الجمهورية العربية السورية
وزارة التعليم العالي
المعهد العالي لإدارة الأعمال

السنة: الرابعة
الاختصاص: إدارة العمليات والمعلومات

تي-سويفت: نظام إدارة الجدولة الزمنية

المعهد العالي لإدارة الأعمال

TSwift: Timetable Management System

Higher Institute of Business Administration

إشراف الأستاذ الدكتور:
كادان جمعة

إعداد الطلاب:
أمين أيوب
عبد الكريم رجب

2021 - 2022

جميع الآراء الواردة في هذا البحث تعبر عن وجهة نظر مُعدّيه ولا يتحمل المعهد أي مسؤولية عن مدى دقة أو مصداقية الآراء المطروحة فيه

ملخص البحث

يهدف هذا البحث إلى تحليل وتصميم وتنفيذ نظام لحل مشكلة الجدولة الزمنية للمقررات الدراسية اعتماداً على البرمجة الطبيعية المختلطة وخوارزمية مستقلة لتعيين القاعات الدراسية للمحاضرات. تم تطبيق النظام في المعهد العالي لإدارة الأعمال HIBA وهو معهد مختص في علوم الإدارة يمنح درجة الإجازة ويتضمن برامج مختلفة للدراسات العليا.

ضمن الإطار النظري للبحث، يتم توضيح جميع المفاهيم التي لها علاقة بموضوع البحث، حيث يبدأ الفصل بالتعريف بمسألة الجدولة الزمنية في المؤسسات التعليمية بأنواعها المختلفة وبيان نقاط التشابه والاختلاف بين هذه الأنواع. ثم يتناول مسألة الجدولة الزمنية للمقررات الدراسية بشيء من التفصيل. ينتقل بعد ذلك للحديث عن بحوث العمليات وتطبيقاتها المختلفة مثل البرمجة الخطية LP والبرمجة الطبيعية IP اللتان تم تناولهما من خلال عرض تطبيقاتهم المختلفة ومبررات استخدام كل منهما. ثم يختتم الفصل بشرح أدوات تحليل وتصميم النظم.

في الإطار العملي للبحث، يتم البدء بمرحلة تحليل النظام، حيث يتم تقديم لمحة عن المعهد العالي لإدارة الأعمال وطبيعة أنشطته ومن ثم شرح طبيعة عمل نظام الجدولة الحالي فيه كنقطة انطلاق في عملية تحديد متطلبات النظام الجديد. بعد ذلك تتم صياغة مسألة الجدولة الزمنية، توضيحاً لكل عناصر ومكونات المسألة وتمهيداً لعملية التطبيق. وبناء على ما سبق يتم استنتاج وتحديد متطلبات النظام الرئيسية. ثم ننتقل إلى مرحلة التصميم والتي تتضمن تصميم قاعدة المعطيات والخوارزميات ونماذج الحل الأساسية، حيث يتم بناء نموذج البرمجة الطبيعية المعتمد وشرحه، بالإضافة الى عرض الخوارزمية المسؤولة عن تعيين القاعات الدراسية للمحاضرات. ننتقل بعد ذلك إلى المرحلة الأخيرة في الإطار العملي وهي مرحلة التنجيز حيث يتم عرض مجموعة الأدوات البرمجية التي تم استخدامها في تطوير النظام، بالإضافة إلى عرض واجهات الاستخدام الخاصة بالنظام وشرحها، انتهاءً بإجراء مجموعة من الاختبارات التي تبين مدى كفاءة أداء النظام. هذا وقد آلت الدراسة إلى مجموعة من التوصيات والافاق التي يمكن العمل على تطويرها في المستقبل.

الكلمات المفتاحية: جدولة المقررات الدراسية، متطلبات ضغط الجدول الزمني، البرمجة الطبيعية المختلطة، الأمثلة، بايثون.

Abstract

This research aims to analyze, design and implement a system to solve course timetabling problem based on Mixed Integer Programming (MIP) and an individual algorithm for Room-Lectures Assignment. The system was implemented at the Higher Institute of Business Administration (HIBA), which is an institute specialized in management sciences and awards a bachelor degree, and various postgraduate programs.

The theoretical framework, clarifies the related research topic notions. The chapter begins by defining the timetabling problem in different educational institutions and points out similarities and differences between these types. Then it discusses the course-timetabling problem in details. Henceforward, it elaborates on Operations Research and its various applications such as Linear Programming (LP) and Integer Programming (IP), which were both addressed by presenting their different applications and the justifications for their use. The chapter concludes with a commentary on system analysis & designing tools.

