعدل، يتم تكرار هذه العملية لأجل كل السنوات الدراسية مع تغيير البارامترات (الرقم الجامعي والسنة والفصل)، في النهاية يتم إظهار كلمة ok للدلالة على نجاح العملية.
* **إجرائية clear():** يتم استخدام هذه الإجرائية عندما يتم إفراغ مربع النص الخاص بالرقم الجامعي ومهمة هذه الإجرائية هي تفريغ الأدوات الموجودة في الـ groupBoxالتي كانت تحوي علامات الطلاب.
* **حدث النقر على زر إخراج إلى ملف pdf (BtnExport\_Click):** بعد جلب درجات الطالب في جميع سنواته الدراسية ومعدله في كل سنة، بالإضافة إلى بياناته الشخصية (تفاصيله) أصبح كشف الدرجات جاهزاً، يقوم هذا الزر عند النقر عليه بإخراج بيانات هذا الكشف إلى ملف Excel مُصمم مسبقاً ليمكّن المستخدم (موظف شؤون الامتحانات) من التعامل معه بسهوله وإمكانية تعديله وتحويله إلى صيغ مختلف (pdf مثلاً) وإمكانية طباعته.
* يتم داخل هذا الزر في البداية جلب مسار ملف الإكسل الذي سيتم إخراج بيانات الكشف إليه وتخزين المسار في المتحول النصي xx وجلب اسم المجلد الذي يحتوي ملف الإكسل الموجود ضمن المسار ذاته، ويُخزن اسم المجلد في المتحول النصي path2 (يُمثّل مسار الملف واسم مجلده)، بعد ذلك يُحول مؤشر الماوس إلى شكل دائرة للدلالة على أن البرنامج في حالة تنفيذ، بعد ذلك يتم إنشاء كائن من الصف Excel.Application باسم app1 لنتمكن من التعامل مع ملفات الإكسل، وكائن من الصف Excel.Workbook باسم work1 الذي يقوم بفتح ملف الإكسل الموجود في المسار path2 وذو الاسم والامتداد ".xlsxكشف علامات"، وكائن من الصف Excel.Worksheet باسم x لتحديد الورقة الأولى في ملف الإكسل التي سيتم التخزين فيها، بعد ذلك يتم جلب بيانات الطالب (تفاصيله) من مربعات النص وتخزينها في الخلايا المقابلة في ملف الإكسل.
* الخطوة التالية هي نسخ أسماء المواد ودرجات هذه المواد إلى ملف الإكسل، حيث يتم أولاً التأكد أن عدد أسطر أداة dataGridView الخاصة بكل فصل لا تساوي الصفر أي تحتوي على بيانات فإذا تحقق الشرط يتم جلب أسماء المواد من الخلية الأولى من كل سطر من أسطر أداة dataGridView الخاصة بالفصل، ثم تُخزن هذه الأسماء في الخلايا المقابلة في ملف الإكسل، بعد جلب أسماء المواد يتم جلب درجات هذه المواد من الخلية الرابعة من كل سطر من أسطر أداة dataGridView الخاصة بالفصل، ثم تُخزن هذه الدرجات في الخلايا المقابلة في ملف الإكسل مع فحص درجات النظري إذا كانت أقل من 60 (أي المادة راسبة).
* الطالب الذي لديه علامة عملي فقط يتم إظهار هذه العلامة في الخلية مع عبارة "عملي فقط"، بعد ذلك يتم جلب معدل الطالب في السنة من مربع نص المعدل ووضعه في الخلية المقابلة في ملف الإكسل، يتم تكرار هذه العمليات من أجل كل فصل في السنوات الخمس فينتج كشف علامات بصيغة ملف إكسل.
* **الخطوة التالية يتم فيها إنشاء مجلد اسمه (**أرشيف كشف العلامات) **داخل مجلد (المستندات) في الكمبيوتر، فيتم تحويل كشف العلامات من صيغة ملف إيكسل إلى ملف** pdf **وتخزينه ضمن مجلد أرشيف كشف العلامات، وحذف ملف كشف العلامات الذي هو بصيغة ملف إكسل.**
* **حدث تحميل أداة الـ User Control الخاصة بكشف الدرجات (US\_Dergree\_score\_Load):** عند تنفيذ البرنامج يتم وضع مؤشر الكتابة في مربع النص الخاص بالرقم الجامعي.
* **حدث النقر على زر إصدار العلامات (button7\_Click):**

عند النقر على هذا الزر يتم إخفاء (panel\_exam\_record) وإظهار المجموعة الخاصة بإصدار العلامات (groupBox\_mat\_year).

* **حدث النقر على زر صحيفة الطالب (button8\_Click):**
* عند الضغط على هذا الزر يتم تحويل لون الزر (button8) الخاص بصحيفة الطالب إلى اللون الأخضر ويتم عرض الصفحة الخاصة (panel\_newspaper) بصحيفة الطالب الطالب وإخفاء الصفحة (panel\_score) الخاصة بكشف علامات بعد إدخال الرقم الجامعي للطالب في المربع الخاص ويم جلب يبانات الطالب بالإضافة إلى علاماته وبيانات خاصة بالدورة التي قدمت قيها المادة كذلك العام، أولاً يتم فحص مربع النص الخاص بالرقم الجامعي إذا لم يكن فارغ يبدأ العمل حيث يتم في البداية إخفاء أدوات الـ groupbox الخمسة الخاصة بالسنوات الدراسية ثم يتم استدعاء التابع GET\_DETILE\_STUD الذي يقوم بجلب تفاصيل الطالب (بياناته الشخصية) بعد أن نمرر له الرقم الجامعي للطالب بإدخاله في مربع النص، ثم يتم إسناد هذه البيانات إلى أداة الـ dataGridView الخاصة بتفاصيل الطالب (dataGridView\_detils)، بعد ذلك يتم فحص أداة dataGridView\_detils إذا كانت لا تحوي أسطر يتم إظهار رسالة للمستخدم تقول أنه لا يوجد طالب بهذا الرقم، أما إذا كانت تحوي أسطر فيتم جلب قيمة الخلية الخامسة منها التي تمثل السنة الدراسية للطالب (أولى، ثانية، ثالثة، ....) وتخزين هذه القيمة في المتحول الرقمي x الذي يتم فحصه إذا كانت قيمته أكبر أو تساوي واحد (أي أن الطالب هو سنة أولى) يتم إظهار الـ groupbox الخاصة بالسنة الأولى، يتم مقارنة قيمة المتحول x مع القيم 2 و3 و4 و5 إذا كانت قيمة x أكبر أو تساوي هذه الأرقام يتم إظهار الـ groupbox المقابلة لهذه الأرقام، بعد ذلك يتم إسناد قيم خلايا أداة dataGridView\_detils التسعة إلى مربعات النص الخاصة بتفاصيل الطالب، يتم بعدها جلب درجات كل سنة من السنوات باستدعاء التابع GET\_DEGREE مع تمرير له الرقم الجامعي للطالب المراد جلب درجاته والسنة الدراسية له والفصل الدراسي بعد جلب الدرجات يتم تخزينها في أداة الـ dataGridView المقابلة لكل فصل ويتم جلب معدل الطالب في السنة الموجود في الخلية الأولى من أداة dataGridView\_detils بعد التأكد من أن هذه الأداة تحوي بيانات (الدرجات) فيتم تخزين هذا المعدل في مربع النص الخاص بالمعدل، أما إذا كانت أداة dataGridView\_detils لا تحوي درجات يتم وضع القيمة صفر في مربع النص الخاص بالمعدل، يتم تكرار هذه العملية لأجل كل السنوات الدراسية مع تغيير البارامترات (الرقم الجامعي والسنة والفصل)، في النهاية. وكذلك يتم عرض معدل كل سنة وإظهار زر اخراج ملف pdf حيث يتم أولاً تصدير الصفحة إلى ملف اكسيل ثم توليد منه ملف pdf وحذف ملف الاكسل.
* **حدث النقر على زر إصدار علامات العملي button5\_Click:** عند الضغط على زر إصدار علامات العملي يتم البدء بعملية التصدير وهي تحويل العلامات من جدول الداتاغرادفيو إلى ملف من نوع PDF**.**
* **إجرائية degree\_particla:**

ومهمة هذه الإجرائية هي تصدير علامات العملي إلى ملف pdf وتعمل على الشكل التالي: حيث يتم جلب ملف الإيكسل الفارغ الخاص بإصدار علامات العملي وتعبئة الخلايا العلوية منه بالبيانات الخاصة بالمادة مثل اسم المادة ورمزها والعام الدراسي الذي تم تقديم المادة فيه والسنة الدراسية , وبعد ذلك يتم البدء بنقل العلامات من dataGridView\_mat\_year إلى خلايا ملف الإيكسل وتم استخدام البرمجة التفرعية في هذه العملية من أجل تسريعها , وهكذا يكون قد تم إنشاء ملف الإيكسل لعلامات العملي، ثم يتم إنشاء مجلد اسمه (أرشيف علامات المواد) داخل مجلد المستندات في الكمبيوتر من أجل تخزين ملفات العلامات فيه، ثم يتم تحويل ملف الإيكسل إلى ملف pdf وتخزينه في المستندات وحذف ملف الإيكسل وإظهار رسالة تفيد بنجاح عملية التصدير.