The practical framework begins with the system analysis phase, where it provides an overview of the Higher Institute of Business Administration and its activities, then explains how does the current timetabling system work which serves as a starting point in the process of determining system requirements. After that, the timetabling problem is formulated to clarify all the elements and components of the problem and in preparation for the implementation process. Based on the above, the main system requirements are identified. Then it moves to the designing phase, which includes the design of database, algorithms and main solution models, where the Integer Programming model is built and explained, in addition to the presentation of the rooms assigning algorithm used. Then it moves to the last phase in the practical framework, which is the implementation stage, where the set of tools that were used in the development of the system are presented. In addition to the presentation and explanation of the system's graphical user interfaces, ending with a set of tests that show the efficiency of the system's performance. The study ended with a set of recommendations and horizons that could be developed in the future.

Keywords: Course Timetabling, Compactness Requirements, Mixed Integer Programming, MIP, Optimization, Python.

الشكر

جزيل الشكر والتقدير لمعلمنا الذي قدم لنا الكثير الكثير وكان قريبا منا مساعدا لنا في كل خطوة،

ولتفضله الكريم بالإشراف على هذا البحث

الاستاذ الدكتور كادان جمعة.

نتقدم بوافر الشكر والامتنان لمن تفانى في تعليمنا ما هو أسمى وأعلى من العلم والمعرفة

الأستاذ الدكتور وائل خنسا .

لست من النوع الذي ينتظر الشكر من الآخرين، وتقدم كل هذه التضحيات الرائعة من أجل الحفاظ

على المكانة العلمية للمعهد، ولكن الشكر والاحترام والتقدير من أهم الواجبات التي يجب علينا

تقديمها لك

الأستاذ الدكتور طلال عبود . . .شكراً جزيلاً لك.

لكِ منّا أسمى و أعطر عبارات الشكر و التقدير دكتورتي الحبيبة، فلطالما كانت روحك جميلة و

طيبة و أخلاقك نبيلة و حسنة و إبتسامتك تبعث في النفوس الأمل، فقيمتك عندنا عظيمة

المهندسة نظرة رحمة.

إهداء

أُهدي بحثي هذا إلى والدي الغالي

الذي كان مثالاً لي في علو الهمة والعطاء المستمر

. . . .

وإلى والدتي الغالية التي كانت دعوتها نسمات فرح على قلبي

. . .

والى أخوتي وأصدقائي وزملائي الذين لا تحلو الحياة بدونهم

. . .

وأخيراً وليس آخراً إلى جامعتي ودكاترتي الذين لم يبخلوا علينا بعلمهم وخبرتهم اهديكم ثمرة عملي

أمين أيوب

إهداء

أشكر الله سبحانه وتعالى على توفيقه الذي يفوق الادراك
على نعمه التي لا حصر لها وفضله الذي لا حدود له
على نعمة أمي . . . وأبي . . . وأخوتي
على أن جمعني بخير المعلمين وخير الأصدقاء

أهدي هذا العمل
لمن لا يسع قلباً حنان قلبها . . . ولا يكفي جميل لرد فضلها
لمن تحملتني كثيراً . . . كثيراً . . . حبيبة قلبي . . . أمي

للصبور المكافح المعطاء
لأنقى قلب رأيته في حياتي
قُدوتي . . . أبي

إلى من هم سندي وعزوتي في هذه الحياة
إخوتي

إلى كل من له فضل علي . . . وهم كثر

عبد الكريم رجب

فهرس المحتويات

ملخص البحث I
الشكر III
إهداء IV
إهداء V
فهرس المحتويات VI
فهرس الجداول IX
فهرس الأشكال X
1. الفصل الأول (الإطار العام للبحث) 1
1.1 تمهيد 2
2.1 مشكلة البحث 3
3.1 أهمية البحث 4
4.1 أهداف البحث 4
5.1 منهجية البحث 5
6.1 حدود البحث ومحدداته 5
1.6.1 الحدود الزمانية 5
2.6.1 الحدود المكانية 5
3.6.1 المحددات 5
7.1 فكرة الحل المقترح 6
2. الفصل الثاني (الأنظمة المشابهة) 7
1.2 تمهيد 8
2.2 التقنيات المستخدمة في حل مسألة الجدولة 8
3.2 أمثلة عن الأنظمة المشابهة 10
3. الفصل الثالث (الإطار النظري العام) 12
1.3 تمهيد 13
2.3 مسألة الجدولة الزمنية 13
1.2.3 خصائص مسألة الجدولة الزمنية 15
2.2.3 مسألة الجدولة الزمنية للمقررات الدراسية 16
3.3 بحوث العمليات 18
1.3.3 البرمجة الخطية Linear Programming 18
2.3.3 البرمجة الطبيعية المختلطة Mixed-Integer Programming 21
4.3 أدوات تحليل وتصميم النظم 24
1.4.3 دورة حياة تطوير النظم Systems Development Life Cycle 24
1.1.4.3 مراحل دورة حياة تطوير النظام 24