* **حدث النقر على زر إصدار علامات النظري button10\_Click\_1:** عند الضغط على زر إصدار العلامات النهائية يتم البدء بعملية التصدير وهي تحويل العلامات من جدول الداتاغرادفيو إلى ملف من نوع PDF**.**
* **إجرائية degree\_final:**

ومهمة هذه الإجرائية هي تصدير علامات النظري إلى ملف pdf وتعمل على الشكل التالي: حيث يتم جلب ملف الإيكسل الفارغ الخاص بإصدار علامات النظري وتعبئة الخلايا العلوية منه بالبيانات الخاصة بالمادة مثل اسم المادة ورمزها والعام الدراسي الذي تم تقديم المادة فيه والسنة الدراسية بالإضافة إلى كل من نسبة النجاح ونسبة الرسوب، وبعد ذلك يتم البدء بنقل العلامات من dataGridView\_mat\_year إلى خلايا ملف الإيكسل وتم استخدام البرمجة التفرعية في هذه العملية من أجل تسريعها، وهكذا يكون قد تم إنشاء ملف الإيكسل لعلامات النظري، ثم يتم في أول عملية تصدير إنشاء مجلد اسمه (أرشيف علامات المواد) داخل مجلد المستندات في الكمبيوتر أول من أجل تخزين ملفات العلامات فيه، وفي عمليات التصدير الأخرى يتم تخزين ملفات العلامات في نفس المجاد ولا حاجة لإنشاء مجلد من جديد، ثم يتم تحويل ملف الإيكسل إلى ملف pdf وتخزينه في المستندات وحذف ملف الإكسل وإظهار رسالة تفيد بنجاح عملية التصدير وفتح ملف الـ pdf.

* **حدث تغيير الكتابة في مربع نص العشرة الأوائل (txt\_top10\_TextChanged):** عند كتابة رقم سنة من السنوات الستة يتم جلب قائمة الطلاب العشرة الأوائل عن هذه السنة (الاسم، الرقم الجامعي، المعدل) عن طريق التابع Get\_Top10 الذي يقوم بتنفيذ الإجرائية المخزنة Get\_Top10 في قاعدة البيانات.
* **إجرائية تصدير الطلاب العشرة الأوائل (export\_top10):** باستخدام هذه الإجرائية يتم تصدير بيانات الطلاب العشرة الأوئل الموجودة في أداة dataGridView\_top10 إلى ملف إكسل، ثم تحويله وعرضه بصيغة PDF.
* **إجرائية السجل السنوي للسنة الأولى (exam\_record1):** تقوم هذه الإجرائية بإصدار السجل الامتحاني السنوي للسنة الأولى وذلك بهدف حفظ نسخة من الدرجات، يتم جلب جميع الدرجات لجميع الطلاب المتقدمين من السنة الأولى في السنة الدراسية الحالية عن طريق الإجرائية المخزنة exam\_record التي نمرر لها (السنة وسنة التقديم)، ثم يتم وضع هذه الدرجات في الأداة dt\_exam\_rec، بعد ذلك يتم نسخ هذه الدرجات ووضعها في الورقة الخاصة بالسنة الأولى ضمن ملف الإكسل "سجل سنوي معلوماتية"، ثم يتم تصدير الملف بصيغة PDF.
* **إجرائية السجل السنوي للسنة الثانية (exam\_record2):** تقوم هذه الإجرائية بإصدار السجل الامتحاني السنوي للسنة الثانية وذلك بهدف حفظ نسخة من الدرجات، يتم جلب جميع الدرجات لجميع الطلاب المتقدمين من السنة الثانية في السنة الدراسية الحالية عن طريق الإجرائية المخزنة exam\_record التي نمرر لها (السنة وسنة التقديم)، ثم وضع هذه الدرجات في الأداة dt\_exam\_rec، بعد ذلك يتم نسخ هذه الدرجات ووضعها في الورقة الخاصة بالسنة الثانية ضمن ملف الإكسل "سجل سنوي معلوماتية"، ثم يتم تصدير الملف بصيغة PDF.
* **إجرائية السجل السنوي للسنة الثالثة (exam\_record3):** تقوم هذه الإجرائية بإصدار السجل الامتحاني السنوي للسنة الثالثة وذلك بهدف حفظ نسخة من الدرجات، يتم جلب جميع الدرجات لجميع الطلاب المتقدمين من السنة الثالثة في السنة الدراسية الحالية عن طريق الإجرائية المخزنة exam\_record التي نمرر لها (السنة وسنة التقديم)، ثم وضع هذه الدرجات في الأداة dt\_exam\_rec، بعد ذلك يتم نسخ هذه الدرجات ووضعها في الورقة الخاصة بالسنة الثالثة ضمن ملف الإكسل "سجل سنوي معلوماتية"، ثم يتم تصدير الملف بصيغة PDF.
* **إجرائية السجل السنوي للسنة الرابعة (exam\_record4):** تقوم هذه الإجرائية بإصدار السجل الامتحاني السنوي للسنة الرابعة وذلك بهدف حفظ نسخة من الدرجات، يتم جلب جميع الدرجات لجميع الطلاب المتقدمين من السنة الرابعة في السنة الدراسية الحالية عن طريق الإجرائية المخزنة exam\_record التي نمرر لها (السنة وسنة التقديم)، ثم وضع هذه الدرجات في الأداة dt\_exam\_rec، بعد ذلك يتم نسخ هذه الدرجات ووضعها في الورقة الخاصة بالسنة الرابعة ضمن ملف الإكسل "سجل سنوي معلوماتية"، ثم يتم تصدير الملف بصيغة PDF.
* **إجرائية السجل السنوي للسنة الخامسة (exam\_record5):** تقوم هذه الإجرائية بإصدار السجل الامتحاني السنوي للسنة الخامسة وذلك بهدف حفظ نسخة من الدرجات، يتم جلب جميع الدرجات لجميع الطلاب المتقدمين من السنة الخامسة في السنة الدراسية الحالية عن طريق الإجرائية المخزنة exam\_record التي نمرر لها (السنة وسنة التقديم)، ثم وضع هذه الدرجات في الأداة dt\_exam\_rec، بعد ذلك يتم نسخ هذه الدرجات ووضعها في الورقة الخاصة بالسنة الخامسة ضمن ملف الإكسل "سجل سنوي معلوماتية"، ثم يتم تصدير الملف بصيغة PDF.
* **إجرائية إحصائيات المادة (exam\_statistics):** هذه الإجرائية تقوم بجلب إحصائيات المادة المقدمة (عدد المتقدمين، عدد الناجحين، عدد الراسبين، نسبة النجاح والرسوب) عن طريق التابع exam\_takers الذي نمرر له (السنة وعام التقديم)، ثم يتم وضع هذه القيم في أدوات مربع النص الخاصة بها.
* **إجرائية مسح إحصائيات المادة (clear\_statistics):** تستخدم هذه الإجرائية لمسح محتويات أدوات مربع النص الخاصة بالإحصائيات بعد الانتهاء من العمل.

**سادساً: صف الـ User Controls الخاص بالنسخ الاحتياطي (US\_Backup)**

أولاً يتم إنشاء كائن من صف تسجيل الدخول باسم log، وكائن من صف الدرجة باسم dg وكائن من صف الطالب باسم std.

يحتوي هذا الصف على الاحداث التالية:

* **حدث النقر على زر استعراض (btnBrowse\_Click):** عند النقر على هذا الزر تظهر نافذة لاختيار المجلد الذي نرغب بحفظ النسخة الاحتياطية فيه، ثم بعد اختيار المجلد والضغط على زر "ok" يتم نسخ المسار إلى مربع النص الخاص بمكان النسخة الاحتياطية.
* **حدث النقر على زر إنشاء (btn\_Create\_Click):** بالنقر على هذا الزر يتم فحص مربع النص الخاص بمكان النسخة الاحتياطية إذا كان فارغ سيتم إنشاء نسخة احتياطية للمسار الافتراضي "المستندات/أرشيف/ ملفات قاعدة البيانات الاحتياطية"، أما إذا لم يكن مربع النص فارغ يتم تحويل مؤشر الماوس إلى شكل دائرة للدلالة على أنه يتم التنفيذ، ثم يتم دمج المسار الموجود في مربع النص مع اسم النسخة الاحتياطية AleppoFreeUniversity، وإضافة التاريخ الحالي في نهاية الاسم لإظهار تاريخ إنشاء النسخة، وتخزين هذه الأسماء في المتحول النصي filename، ثم بعد ذلك يُكتب الأمر الخاص بالنسخ الاحتياطي ودمجه مع المتحول filename (اسم النسخة) ومع الامتداد bak، ومن ثم تخزينهم في المتحول النصي strquery، يتم بعدها تمرير المتحول strquery والكائن con إلى الكائن cmd ليتم تنفيذه باستخدام الإجرائية ExecuteNonQuery بعد فتح الاتصال، ثم إغلاقه بعد انتهاء التنفيذ، في النهاية يتم إظهار رسالة تدل على نجاح عملية النسخ الاحتياطي ويتم إفراغ محتوى مربع النص وإرجاع مؤشر الماوس إلى الشكل الافتراضي.
* **حدث النقر على زر قائمة الطلاب (btn\_list\_Click):** بالنقر على هذا الزر يتم إظهار الأداة dataGridViewAll\_Std على الشاشة ويتم تعبئتها ببيانات الطلاب عن طريق التابع GET\_All\_Student() من الكائن std , ويتم تعبئة المتحول النصي backup بالقيمة GET\_All\_Student"" .
* **حدث النقر على زر قائمة الدرجات (btn\_list\_degrees\_Click):** بالنقر على هذا الزر يتم إظهار الأداة dataGridViewAll\_Std على الشاشة ويتم تعبئتها ببيانات الدرجات عن طريق التابع backup\_degree() من الكائن dg , ويتم تعبئة المتحول النصي backup بالقيمة backup\_degree"" .
* **حدث النقر على زر قائمة المواد (btn\_list\_mat\_Click):** بالنقر على هذا الزر يتم إظهار الأداة dataGridViewAll\_Std على الشاشة ويتم تعبئتها ببيانات المواد بإرسال الكلية من قائمة cmbfaculty عن طريق التابع all\_mat(cmbfaculty.SelectedIndex+1) من الكائن dg , ويتم تعبئة المتحول النصي backup بالقيمة backup\_materials"" .
* **حدث النقر على زر قائمة العلامات والطلاب (btn\_list\_stu\_degree\_Click):** بالنقر على هذا الزر يتم إظهار الأداة dataGridViewAll\_Std على الشاشة ويتم تعبئتها ببيانات العلامات والطلاب بإرسال الكلية من قائمة cmbfaculty عن طريق التابع all\_degree\_with\_student(cmbfaculty.SelectedIndex+1) من الكائن dg , ويتم تعبئة المتحول النصي backup بالقيمة all\_degree\_with\_student"" .
* **حدث النقر على زر تصدير البيانات إلى ملف إكسل (BtnExport\_n\_Click):** بالنقر على هذا الزر يتم تعطيل كافة أزرار البرنامج واستدعاء أداة العمل في الخلفية backgroundWorker1 وتشغيلها عن طريق الأمر backgroundWorker1.RunWorkerAsync(); .
* أداة backgroundWorker1 لها ثلاث أحداث:

حدث التشغيل (backgroundWorker1\_DoWork): يقوم بفحص المتحول النصي backup في حال كانت "GET\_All\_Student" يتم استدعاء التابع exam\_record1(); الذي يقوم بتعبئة البيانات في ملف الاكسل بيانات الطلاب الاحتياطية.xlcx ويتم استعمال المعالجة التفرعية لتسريع العملية.

اذا كان المتحول النصي backup في حال كانت "backup\_degree" يتم استدعاء التابع backup\_degree1(); الذي يقوم بتعبئة البيانات في ملف الاكسل بيانات العلامات الاحتياطية.xlcx ويتم استعمال المعالجة التفرعية لتسريع العملية.

اذا كان المتحول النصي backup في حال كانت "backup\_materials" يتم استدعاء التابع backup\_materials(); الذي يقوم بتعبئة البيانات في ملف الاكسل بيانات المواد الاحتياطية.xlcx ويتم استعمال المعالجة التفرعية لتسريع العملية.

اذا كان المتحول النصي backup في حال كانت "all\_degree\_with\_student" يتم استدعاء التابع all\_degree\_with\_student1(); الذي يقوم بتعبئة البيانات في ملف الاكسل قائمة العلامات والطلاب.xlcx ويتم استعمال المعالجة التفرعية لتسريع العملية.

يتم اظهار تقدم العملية عبر الأداة progressBar1.

عند انتهاء حدث التشغيل يتم استدعاء حدث الانتهاء (backgroundWorker1\_RunWorkerCompleted): يتم إعادة تفعيل كافة أزرار البرنامج.

**سابعاً: صف الـ User Controls الخاص باستعادة النسخة الاحتياطية (US\_Restore)**

أولاً يتم إنشاء كائن من صف تسجيل الدخول باسم log، وكائن من صف الدرجة باسم dg وكائن من صف الطالب باسم std.

يحوي هذا الصف أيضاً على الاحداث التالية:

* **حدث النقر على زر استعراض (btnBrowse\_Click):** عند النقر على هذا الزر تظهر نافذة لاختيار ملف النسخة الاحتياطية التي قمنا بإنشائها سابقاً، ثم بعد اختيار الملف والضغط على زر ok"" يتم نسخ اسم الملف والمسار المخزن به إلى مربع النص الخاص باسم ومسار النسخة الاحتياطية.
* **حدث النقر على زر استعادة (Btn\_Restore\_Click):** بالنقر على هذا الزر يتم فحص مربع النص الخاص باسم ومسار النسخة الاحتياطية إذا كان فارغ سيتم إظهار رسالة تنبيه للمستخدم وتطلب منه اختيار مكان تواجد ملف النسخة الاحتياطية، أما إذا لم يكن مربع النص فارغ يتم تحويل مؤشر الماوس إلى شكل دائرة للدلالة على أنه يتم التنفيذ، ثم بعد ذلك يُكتب الأمر الخاص باسترجاع ملف النسخة الاحتياطية ودمجه مع قيمة مربع النص (اسم ومكان النسخة) log.Restore\_database(txt\_Restore.Text)، في النهاية يتم إظهار رسالة تدل على نجاح عملية استعادة النسخة الاحتياطية ويتم إفراغ محتوى مربع النص وإرجاع مؤشر الماوس إلى الشكل الافتراضي.

**الخلاصة**

الخلاصة هي أننا وصلنا إلى الأهداف التي ننشدها من هذا المشروع، وتم بناء وتصميم البرنامج بنجاح، حيث يمكن الآن أن يستفيد موظفي شؤون الامتحانات من هذا البرنامج، والذي يقدم الخدمات التالية:

* إدارة الطلاب والتي تتمثل بإدخال بيانات الطلاب المستجدين والتعديل على بيانات الطلاب المدخلين سابقاً، وإدارة الرسوم الجامعية.
* ترفيع الطلاب الناجحين إلى السنة التالية.
* إدخال الدرجات (درجات المواد العملية والنظرية) لطالب واحد أو لعدة طلاب وإمكانية التعديل على البيانات.
* إصدار كشف درجات لطالب محدد بضغطة زر.
* إصدار صحيفة لطالب محدد بضغطة زر.
* إصدار السجل الامتحاني السنوي.
* جلب الطلاب العشرة الأوائل.
* إدارة المدرسين والمواد.
* إنشاء نسخة احتياطية لقاعدة البيانات.
* استعادة النسخة الاحتياطية.