2.4.3 منهجيات تحليل وتصميم النظم 28
1.2.4.3 التصميم البنيوي Structures Design 28
2.2.4.3 التطوير السريع للتطبيقات Rapid Application Development 30
3.2.4.3 التطوير الرشيق Agile Development 34
3.4.3 اختيار منهجية التطوير المناسبة 35
4. الفصل الرابع (الإطار العملي) 37
1.4 تمهيد 38
2.4 تحليل النظام 38
1.2.4 مقدمة عن المعهد العالي لإدارة الأعمال 39
1.1.2.4 تحليل نظام الجدولة الحالي في المعهد 40
2.2.4 صياغة المسألة وتحديد المتطلبات 42
3.2.4 تحليل حالات الاستخدام 45
4.2.4 تصميم قاعدة المعطيات 49
1.4.2.4 الكائنات ومخطط العلاقات بين الكيانات 49
2.4.2.4 تصميم ووضع الجداول 52
3.4 تصميم النظام 54
1.3.4 الحل المقترح 54
1.1.3.4 النموذج الرياضي 55
1.1.1.3.4 المجموعات والمعامِلات Sets & Parameters 55
2.1.1.3.4 تابع الهدف والقيود الصلبة والمرنة 57
3.1.1.3.4 النموذج الرياضي 58
4.1.1.3.4 شرح النموذج 60
2.1.3.4 خوارزمية تعيين القاعات 61
4.4 تنجيز النظام 63
1.4.4 الأدوات المستخدمة 63
1.1.4.4 لغة بايثون Python 63
2.1.4.4 واجهة المستخدم الرسومية المستخدمة (GUI) 64
3.1.4.4 أدوات النمذجة الرياضية 65
1.3.1.4.4 مكتبة Numeric Python (NumPy) 65
2.3.1.4.4 أداة النمذجة Python-MIP 65
3.3.1.4.4 أداة الحل CBC Solver 66
4.1.4.4 نظام إدارة قاعدة المعطيات 66
2.4.4 تطبيق النظام 68
1.2.4.4 عرض واجهات الاستخدام 68
2.2.4.4 الاختبارات 92
1.2.2.4.4 تصنيف الاختبارات 92
2.2.2.4.4 نتائج الاختبارات 95

5 الفصل الخامس (النتائج والتوصيات)..... 103
1.5 تمهيد 104
2.5 خلاصة البحث..... 104
3.5 الآفاق المستقبلية..... 105
6. المراجع.....106
1.6 المراجع الأجنبية 107
2.6 المراجع العربية 110

فهرس الجداول

الجدول (1) معطيات المسألة..... 20
الجدول (2) مقارنة منهجيات التطوير 36
الجدول (3) الكيانات..... 49
الجدول (4) العلاقات بين الكيانات 50
الجدول (5) تفاصيل الجداول 52
الجدول (6) المجموعات الأساسية 55
الجدول (7) المجموعات الجزئية..... 56
الجدول (8) المعاملات..... 56
الجدول (9) متغيرات القرار والمتغيرات المساعدة 58
الجدول (10) مدخلات خوارزمية تعيين القاعات 61
الجدول (11) تصنيف الاختبارات 92
الجدول (12) مدخلات مجموعة اختبارات A 92
الجدول (13) مدخلات مجموعة اختبارات B وC..... 94
الجدول (14) نتائج الاختبارات..... 95
الجدول (15) تفاصيل نتائج مجموعة اختبارات A بالنسبة للمدرسين 99
الجدول (16) Paired Samples Statistics للمدرسين 100
الجدول (17) Paired Sample Test للمدرسين..... 100
الجدول (18) تفاصيل نتائج مجموعة اختبارات A بالنسبة للفئات 101
الجدول (19) Paired Sample Statistics للفئات 101
الجدول (20) Paired Sample Test للفئات..... 102