**المقترحات والتوصيات**

فيما يلي بعض المقترحات التي يمكن أن تزيد من قوة المشروع، ومن أجل تطويره مستقبلاً:

* زيادة الأمن والحماية على قاعدة البيانات وعلى البرنامج باستخدام خوازرميات التشفير الشهيرة.
* رفع قاعدة البيانات على الإنترنت حيث يمكن للمستخدمين الاتصال بها من أكثر من جهاز حاسب ومن أي مكان.
* إنشاء موقع ويب خاص للبرنامج وربطه بقاعدة البيانات مرفوعة على الإنترنت حيث يمكن للمستخدمين التحكم المباشر عن طريق الإنترنت.
* تطبيق أندرويد للاتصال والدخول إلى البرنامج عن بُعد لاسلكياً.

**المراجع**

1. خالد الشقروني، (2006-2004)” تطبيق UML“.

(29/11/2020).

1. حسن م. الفحل، (2019)” # Cمن البداية حتى الإتقان“، ط1.

(7/12/2019).

1. خالد السعداني،” الشرح الوافي لتعلم لغة sql من نبعها الصافي“.

(19/1/2019).

1. <https://docs.microsoft.com/en-us/sql/sql-server/?view=sqlallproducts-allversions>

(9/4/2020).

1. <https://docs.microsoft.com/en-us/dotnet/csharp/>

(10/5/2020).

**الملاحق**

**أكواد قاعدة البيانات**

* **الإجرائيات المخزنة:**
* **إجرائية إضافة الرسوم(add\_fees)**

CREATE proc [dbo].[add\_fees]

@student\_id nvarchar(15),

@full\_year\_fees int,

@fees\_paid float,

@does\_paid bit,

@discount\_rate float,

@amount\_of\_discount float,

@year\_fees\_after\_discount float,

@reason\_for\_discount nvarchar(20),

@old\_fees\_unpaid float,

@academic\_year int,

@stady\_year nvarchar(20),

@voucher\_number nvarchar(20),

@date\_paid date,

@notes nvarchar(50)

as

begin

INSERT INTO [dbo].[Student\_fees]

([student\_id]

,[full\_year\_fees]

,[fees\_paid]

,[does\_paid]

,[discount\_rate]

,[amount\_of\_discount]

,[year\_fees\_after\_discount]

,[reason\_for\_discount]

,[old\_fees\_unpaid]

,[academic\_year]

,[stady\_year]

,[voucher\_number]

,[date\_paid]

,[notes])

VALUES

( @student\_id,

@full\_year\_fees ,

@fees\_paid ,

@does\_paid ,

@discount\_rate ,

@amount\_of\_discount ,

@year\_fees\_after\_discount ,

@reason\_for\_discount,

@old\_fees\_unpaid ,

@academic\_year ,

@stady\_year,

@voucher\_number ,

@date\_paid ,

@notes )

end

* **إجرائية إضافة مادة (add\_material)**

create proc [dbo].[add\_material]

@id nvarchar(12),

@mat\_name nvarchar(30),

@fac int,

@year int,

@seas\_mat bit

as

begin

insert into Material

(id,

material\_name,

faculty,

[year],

season\_mat

)

values

(@id,

@mat\_name,

@fac,

@year,

@seas\_mat

)

end

* **إجرائية إضافة طالب (add\_student)**

create proc [dbo].[add\_student]

@id int،

@first\_name nvarchar(20)،

@last\_name nvarchar(20)،

@father nvarchar(20)،

@mother nvarchar(20)،

@gender bit،

@birth\_city nvarchar(20)،

@birth\_date date،

@box\_Number nvarchar(20)،

@faculty int،

@year int،

@tel nvarchar(20)،

@Certificate\_Sec int،

@Certifcate\_Source nvarchar(50)،

@Certificate\_date date،

@yearOfAdmission date،

@typeOfAdmission bit،

@Department nvarchar(20)

as

begin

insert into Student

(id،FName،LName،Father،Mother،gender،birth\_city،birth\_date،box\_Number،faculty،[year]،

telephon،Certificate\_Sec،Certifcate\_Source،Certificate\_date،yearOfAdmission،typeOfAdmission،Department)

values (@id،@first\_name،@last\_name،@father،@mother،@gender،@birth\_city،@birth\_date،@box\_Number،@faculty،@year،@tel،@Certificate\_Sec،@Certifcate\_Source،@Certificate\_date،@yearOfAdmission،@typeOfAdmission،@Department)

end

* **إجرائية إضافة تفاصيل الطالب (add\_student\_detile)**

CREATE proc [dbo].[add\_student\_detile]

@id nvarchar(15),

@pictur image,

@section int,

@second\_language nvarchar(15),

@Matrial\_status bit,

@personal\_ID nvarchar(15),

@husband\_name nvarchar(15),

@husband\_work nvarchar(15),

@Children\_Number int,

@Children\_age nvarchar(15),

@migration bit,

@migration\_date date,

@father\_work nvarchar(15),

@Health\_status nvarchar(50),

@personal\_notes nvarchar(50),

@email nvarchar(50),

@address\_original nvarchar(50),

@address\_current nvarchar(50),

@Certifcate\_state nvarchar(15),

@Certifcate\_id nvarchar(50),

@Certifcate\_total int,

@Certifcate\_total\_document varbinary(max),

@old\_universirt nvarchar(25),

@old\_faculty nvarchar(25),

@old\_Department nvarchar(25),

@old\_yearOfAdmission date,

@date\_ofStop date,

@old\_yearStudy int,

@Old\_documents\_note nvarchar(50),

@Old\_documents varbinary(max),

@old\_Institute nvarchar(25),

@old\_Institute\_department nvarchar(15),

@old\_Institute\_date date,

@old\_Institute\_seson int,

@old\_Institute\_total int,

@old\_Institute\_rating nvarchar(15)

as

begin

insert into Student\_detile

(id,pictur,section,second\_language,Matrial\_status,personal\_ID,husband\_name,husband\_work,Children\_Number,Children\_age,migration,

migration\_date,father\_work,Health\_status,personal\_notes,email,address\_original,address\_current,Certifcate\_state,Certifcate\_id,

Certifcate\_total,Certifcate\_total\_document,old\_universirt,old\_faculty,old\_Department,old\_yearOfAdmission,date\_ofStop,old\_yearStudy,

Old\_documents\_note,Old\_documents,old\_Institute,old\_Institute\_department,old\_Institute\_date,old\_Institute\_seson,old\_Institute\_total,old\_Institute\_rating)

values (@id,@pictur,@section,@second\_language,@Matrial\_status,@personal\_ID,@husband\_name,@husband\_work,@Children\_Number,@Children\_age,@migration,

@migration\_date,@father\_work,@Health\_status,@personal\_notes,@email,@address\_original,@address\_current,@Certifcate\_state,@Certifcate\_id,

@Certifcate\_total,@Certifcate\_total\_document,@old\_universirt,@old\_faculty,@old\_Department,@old\_yearOfAdmission,@date\_ofStop,@old\_yearStudy,

@Old\_documents\_note,@Old\_documents,@old\_Institute,@old\_Institute\_department,@old\_Institute\_date,@old\_Institute\_seson,@old\_Institute\_total,@old\_Institute\_rating)

end

* **إجرائية ضافة معلم (add\_teacher)**

create proc [dbo].[add\_teacher]

@id int,

@teach\_name nvarchar(20),

@scien\_ran int,

@num\_of\_mat int

as

begin

insert into Teacher

(id,

teacher\_name,

scientific\_rank,

num\_of\_material

)

values

(@id,

@teach\_name,

@scien\_ran,

@num\_of\_mat

)

end

* **إجرائية عرض علامات جميع الطلاب (all\_degree\_with\_student)**

CREATE proc [dbo].[all\_degree\_with\_student]

@faculty int

as

select student\_id as'الرقم\_الجامعي',FName+' '+LName as 'الاسم',Material.material\_name as 'المادة',degree\_practical as 'العملي'