فهرس الأشكال

الشكل (1) شعار نظام UniTime 10
الشكل (2) شعار نظام Prime Timetable 10
الشكل (3) شعار نظام Bullet TimeTabler 10
الشكل (4) التمثيل البياني للمسألة 20
الشكل (5) التمثيل البياني الجديد للمسألة 23
الشكل (6) التطوير الشلالي 29
الشكل (7) التطوير على التوازي 30
الشكل (8) التطوير التكراري 31
الشكل (9) النمذجة الأولية 32
الشكل (10) النمذجة الأولية مع رمي النموذج 33
الشكل (11) البرمجة الحدية 34
الشكل (12) جدول المحاضرات الأسبوعي للسنة الرابعة 40
الشكل (13) المراحل الدراسية في المعهد 41
الشكل (14) صيغة موحدة للمحاضرة 42
الشكل (15) مسألة الجدولة كمسألة تعيين 43
الشكل (16) المخطط العام لحالات الاستخدام 45
الشكل (17) مخطط العلاقات بين الكيانات ERD 51
الشكل (18) النموذج المحسن لعلاقات الكيانات 53
الشكل (19) خطوات الحل المقترح 54
الشكل (20) مخطط تدفق خوارزمية تعيين القاعات 62
الشكل (21) شعار لغة بايثون 63
الشكل (22) تصنيف لغات البرمجة حسب شهرتها لعام 2020 حسب Github 64
الشكل (23) شعار حزمة Tcl\Tk 64
الشكل (24) شعار مكتبة NumPy 65
الشكل (25) شعار نظام MySQL 66
الشكل (26) تصنيف أنظمة إدارة قواعد المعطيات حسب Stack Overflow 66
الشكل (27) واجهة الاتصال مع قاعدة المعطيات 68
الشكل (28) واجهة الاستخدام الرئيسية 69
الشكل (29) واجهة استيراد البيانات 70
الشكل (30) واجهة إدخال المقررات 71
الشكل (31) واجهة إدخال المدرسين 72
الشكل (32) واجهة إدخال السنوات الدراسية 72
الشكل (33) واجهة إدخال القاعات الدراسية 73
الشكل (34) واجهة إدخال فئات الطلاب 74
الشكل (35) واجهة إدخال المحاضرات 1 75
الشكل (36) واجهة إدخال المحاضرات 2 76
الشكل (37) واجهة تحديد الأوقات 77
الشكل (38) واجهة اختيار المحاضرات 1 78
الشكل (39) واجهة اختيار المحاضرات 2 79
الشكل (40) واجهة خصائص المدرسين 1 80
الشكل (41) واجهة خصائص المدرسين 2 81
الشكل (42) واجهة اختيار القيود المرنة 82
الشكل (43) واجهة إضافة قيود اختيارية 1 83
الشكل (44) واجهة محرك الحل Solver 84

الشكل (45) واجهة فتح حل سابق..... 85
الشكل (46) واجهة قاعات المحاضرات 1..... 86
الشكل (47) واجهة قاعات المحاضرات 2..... 87
الشكل (48) واجهة الجداول الأسبوعية 1..... 88
الشكل (49) واجهة الجداول الأسبوعية 2..... 89
الشكل (50) واجهة الجداول الأسبوعية 3..... 90
الشكل (51) واجهة الجداول الأسبوعية 4..... 91
الشكل (52) واجهة الجداول الأسبوعية 5..... 91
الشكل (53) واجهة إضافة قيود اختيارية 2..... 93
الشكل (54) واجهة الجداول الأسبوعية 6..... 96
الشكل (55) واجهة الجداول الأسبوعية 7..... 97
الشكل (56) واجهة الجداول الأسبوعية 8..... 98
الشكل (57) واجهة الجداول الأسبوعية 9..... 98

قائمة المصطلحات

المصطلح أو الاختصار	المعنى باللغة العربية
Timetabling	الجدولة الزمنية
Hard Constraints	القيود الصلبة
Soft Constraints	القيود المرنة
Lecture	محاضرة
Lesson	حصة دراسية
Course	مقرر دراسي
Optimization Problem	مسألة أمثلة
Multigraph	بيان مضاعف
Assignment	تعيين
multi-dimensional assignment problem	مسألة تعيين متعددة الأبعاد
Supply Chain Management	إدارة سلسلة التوريد
Revenue Management	إدارة الإيرادات
Work Crew Scheduling	إدارة أطقم العمل
Production Scheduling	جدولة الانتاج
Shipping Networks	شبكات الشحن
Facility Locations	مواقع المنشآت
Model	نموذج
Linear Programming (LP)	البرمجة الخطية
Integer Programming (IP)	البرمجة الطبيعية
Mixed Integer Programming (MIP)	البرمجة الطبيعية المختلطة
Mathematical Programming	البرمجة الرياضية
Optimization	الأمثلة
Variable	متغير
Constraint	قيد
Objective Function	تابع الهدف
Non-negativity Constraints	قيود اللاسلبية
Feasible Region	فضاء الحلول المقبولة
Optimal Solution	حل أمثل
Upper Bound	حد أعلى
Lower Bound	حد أدنى
Branch & bound	التفرع والحد

1

الفصل الأول

مدخل إلى موضوع البحث وأهميته

- تمهيد
- مشكلة البحث
- أهمية البحث
- أهداف البحث
- منهجية البحث
- حدود البحث ومحدداته
- فكرة الحل المقترح

SWIFT

Timetable Management System

1.1 تمهيد

التعليم ضروري لاستمرار تحسين البشرية. ولحسن الحظ، يعد التعليم الأساسي في معظم الأماكن حول العالم جزءً طبيعياً من النمو. ينقسم التعليم الرسمي عادةً إلى أربع مراحل متمايزة: مرحلة ما قبل المدرسة، والمدرسة الابتدائية، والمدرسة الثانوية، والجامعة او المعاهد التقانية. تمارس المؤسسات التعليمية بمختلف أشكلها مجموعة واسعة من الأنشطة من أهمها عملية تخصيص موارد المؤسسة وتنظيم مواعيد الحصص الدراسية والأنشطة المختلفة الأخرى. يحظى الجدول الزمني في مختلف المؤسسات التعليمية بأهمية كبيرة، للطلاب والموظفين من جهة والمؤسسة نفسها من جهة أخرى. حيث يريد الطلاب والموظفون جداول زمنية منطقية تحقق رغباتهم ولا تسبب تعارض مع بقية أنشطتهم، أما المؤسسة فتسعى إلى استخدام مواردها بأفضل وجه ممكن. لذلك، من الضروري إنشاء جداول زمنية تحقق التوازن بين احتياجات الطلاب والموظفين والمؤسسة التعليمية.

في المراحل المبكرة من التعليم الرسمي، يتعلم الطلاب المواد الأساسية وعادة ما يكون لديهم عدد قليل من المعلمين، بينما تتميز المراحل اللاحقة بمزيد من المواد والاختيارية والتفرع والتخصصات الدراسية، وبالتالي المزيد من المتطلبات، الأمر الذي أدى الى اختلاف طبيعة الجداول وخصائصها بين المؤسسات التعليمية مما يجعل مشكلة إنشاء الجداول الزمنية أكثر صعوبة. إن إعداد مثل هذه الجداول الزمنية يدوياً يستغرق وقتاً طويلاً ومرهقاً ومن المرجح أن يؤدي إلى جداول زمنية متعارضة وغير مرضية. لذلك، تم تكريس الكثير من الاهتمام في مجال البحث الخاص بالجداول الزمنية التعليمية.

يتناول البحث دراسة حالة عملية لتطبيق نظام لجدولة المحاضرات الأسبوعية في المعهد العالي لإدارة الأعمال، وهو مؤسسة تعليمية تابعة لوزارة التعليم العالي في الجمهورية العربية السورية. يقدم المعهد مجموعة من البرامج التعليمية باختصاصات مختلفة حيث يمنح درجة الإجازة الجامعية في علوم الإدارة ودرجة الماجستير والدكتوراه. يؤدي تعدد البرامج التعليمية والتخصصات التي يقدمها المعهد واشتراكها جميعاً بنفس الموارد إلى صعوبة إنشاء الجداول الأسبوعية للمحاضرات.

وبناء على هذه المشكلة يقدم البحث دراسة وتحليل نظام الجدولة الحالي في المعهد بحيث يتم بعد ذلك تصميم نظام جديد يقدم حل لهذه المشكلة ويحقق جميع المتطلبات والقيود بشكل مؤتمت. حيث سيتم تنفيذ النظام باستخدام مجموعة من الأدوات الرياضية والبرمجية المناسبة وإضافة مجموعة من الميزات لتسهيل عملية استخدام النظام.

2.1 مشكلة البحث

تعاني الجامعات كغيرها من المؤسسات التعليمية من صعوبة بالغة في إنشاء الجداول الدراسية للمحاضرات، خاصةً أن هذه العملية الصعبة تتكرر بشكل دوري بداية كل فصل دراسي وهي من المشاكل التي لا يوجد لها حل سهل أو بسيط حتى مع استخدام الحواسيب الخارقة (Super-Computers)، حيث تتعقد المشكلة وتصبح أحياناً مستحيلة الحل مع زيادة المعطيات والقيود المطلوبة.