,degree\_theoretical as 'النظري',degree\_final as 'النهائية'

into ##degree

from Degree

join Material

on Material.id=Degree.material\_id

join Student on Student.id=Degree.student\_id

where Material.faculty=@faculty

-------------------------------

select \* into ##deg

from

(

select الرقم\_الجامعي ,الاسم,المادة,النهائية

from ##degree

)

as SourceTable PIVOT(AVG(النهائية) FOR المادة IN(

[الثقافة والحضارة],[لغة أجنبية /1/],عربي,[مبادئ عمل الحواسيب],[برمجة/1/],[فيزياء/1/],[تحليل/1/],[لغة أجنبية/2/],[دارات كهربائية],[برمجة/2/],[فيزياء/2/],[تحليل/2/],[رياضيات متقطعة],[جبر خطي],[إلكترونيات],[برمجة/3/],[لغة أجنبية تخصصية/1/],[احتمالات],[تحليل/3/],[تحليل عددي/1/],[برمجة رياضية],[نظم ودارات منطقية],[خوارزميات وبنى المعطيات/1/],[إحصاء],[تحليل/4/],[تحليل عددي/2/],[مهارات تواصل],[لغة أجنبية تخصصية /2/],[نظرية المخططات],[معالج مصغر],[معالجة الإشارة],[نظرية المعلومات],[خوارزميات وبنى المعطيات/2/],[رسوميات حاسوبية],[لغات صورية],[مبادئ الذكاء الصنعي],[بنية وتنظيم الحواسيب/1/],[شبكات حاسوبية],[اتصالات تشابهية ورقمية],[هندسة البرمجيات/1/],[خوارزميات وبنى المعطيات/3/],[قواعد معطيات/1/],[بحوث عمليات],[نظم تشغيل/1/],[بنية وتنظيم الحواسيب/2/],[النظم الرقمية المبرمجة],[شبكات حاسوبية متقدمة],[أمن معلومات],[قواعد معطيات/2/],[مشروع],[روبوتية],[نظم تشغيل/2/],[البرمجة التفرعية],[نظم وسائط متعددة],[التسويق وإدارة المشاريع],[تصميم مترجمات],[النمذجة والمحاكاة],[نظم خبيرة],[PLC التحكم المنطقي المبرمج],[أمن الشبكات],[نظم الزمن الحقيقي],[الشبكات اللاسلكية],[جودة ووثوقية],[إدارة النظم الإنتاجية],[تنقيب المعطيات],[مشروع تخرج]

)) AS PivotTable;

select \* from ##deg

order by الرقم\_الجامعي

drop table ##degree

drop table ##deg

* **إجرائية عرض مواد إحدى الكليات (all\_mat)**

CREATE proc [dbo].[all\_mat]

@faculty int

as

select ROW\_NUMBER() OVER(ORDER BY id ASC) as' ',id,material\_name,faculty,year,season\_mat from Material

where faculty=@faculty

order by year,season\_mat,id

* **إجرائية نسخ احتياطي (Backup\_database)**

CREATE proc [dbo].[Backup\_database]

@path nvarchar(75)

as

Backup Database AleppoFreeUniversity to Disk=@path

* **(backup\_degree)**

create proc [dbo].[backup\_degree]

as

select ROW\_NUMBER() OVER(ORDER BY Student.id ASC) as' ',

student\_id as'الرقم الجامعي',FName+' '+LName as 'الاسم',Material.material\_name as 'المادة',degree\_practical as 'العملي'

,degree\_theoretical as 'النظري',degree\_final as 'النهائية',season as 'الدورة',year\_ofDegree as'التاريخ'

from Degree

join Material

on Material.id=Degree.material\_id

join Student on Student.id=Degree.student\_id

order by student\_id,material\_id

* **إجرائية التأكد من وجود طالب (chek\_student\_detile)**

CREATE proc [dbo].[chek\_student\_detile]

@id nvarchar(20)

as

select \* from Student\_detile

WHERE id=@id

* **إجرائية كشف الدرجات (DEGREE\_SCORE)**

create PROCEDURE [dbo].[DEGREE\_SCORE]

@id int ، @year int ،@seson bit

AS

BEGIN

SET NOCOUNT ON;

select distinct material.material\_name،degree\_practical،degree\_theoretical،degree\_final from Degree

join Student on Student.id=Degree.student\_id

join Material on material.id =Degree.material\_id

where Student.id=@id and material.[year]=@year and material.season\_mat=@seson

END

* **إجرائية حذف الدرجة (delete\_degree)**

create proc [dbo].[delete\_degree]

@id int

as

delete from [degree]

where student\_id=@id

* **إجرائية حذف رسوم الطالب (delete\_fees)**

CREATE PROCEDURE [dbo].[delete\_fees]

@student\_id nvarchar(15),

@academic\_year int,

@stady\_year nvarchar(20)

AS

BEGIN

DELETE FROM [dbo].[Student\_fees]

WHERE [student\_id] = @student\_id and [academic\_year] =@academic\_year and [stady\_year] =@stady\_year

end

* **إجرائية حذف مادة (delete\_material)**

create proc [dbo].[delete\_material]

@id nvarchar(12)

as

DELETE FROM Material

WHERE id=@id

* **إجرائية حذف الطالب (delete\_student)**

create PROCEDURE [dbo].[delete\_student]

@id int

AS

BEGIN

delete from [degree]

where student\_id=@id

DELETE FROM [dbo].[Student]

WHERE @id=id

END

* **إجلاائية حذف معلم (delete\_teacher)**

create proc [dbo].[delete\_teacher]

@id int

as

DELETE FROM teacher

WHERE id=@id

* **إجرائية تعديل الدرجة (edit\_degree)**

create proc [dbo].[edit\_degree]

@std\_id int،

@mat\_id int،

@deg\_prac int،

@deg\_theo int،

@season int،

@year\_deg date

as

begin

update Degree

set

degree\_practical=@deg\_prac

،degree\_theoretical=@deg\_theo

،season=@season

،year\_ofDegree=@year\_deg

where student\_id=@std\_id and material\_id=@mat\_id

end

* **إجرائية تعديل الدرجة (edit\_degree)**

CREATE proc [dbo].[edit\_degree\_prac]

@std\_id nvarchar(20),

@mat\_id nvarchar(20),

@deg\_theo int,

@season int,

@year\_deg date,

@teach\_id int

as

begin

update Degree

set

degree\_theoretical=@deg\_theo

,season=@season

,year\_ofDegree=@year\_deg

,teacher\_id=@teach\_id

where student\_id=@std\_id and material\_id=@mat\_id

end

* **إجرائية تعديل رسوم طالب (edit\_fess)**

CREATE proc [dbo].[edit\_fess]

@student\_id nvarchar(15),

@full\_year\_fees int,

@fees\_paid float,

@does\_paid bit,

@discount\_rate float,

@amount\_of\_discount float,

@year\_fees\_after\_discount float,

@reason\_for\_discount nvarchar(20),

@old\_fees\_unpaid float,

@academic\_year int,

@stady\_year nvarchar(20),

@voucher\_number nvarchar(20),

@date\_paid date,

@notes nvarchar(50)

as

begin

UPDATE [dbo].[Student\_fees]

SET [student\_id] = @student\_id

,[full\_year\_fees] =@full\_year\_fees

,[fees\_paid] =@fees\_paid

,[does\_paid] =@does\_paid

,[discount\_rate] =@discount\_rate

,[amount\_of\_discount] =@amount\_of\_discount

,[year\_fees\_after\_discount] =@year\_fees\_after\_discount

,[reason\_for\_discount] =@reason\_for\_discount

,[old\_fees\_unpaid] =@old\_fees\_unpaid

,[academic\_year] =@academic\_year

,[stady\_year] =@stady\_year

,[voucher\_number] =@voucher\_number

,[date\_paid] =@date\_paid

,[notes] =@notes

WHERE [student\_id] = @student\_id and ([academic\_year] =@academic\_year or [voucher\_number] =@voucher\_number) and ([stady\_year] =@stady\_year or [voucher\_number] =@voucher\_number)

end

* **إجرائية تعديل مادة (edit\_material)**

CREATE proc [dbo].[edit\_material]

@mat\_id nvarchar(12),

@mat\_name nvarchar(30),

@fac int,

@year int,

@seas\_mat bit

as

begin

update Material

set

material\_name=@mat\_name,

faculty=@fac,

[year]=@year,

season\_mat=@seas\_mat

where id=@mat\_id

end

* **إجرائية تعديل الطالب (edit\_student)**

create proc [dbo].[edit\_student]

@id int،

@first\_name nvarchar(20)،

@last\_name nvarchar(20)،

@father nvarchar(20)،

@mother nvarchar(20)،

@gender bit،

@birth\_city nvarchar(20)،

@birth\_date date،

@box\_Number nvarchar(20)،

@faculty int،

@year int،

@tel nvarchar(20)،

@Certificate\_Sec int،

@Certifcate\_Source nvarchar(50)،

@Certificate\_date date،

@yearOfAdmission date،

@typeOfAdmission bit،

@Department nvarchar(20)

as

begin

update Student

set

FName=@first\_name

،LName=@last\_name

،Father=@father

،Mother=@mother

،gender=@gender

،birth\_city=@birth\_city

،birth\_date=@birth\_date

،box\_Number=@box\_Number

،faculty=@faculty

،[year]=@year

،telephon=@tel

،Certificate\_Sec=@Certificate\_Sec

،Certifcate\_Source=@Certifcate\_Source

،Certificate\_date=@Certificate\_date

،yearOfAdmission=@yearOfAdmission

،typeOfAdmission=@typeOfAdmission

،Department=@Department

where id=@id

end

* **إجرائية تعديل معلم (edit\_teacher)**

create proc [dbo].[edit\_teacher]