تعتمد البرامج والطرق الحديثة التي تهدف لحل مشكلة الجدولة الزمنية على خوارزميات جديدة وجيدة ولكنها تواجه عدة مشاكل منها:

أن هذه البرامج تفتقر إلى العمومية التي تمكنها من العمل في الكثير من المؤسسات التعليمية، فهذه البرامج غير مناسبة للكثير من المعاهد والجامعات وتحتاج إلى تعديل قبل تطبيقها في هذه المؤسسات. إن تعقيد هذه الخوارزميات قد يصبح من رتبة عالية تصل إلى درجة NP (nondeterministic polynomial) في حال تعقدت شروط وقيود الحل.

يمكن التعبير عن مشكلة البحث باستخدام عبارات لفظية أو باستخدام أسئلة التي يحاول البحث الإجابة عنها وتتمثل مشكلة البحث في العبارة التالية:

"كيف يمكن جدولة المحاضرات الأسبوعية في المعهد العالي لإدارة الأعمال بشكل آلي ؟ "

كما يمكن التعبير عن مشكلة البحث عن طريق طرح عدة أسئلة، يعمل البحث على إيجاد جواب لها ويمكن عرض الأسئلة التالية:

1- كيف تقوم إدارة المعهد بجدولة المحاضرات الأسبوعية؟

2- ما هي حاجة المعهد لنظام جدولة المحاضرات الأسبوعية؟

3- ما هي متطلبات المعهد المراد تحقيقها من تطبيق نظام جدولة المحاضرات الأسبوعية؟

4- ما هي الفوائد التي ستعود على المعهد من جراء تطبيق نظام جدولة المحاضرات الأسبوعية؟

5- كيف يمكن تطبيق نظام جدولة المحاضرات الأسبوعية في المعهد والاستفادة منه بالشكل الأمثل؟

يتم الإجابة عن هذه التساؤلات ضمن هذه الدراسة، بحيث يتم توضيح جميع الجوانب النظرية المتعلقة بموضوع البحث بالإضافة لدراسة الجوانب العملية وعرضها بطريقة منظمة توضح وتقدم حل لمشكلة الدراسة.

3.1 أهمية البحث

تسعى إدارات المؤسسات التعليمية للوصول إلى جدول زمني للمحاضرات يحقق أكبر قدر من الرضى لجميع الأطراف المعنية منه (طلاب، محاضرين، إداريين....) ويناسب قدرات المؤسسة ومواردها المتاحة. عادةً ما تتم هذه العملية بشكل يدوي حيث تحتاج إلى وقت وجهد كبيرين ومع ذلك قد لا تحقق الجداول الناتجة جميع قيود المؤسسة ومتطلباتها وتبقى قيد التعديل خلال الأسابيع الأولى من بداية الفصل الدراسي.

تكمن أهمية البحث النظرية في عرض وتوضيح العديد من المفاهيم والمصطلحات التي تخدم فكرة البحث.

أما أهمية البحث التطبيقية فتكمن في بناء نظام لإنشاء الجداول الزمنية للمحاضرات وتسهيل عملية التعديل عليها والتأكد من صحتها وخلوها من التضاربات وإمكانية تطبيقها. وبهدف إنشاء الجداول الزمنية للمحاضرات تم بناء نموذج لحل هذه المسألة وتوليد جداول دراسية جيدة تراعي موارد المؤسسة المتاحة وتلبي أكبر قدر من المتطلبات الموضوعة من قبل المؤسسة.

حيث تم بناء النظام باستخدام لغة Python وبناء قاعدة معطياته باستخدام نظام إدارة قواعد المعطيات MySQL وتطوير واجهات استخدام توفر العديد من الميزات مثل إضافة\حذف\تعديل المعطيات بطريقة سهلة وواضحة، وعرض الجداول الزمنية للمحاضرات التي سبق وقام النظام بإنشائها وإتاحة إمكانية التعديل عليها وتصنيفها حسب السنوات الدراسية، الفئات، والمدرسين وتخريجها بصيغة Excel.

4.1 أهداف البحث

يهدف النظام المقترح إلى تسهيل واتمتة عملية انشاء جدول المحاضرات الأسبوعي، مما يحقق جملة من المنافع المتنوعة للمستفيدين الرئيسين منه، نذكر منها:

- **على صعيد الإداريين**

1. تسهيل عملية بناء جدول المحاضرات الأسبوعي.
2. تقليل فترة إنشاء جدول المحاضرات الأسبوعي من عدة أيام إلى عدة ساعات.
3. تحقيق أكفأ توزيع لموارد المعهد من قاعات ومدرسين.
4. تحقيق مجموعة من القيود الإدارية المراد تحقيقها من قبل إدارة المعهد.

- **على صعيد المدرسين**

1. إمكانية تحديد أوقات الفراغ المناسبة لكل مدرس لإلقاء محاضراته فيها.
2. إمكانية ضغط الجدول الزمني للمدرس بحيث يكون عدد أيام قدومه للمعهد أقل ما يمكن.
3. إمكانية نقل محاضرة من فترة لأخرى بسهولة.
4. إمكانية نقل المحاضرة من قاعة لأخرى من نفس النوع او من غير نوع.
5. يستلم المدرس بعد إنشاء جدول المحاضرات الأسبوعي جدول خاص فيه يظهر فيه مواعيد وقاعات القاء محاضراته.

❖ **على صعيد الطلاب**

1. يستلم الطلاب جدول المحاضرات الأسبوعي ببداية كل فصل دون تأخير.
2. إمكانية ضغط الجدول لجعل أيام حضور الطلاب أصغر ما يمكن.
3. سهولة تغيير زمن وقاعة المحاضرة بسهولة نزولاً عند رغبة الطلاب.

5.1 منهجية البحث

تم الاعتماد على منهجية التطوير الشلالي (Waterfall Development) في تطوير نظام TSwift، حيث تم اتباع هذه المنهجية لأنه وفي الحالة المدروسة من الممكن تحديد المتطلبات بشكل دقيق في المراحل الأولى من المشروع، وذلك لوجود متطلبات واضحة وتوفر وقت كافي للإنجاز بالإضافة إلى الرغبة بالحصول على جودة مثالية من النظام وموثوقية عالية منه وهي ما توفره هذه المنهجية.

6.1 حدود البحث ومحدداته

تشمل حدود البحث الظروف المحيطة بالبحث والتي تؤثر وتتأثر به، حيث تقسم إلى قسمين وهما الحدود الزمانية والحدود المكانية.

1.6.1 الحدود الزمانية

تم البدء بالعمل على بناء نظام TSwift في الفترة الممتدة من 2021/9/1 إلى 2022/9/1، حيث تم خلال هذه الفترة جمع المعلومات، التخطيط، التحليل، التصميم، التنجيز ومن ثم إجراء الاختبارات اللازمة لضمان عمل النظام بالشكل الصحيح.

2.6.1 الحدود المكانية

تشمل الحدود المكانية مكان التطبيق، حيث تمت عملية صناعة وتطبيق النظام في دمشق، لصالح المعهد العالي لإدارة الأعمال HIBA.

3.6.1 المحددات

تشمل المحددات الصعوبات التي واجهها الباحث أثناء إعداده للبحث. ومن هذه الصعوبات صعوبة الحصول على المعلومات وترجمة المصطلحات وذلك لعدم العثور على أية دراسة مشابهة باللغة العربية. بالإضافة إلى عدم تفرغ الباحثان بشكل كامل بسبب انشغالهما في تحصيلهم العلمي. كما واجه الباحثان صعوبة في اختيار الأدوات المناسبة للتنفيذ.

7.1 فكرة الحل المقترح

يقوم النظام بتوليد الجداول الزمنية للمحاضرات على مرحلتين. المرحلة الأولى تتم باستخدام البرمجة الطبيعية المختلطة (MIP) حيث تم بناء نموذج يحقق مجموعة من المتطلبات مثل عدم تعارض المحاضرات وضغط الجدول الزمني، يتم في هذه المرحلة توزيع المحاضرات على الجدول الزمني دون إسناد القاعات، والمرحلة الثانية يقوم فقط بإسناد وتوزيع للقاعات بعد انتهاء المرحلة الأول باستخدام خوارزمية أخرى مخصصة لهذا الغرض.

2

الفصل الثاني

الأنظمة المشابهة

- تمهيد
- التقنيات المستخدمة
- أمثلة عن الأنظمة المشابهة

1.2 تمهيد

يقدم هذا الفصل لمحة عامة عن أبرز التقنيات المستخدمة لحل مشكلة الجدولة الزمنية، مشيراً إلى أهم الدراسات المتخصصة في هذه التقنيات. ثم سنعرض بعض الأنظمة المتوفرة في الأسواق، موضحين أبرز خصائصها وميزاتها وعيوبها.

2.2 التقنيات المستخدمة في حل مسألة الجدولة

يمكن صياغة مسائل الجدولة الزمنية بالكثير من الطرق وذلك حسب تفضيلات المؤسسة التعليمية، هذا ما أدى إلى تطوير نماذج مختلفة في الأدبيات السابقة. إن بعض القيود شائعة بين معظم المؤسسات التعليمية، في حين إن هنالك قيود أخرى تخصصية جداً حيث تعنى بها مؤسسة محددة دون غيرها. ومن أجل هذه القيود التخصصية يقوم الباحثون بمحاولات كثيرة لإيجاد منهجيات جديدة أو تطوير المنهجيات الحالية بهدف حل مشاكلهم الخاصة. تم استخدام طرق مختلفة، بما في ذلك مفاهيم من نظرية البيان Graph Theory أو بشكل أكثر دقة تلوين البيان Graph Coloring، والبرمجة الخطية الطبيعية Integer Linear Programming، والخوارزميات الاستدلالية Metaheuristics، والبرمجة المقيدة Constraint Programming، والشبكات العصبونية Neural Networks.