@id int,

@teach\_name nvarchar(20),

@scien\_ran int,

@num\_of\_mat int

as

begin

update teacher

set

id=@id

,teacher\_name=@teach\_name

,scientific\_rank=@scien\_ran

,num\_of\_material=@num\_of\_mat

where id=@id

end

* **إجرائية تعديل بيانات مستخدم (edit\_user)**

create proc [dbo].[edit\_user]

@Unameold nvarchar(50)،

@Unamenew nvarchar(50)،

@pass nvarchar(50)،

@name nvarchar(50)،

@usertype nvarchar(50)

as

UPDATE [User]

SET UserName=@Unamenew

،[Password] = @pass

،usertype= @usertype

،[name] = @name

WHERE [UserName]=@Unameold

* **إجرائية السجل الإمتحاني (exam\_record)**

CREATE PROCEDURE [dbo].[exam\_record]

@year\_mat int, @year\_deg int --,@seson bit

AS

BEGIN

select distinct Student.FName+' '+Student.LName as 'الاسم الكامل' ,Student.Father as 'اسم الأب' ,Student.id as 'الرقم الجامعي',Student.year as 'سنة الطالب',material.material\_name as 'المادة',Material.year as 'سنة المادة',degree\_practical as 'علامة العملي',degree\_theoretical as 'علامة النظري' ,degree\_final as 'العلامة النهائية'

from Degree

join Student on Student.id=Degree.student\_id

join Material on material.id =Degree.material\_id

where material.[year]=@year\_mat and DATEPART(YEAR, Degree.year\_ofDegree)=@year\_deg

order by Material.year,Student.year

END

* **إجرائية عرض عدد المواد الراسبة (exam\_takers)**

CREATE proc [dbo].[exam\_takers]

@mat nvarchar(10) ,@year int

as

select count(Student.id) as 'عدد الطلاب المتقدمين' from Degree

join Student on Student.id=Degree.student\_id

where material\_id=@mat and DATEPART(YEAR, Degree.year\_ofDegree)=@year

* **إجرائية عرض المواد الراسبة (fail\_degree)**

CREATE proc [dbo].[fail\_degree]

@std int

as

(select Material.material\_name,Degree.degree\_practical,Degree.degree\_final

from Degree join Material on Material.id=Degree.material\_id

where degree\_final<60 and student\_id=@std)

* **إجرائية جلب عدد الطلاب الراسبين (fail\_num)**

CREATE proc [dbo].[fail\_num]

@mat nvarchar(10) ,@year int

as

select count(Student.id) as 'عدد الطلاب الراسبين' from Degree

join Student on Student.id=Degree.student\_id

where material\_id=@mat and DATEPART(YEAR, Degree.year\_ofDegree)=@year and Degree.degree\_final<60

* **إجرائية جلب الدرجات (Get\_All\_Degree)**

CREATE proc [dbo].[Get\_All\_Degree]

as

begin

select

[student\_id] as 'الرقم الجامعي'

,[material\_name] as 'اسم المادة'

,[material\_id]as ' رمز المادة'

,[degree\_practical] as ' درجة العملي'

,[degree\_theoretical] as ' درجة النظري'

,[degree\_final] as ' الدرجة النهائية'

,[season] as ' الفصل'

,[year\_ofDegree] as ' سنة المادة'

,[teacher\_name] as 'مدرس المادة'

from Degree join Material

ON Material.id=Degree.material\_id

join Teacher

on Degree.teacher\_id=Teacher.id

end

* **إجرائية جلب الرسوم (get\_all\_fees)**

CREATE proc [dbo].[get\_all\_fees]

as

SELECT [Student\_fees].[student\_id] as 'الرقم الجامعي'

,Student.FName+' '+Student.LName as 'اسم الطالب'

,[full\_year\_fees] as 'الرسم السنوي'

,[fees\_paid] as 'الرسم المدفوع'

,[does\_paid] as 'هل دفع هذا العام'

,[discount\_rate] as 'نسبة الحسم'

,[amount\_of\_discount] as 'مقدار الحسم'

,[year\_fees\_after\_discount] as 'الرسم بعد الحسم'

,[reason\_for\_discount] as 'سبب الحسم'

,[old\_fees\_unpaid] as 'الرسم المتبقي على الطالب'

,[academic\_year] as 'السنة الدراسية'

,[stady\_year] as 'العام الدراسي'

,[voucher\_number] as 'رقم الايصال'

,[date\_paid] as 'تاربخ الدفع'

,[notes] as 'ملاحظات'

from Student\_fees join Student

on [Student\_fees].student\_id=Student.id

* **إجرائية جلب معلومات المواد (Get\_All\_Material)**

CREATE PROCEDURE [dbo].[Get\_All\_Material]

@faculty int,@year int ,@seson bit

AS

BEGIN

select id as 'رمز المادة',material\_name as 'اسم المادة' from Material

where faculty=@faculty and year=@year and season\_mat=@seson

END

* **إجرائية عرض أسماء طلاب السنة (get\_all\_name\_in\_year)**

CREATE proc [dbo].[get\_all\_name\_in\_year]

@faculty int ,@year int

as

select ROW\_NUMBER() OVER(ORDER BY id ASC) as' ', id as 'الرقم الجامعي',FName+' '+LName as'الاسم' from Student

where faculty=@faculty and year=@year

* **إجرائية جلب بيانات الطلاب (Get\_All\_student)**

CREATE proc [dbo].[Get\_All\_student]

as

begin

select ROW\_NUMBER() OVER(ORDER BY Student.id ASC) as' ',

Student.id as 'الرقم الجامعي'

,[FName] as 'الاسم '

,[LName] as 'الكنية'

,[Father] as 'الأب'

,[Mother] as 'الأم'

,[gender] as 'الجنس'

,[birth\_city] as 'مكان الميلاد'

,[birth\_date] as 'تاريخ الميلاد'

,[box\_Number] as 'القيد'

,Student.faculty as 'الكلية'

,[year] as 'السنة'

,[telephon]as 'رقم الهاتف'

,[Certificate\_Sec]as 'نوع الشهادة'

,[Certifcate\_Source]as 'مصدر الشهادة'

,[Certificate\_date]as 'تاريخ الشهادة'

,[yearOfAdmission]as 'سنة القبول'

,[typeOfAdmission]as 'نوع القبول'

,[Department]as 'القسم'

,pictur as'الصورة'

,[section]as'الشعبة'

,[second\_language]as'اللغة الأجنبية'

,[Matrial\_status]as'الوضع العائلي'

,[personal\_ID]as'الرقم الوطني'

,[husband\_name]as'اسم الزوج/ـة'

,[husband\_work]as'عمل الزوج/ـة'

,[Children\_Number]as'عدد الأولاد'

,[Children\_age]as'أعمار الأولاد'

,[migration]as'مهجر'

,[migration\_date]as'تاريخ التهجير'

,[father\_work]as'عمل الأب'

,[Health\_status]as'الحالة الصحية'

,[personal\_notes]as'معلومات شخصية إضافية'

,[email]as'البريد الإلكتروني'

,[address\_original]as'العنوان الأصلي'

,[address\_current]as'العنوان الحالي'

,[Certifcate\_state]as'المحافظة المانحة للشهادة'

,[Certifcate\_id]as'رقم الاكتتاب'

,[Certifcate\_total]as'المعدل'

-- ,[Certifcate\_total\_document]

,[old\_universirt]as'الجامعة السابقة'

,[old\_faculty]as'الكلية السابقة'

,[old\_Department]as'القسم السابق'

,[old\_yearOfAdmission]as'عام التسجيل في الجامعة'

,[date\_ofStop]as'عام الانقطاع'

,[old\_yearStudy]as'السنة الدراسية السابقة'

,[Old\_documents\_note]as'الوثائق المقدمة'

-- ,[Old\_documents]

,[old\_Institute]as'المعهد السابق'

,[old\_Institute\_department]as'القسم'

,[old\_Institute\_date]as'العام'

,[old\_Institute\_seson]as'الدورة الفصلية'

,[old\_Institute\_total]as'معدل التخرج'

,[old\_Institute\_rating]as'التقدير'

from [dbo].[Student]

left join Student\_detile on Student.id=Student\_detile.id

end

* **إجرائية جلب بيانات المعلمين (Get\_all\_teacher)**

CREATE proc [dbo].[Get\_all\_teacher]

as

select id as 'رقم المدرس'

,teacher\_name as 'اسم المدرس'

, scientific\_rank as 'الرتبة العلمية'

,num\_of\_material as 'عدد المواد'

from teacher

* **إجرائية جلب بيانات المستخدمين (Get\_All\_Users)**

create proc [dbo].[Get\_All\_Users]

as

SELECT \* FROM [User]

* **إجرائية جلب المعدل (GET\_AVG\_DEGREE)**

CREATE PROCEDURE [dbo].[GET\_AVG\_DEGREE]

@id nvarchar(20), @year int

AS

BEGIN

select round( (CONVERT(float, sum(degree\_final))/(count(degree\_final))),2) from Degree

join Material on Material.id=Degree.material\_id

where Material.year=@year and student\_id=@id and Degree.degree\_final>=60

group by student\_id

end

* **إجرائية جلب درجة طالب في فصل وعام ()**

create proc [dbo].[get\_degreesWithSeson]

@id nvarchar(20) , @year int ,@seson bit

AS

BEGIN

SET NOCOUNT ON;

-- Insert statements for procedure here'

select distinct material.material\_name,degree\_practical,degree\_theoretical,degree\_final, Degree.season,RIGHT ( year(Degree.year\_ofDegree),2) as 'العام الدراسي'

from Degree

join Student on Student.id=Degree.student\_id

join Material on material.id =Degree.material\_id

where Student.id=@id and material.[year]=@year and material.season\_mat=@seson

END

* **إجرائية (get\_mat\_year)**

CREATE proc [dbo].[get\_mat\_year]

@mat nvarchar(10) ,@year int

as

select ROW\_NUMBER() OVER(ORDER BY id ASC) as' ',Student.id as 'الرقم الجامعي', Student.FName+' '+LName as 'الاسم',Student.Father as 'اسم الأب',degree\_practical as ' العملي',degree\_theoretical as 'النظري',degree\_final as'العلامة' from Degree

join Student on Student.id=Degree.student\_id

where material\_id=@mat and DATEPART(YEAR, Degree.year\_ofDegree)=@year

* **إجرائية (get\_mat\_year2)**

CREATE proc [dbo].[get\_mat\_year2]

@mat nvarchar(10) ,@year int

as

select ROW\_NUMBER() OVER(ORDER BY id ASC) as' ',Student.id as 'الرقم الجامعي', Student.FName+' '+LName as 'الاسم',Student.Father as 'اسم الأب',degree\_practical as ' العملي',degree\_theoretical as 'النظري',degree\_final as'العلامة' from Degree

join Student on Student.id=Degree.student\_id

where material\_id=@mat and DATEPART(YEAR, Degree.year\_ofDegree)=@year and Degree.degree\_final>=60

* **إجرائية (get\_mat\_year3)**

CREATE proc [dbo].[get\_mat\_year3]

@mat nvarchar(10) ,@year int

as

select ROW\_NUMBER() OVER(ORDER BY id ASC) as' ',Student.id as 'الرقم الجامعي', Student.FName+' '+LName as 'الاسم',Student.Father as 'اسم الأب',degree\_practical as ' العملي',degree\_theoretical as 'النظري',degree\_final as'العلامة' from Degree

join Student on Student.id=Degree.student\_id

where material\_id=@mat and DATEPART(YEAR, Degree.year\_ofDegree)=@year and Degree.degree\_final<60

* **إجرائية جلب جميع المواد ()**

create proc [dbo].[Get\_Material]

as

select id as 'رمز المادة'

,material\_name as 'اسم المادة'

, faculty as 'الكلية'

,[year] as 'سنة المادة'

,season\_mat as 'الفصل الدراسي للمادة'

from Material

* **إجرائية جلب بيانات رسوم طالب (get\_std\_fees)**

CREATE proc [dbo].[get\_std\_fees]

@id\_std nvarchar(15)

as

SELECT [Student\_fees].[student\_id] as 'الرقم الجامعي'

,Student.FName+' '+Student.LName as 'اسم الطالب'

,[full\_year\_fees] as 'الرسم السنوي'

,[fees\_paid] as 'الرسم المدفوع'

,[does\_paid] as 'هل دفع هذا العام'

,[discount\_rate] as 'نسبة الحسم'

,[amount\_of\_discount] as 'مقدار الحسم'

,[year\_fees\_after\_discount] as 'الرسم بعد الحسم'

,[reason\_for\_discount] as 'سبب الحسم'

,[old\_fees\_unpaid] as 'الرسم المتبقي على الطالب'

,[academic\_year] as 'السنة الدراسية'

,[stady\_year] as 'العام الدراسي'

,[voucher\_number] as 'رقم الايصال'

,[date\_paid] as 'تاربخ الدفع'

,[notes] as 'ملاحظات'

from Student\_fees join Student

on [Student\_fees].student\_id=Student.id

where [Student\_fees].[student\_id]=@id\_std

* **إجرائية جلب بيانات طالب (Get\_stdname)**

CREATE PROC [dbo].[Get\_stdname]

@stdnum nvarchar(20)

as

begin

select FName+' '+Father+' '+LName from Student

where id=@stdnum

END

* **إجرائية جلب بيانات طالب (Get\_stdnum)**

CREATE PROC [dbo].[Get\_stdnum]

@name nvarchar (50)

as

begin

declare @f nvarchar(50) =@name;

select id from Student

where (FName+' '+Father+' '+LName ) like (@f) or (FName+' '+LName ) like (@f)

END

* **إجرائية جلب العشرة الأوائل (get\_top10)**

CREATE PROCEDURE [dbo].[get\_top10]

@year int

as

select avg,student\_number,Student.FName+' '+Student.LName as name from(select top 10 round( (CONVERT(float, sum(degree\_final))/(count(degree\_final))),2) as avg

,student\_id as student\_number

from Degree join Material on Material.id=Degree.material\_id

where Material.year=@year

group by student\_id

order by avg desc) a

join Student on Student.id=a.student\_number

where Student.year=@year

* **إجرائية إدخال درجة (insert\_degree)**

CREATE proc [dbo].[insert\_degree]

@std\_num nvarchar(20),

@mat\_id nvarchar(20),

@deg\_pract int,

@deg\_theo int,

@seas int,

@year\_ofDeg date,

@teach\_id int

as

INSERT INTO [dbo].[Degree]

([student\_id]

,[material\_id]

,[degree\_practical]

,[degree\_theoretical]

,[season]

,[year\_ofDegree]

,teacher\_id)

VALUES

(@std\_num

,@mat\_id

,@deg\_pract

,@deg\_theo

,@seas

,@year\_ofDeg

,@teach\_id)

* **إجرائية جلب المواد الراسبة (not\_success\_year)**

CREATE proc [dbo].[not\_success\_year]

@year int,

@n\_of\_mat int,

@fuclty int

as

select student\_id as 'a',count(case when degree\_final<60 then 1 else null end ) as 'fail' ,count(material\_id)as 'all\_mat' into ##fail

from Degree group by Degree.student\_id

select ROW\_NUMBER() OVER(ORDER BY Student.id ASC) as' ',FName+' '+LName as 'الاسم',id as 'الرقم الجامعي' , year as 'السنة',fail as'عدد المواد الراسبة',all\_mat as'المواد المقدمة' from Student

join ##fail

on ##fail.a=Student.id

where year=@year and( fail>( @n\_of\_mat+1) or fail=( @n\_of\_mat+1) )and Student.faculty=@fuclty

--select \* from ##fail

drop table ##fail

* **إجرائية جلب المواد التي لم يتم تقديمها (null\_matirial)**

CREATE proc [dbo].[null\_matirial]

@std nvarchar(15)

as

(

select material\_name from

(select material\_name,id from Material where year<(select Student.year from Student where id =@std)+1) a

full outer join (select student\_id,material\_id,degree\_final from Degree where student\_id =@std)b

on b.material\_id=a.id

where b.degree\_final is null

)

* **إجرائية استبدال درجة الرسوب (replace\_fail\_degree)**

CREATE proc [dbo].[replace\_fail\_degree]

@s\_id nvarchar(10),@m\_id nvarchar(10)

as

select \* from degree

where degree\_final<60 and

student\_id=@s\_id and material\_id=@m\_id

* **إجرائية استعادة نسخة احتياطية (restore\_db)**

CREATE proc [dbo].[restore\_db]

@path nvarchar(75)

as

alter database [AleppoFreeUniversity] set offline with rollback immediate

RESTORE DATABASE [AleppoFreeUniversity] FROM DISK =@path

alter database [AleppoFreeUniversity] set online

* **إجرائية البحث عن الطالب بالرقم (Search\_by3id)**

CREATE proc [dbo].[Search\_by3id]

@id nvarchar(10), @facluty nvarchar(2)

as

select SUBSTRING(id,7,9),id,FName,LName from Student where

id like '%'+@facluty+'10%'+(@id)+'%'

or id=@id

* **إجرائية البحث عن الدرجة (Search\_Degree)**

CREATE proc [dbo].[Search\_Degree]

@std\_id nvarchar(20)

,@mat\_id nvarchar(20)

as

begin

select

[student\_id] as 'الرقم الجامعي'

,[material\_name] as 'اسم المادة'

,[material\_id]as ' رمز المادة'

,[degree\_practical] as ' درجة العملي'

,[degree\_theoretical] as ' درجة النظري'

,[degree\_final] as ' الدرجة النهائية'

,[season] as ' الفصل'

,[year\_ofDegree] as ' سنة المادة'

from Degree join Material

ON Material.id=Degree.material\_id

where student\_id = @std\_id or material\_id like (@mat\_id+'%')

end

* **إجرائية البحث عن المادة عبر الرقم (search\_material\_by\_id)**

CREATE proc [dbo].[search\_material\_by\_id]

@id nvarchar(12)

as

begin

select id as 'رمز المادة'

,material\_name as 'اسم المادة'

, faculty as 'الكلية'

,[year] as 'سنة المادة'

,season\_mat as 'الفصل الدراسي للمادة'

from Material

where id = @id

end

* **إجرائية البحث عن المادة عبر الاسم (search\_material\_by\_name)**

CREATE proc [dbo].[search\_material\_by\_name]

@mat\_name nvarchar(30)

as

begin

select id as 'رمز المادة'

,material\_name as 'اسم المادة'

, faculty as 'الكلية'

,[year] as 'سنة المادة'

,season\_mat as 'الفصل الدراسي للمادة'

from Material

where material\_name like (@mat\_name+'%') or material\_name like ('%'+@mat\_name+'%') or material\_name like ('%'+@mat\_name)

end

* **إجرائية البحث عن الطالب عبر الاسم (Search\_Student\_name)**

CREATE proc [dbo].[Search\_Student\_name]

@first\_name nvarchar(50)

as

begin

declare @f nvarchar(50) =@first\_name;

select ROW\_NUMBER() OVER(ORDER BY Student.id ASC) as' ',

Student.id as 'الرقم الجامعي'

,[FName] as 'الاسم '

,[LName] as 'الكنية'

,[Father] as 'الأب'

,[Mother] as 'الأم'

,[gender] as 'الجنس'

,[birth\_city] as 'مكان الميلاد'

,[birth\_date] as 'تاريخ الميلاد'

,[box\_Number] as 'القيد'

,[faculty] as 'الكلية'

,[year] as 'السنة'

,[telephon]as 'رقم الهاتف'

,[Certificate\_Sec]as 'نوع الشهادة'

,[Certifcate\_Source]as 'مصدر الشهادة'

,[Certificate\_date]as 'تاريخ الشهادة'

,[yearOfAdmission]as 'سنة القبول'

,[typeOfAdmission]as 'نوع القبول'

,[Department]as 'القسم'

,pictur

,[section]

,[second\_language]

,[Matrial\_status]

,[personal\_ID]

,[husband\_name]

,[husband\_work]

,[Children\_Number]

,[Children\_age]

,[migration]

,[migration\_date]

,[father\_work]

,[Health\_status]

,[personal\_notes]

,[email]

,[address\_original]

,[address\_current]

,[Certifcate\_state]

,[Certifcate\_id]

,[Certifcate\_total]

-- ,[Certifcate\_total\_document]

,[old\_universirt]

,[old\_faculty]

,[old\_Department]

,[old\_yearOfAdmission]

,[date\_ofStop]

,[old\_yearStudy]

,[Old\_documents\_note]

-- ,[Old\_documents]

,[old\_Institute]

,[old\_Institute\_department]

,[old\_Institute\_date]

,[old\_Institute\_seson]

,[old\_Institute\_total]

,[old\_Institute\_rating]

from [dbo].[Student]

left join Student\_detile on Student.id=Student\_detile.id

where (FName+' '+LName ) like (@f+'%') or (FName+' '+Father+' '+LName ) like (@f+'%') or LName like @f or Student.id like ('%'+@f+'%')

end

* **إجرائية البحث عن معلم عبر الرقم (search\_teacher\_by\_id)**

create proc [dbo].[search\_teacher\_by\_id]

@id int

as

begin

select id as 'رقم المدرس'

,teacher\_name as 'اسم المدرس'

, scientific\_rank as 'الرتبة العلمية'

,num\_of\_material as 'عدد المواد'

from teacher

where id = @id

end

* **إجرائية البحث عن معلم عبر الاسم (search\_teacher\_by\_name)**

CREATE proc [dbo].[search\_teacher\_by\_name]

@teach\_name nvarchar(20)

as

begin

select id as 'رقم المدرس'

,teacher\_name as 'اسم المدرس'

,scientific\_rank as 'الرتبة العلمية'

,num\_of\_material as 'عدد المواد'

from teacher

where teacher\_name like (@teach\_name+'%') or teacher\_name like ('%'+@teach\_name+'%') or teacher\_name like ('%'+@teach\_name)

end

* **إجرائية تسجيل الدخول (Sp\_Login)**

create proc [dbo].[Sp\_Login]

@UserName varchar(50)،@Password varchar(50)

as

select \* from [User]

where UserName=@UserName and Password=@Password

* **إجرائية تفاصيل طالب (Student\_detiles)**

CREATE PROCEDURE [dbo].[Student\_detiles]

@id nvarchar(20)

AS

BEGIN

select FName+' '+LName,Father,Mother,Left(birth\_date,10),year(yearOfAdmission),[year]

,University\_Faculty.faculty\_name,birth\_city,typeOfAdmission,Department

from Student

join University\_Faculty on University\_Faculty.id=Student.faculty

where Student.id=@id

END

* **إجرائية عدد الطلاب الناجحين (success\_num)**

CREATE proc [dbo].[success\_num]

@mat nvarchar(10) ,@year int

as

select count(Student.id) as 'عدد الطلاب الناجحين' from Degree

join Student on Student.id=Degree.student\_id

where material\_id=@mat and DATEPART(YEAR, Degree.year\_ofDegree)=@year and Degree.degree\_final>=60

* **إجرائية انتقال الطلاب للسنة التالية (success\_year)**

CREATE proc [dbo].[success\_year]

@year int,

@n\_of\_mat int,

@fuclty int

as

select student\_id as 'a',count(case when degree\_final<60 then 1 else null end ) as 'fail' ,count(material\_id)as 'all\_mat' into ##fail

from Degree group by Degree.student\_id

--where degree\_final<60

select ROW\_NUMBER() OVER(ORDER BY Student.id ASC) as' ',FName+' '+LName as 'الاسم',id as 'الرقم الجامعي' , year as 'السنة',fail as'المواد الراسبة',all\_mat as'المواد المقدمة' from Student

join ##fail

on ##fail.a=Student.id

where year=@year and fail<( @n\_of\_mat+1) and Student.faculty=@fuclty

--select \* from ##fail

drop table ##fail

* **إجرائية (success\_year\_ok)**

CREATE proc [dbo].[success\_year\_ok]

@year int,

@n\_of\_mat int,

@fuclty int

as

select student\_id as 'a',count(case when degree\_final<60 then 1 else null end ) as 'fail' ,count(material\_id)as 'all\_mat' into ##fail

from Degree group by Degree.student\_id

update student

set Student.year=Student.year+1

from Student

join ##fail

on ##fail.a=Student.id

where year=@year and fail<( @n\_of\_mat+1) and Student.faculty=@fuclty



**جامعة حلب في المناطق المحررة**