تعد طريقة تلوين البيان Graph Coloring من الأساليب الأولى المستخدمة لحل مشكلة الجدولة الزمنية منذ الستينيات. حيث استخدم [9] هذه التقنية لحل مشكلة الجدولة الزمنية.

البرمجة الرياضية Mathematical Programming هي طريقة أخرى تستخدم أيضاً في حل مشاكل الجدولة الزمنية. أدى التقدم التكنولوجي الكبير في أجهزة الكمبيوتر في السنوات الأخيرة إلى جعل هذه الطريقة شائعة مرة أخرى، بعد أن كان الباحثون يواجهون صعوبات كبيرة في إيجاد حلول للنماذج الرياضية.

اقترح الباحثان [10] و[13] نماذج البرمجة الخطية والبرمجة الخطية الطبيعية كنسخة مبسطة عن مشكلة الجدولة الزمنية. يمكن إدراج هذه النماذج ضمن الأساليب الأولى في البرمجة الرياضية التي أعدت لهذا الغرض. كان [10] يهدف بشكل أساسي إلى منع حدوث أي تضارب أثناء عملية التعيين.

اقترح [18] نموذج برمجة طبيعية وسمونه "مرن flexible" و "معياري modular". حيث اختبروا هذا النموذج في مدرسة ثانوية يونانية مكونة من ستة أقسام، ووجدوا الحل الأمثل الذي يلبي كل من القيود الصلبة والمرنة.

يمكن ايضاً استخدام البرمجة متعددة الأغراض Multi-Objective Programming في حل مشكلة الجدولة الزمنية بنجاح. ذكر [6] أن "طرق منهجيات صنع القرار متعددة المعايير يمكن أن تعطي نظرة هامة لمشاكل الجدولة الزمنية وتوفر المرونة في التعامل مع مختلف القيود التي لا توفرها الأساليب الحالية."

قدم [17] نموذج برمجة صفر وواحد Zero-One goal programming لتعيين أعضاء هيئة التدريس لمقررات دراسية محددة بناء على متطلبات الأقسام والتفضيلات الشخصية للمدرسين. إلى جانب تحقيق قيود العبء التدريسي، كان النموذج يقدم تفضيلات هيئة التدريس. تضمنت المشكلة التي صاغوها 87 متغير قرار و36 قيداً، واستغرق حل المشكلة أقل من 20 دقيقة على جهاز 512 K Macintosh.

اقترح [4] أيضاً نموذج برمجة صفر وواحد متعدد الأهداف multi-objective zero-one programming لجدولة المقررات الدراسية. يتكون النموذج من مرحلتين، في المرحلة الأولى تم تعيين المعلمين للمقررات الدراسية، وفي المرحلة الثانية تم تعيين مجموعات المقررات الدراسية للكلية للفترات الزمنية المناسبة. استندت المرحلة الأولى إلى النموذج الذي اقترحه [17]. تحتوي المشكلة المصاغة على 162 متغير قرار و82 قيد في المرحلة الأولى، و126 متغير قرار و76 قيد في المرحلة الثانية. تم حل كلتا المرحلتين في أقل من دقيقة واحدة. وتم تحقيق القيود الخاصة بأولوية الأحمال التدريسية بشكل جزئي وأولوية تفضيلات أوقات المحاضرات لأعضاء هيئة التدريس.

قدم [5] مرة أخرى نموذج صفر وواحد متعدد الأهداف لجدولة المقررات الدراسة. قدم هذا النموذج حلاً من مرحلة واحدة للمشكلة بدلاً من مرحلتين. حيث تم تعيين أعضاء هيئة التدريس للمقررات الدراسية وكذلك للفترات الزمنية في وقت واحد، اعتماداً على متطلبات القسم وتفضيلات أعضاء هيئة التدريس لكل من المقررات الدراسية والفترات الزمنية. إلا أنه لم يتم تحقيق تفضيلات أوقات هيئة التدريس بشكل كامل في الحل.

الطرق الاستدلالية Meta-heuristic Methods هي من المناهج الأحدث نسبياً. يتم استخدام هذه الطرق للوصول إلى الحلول المثلى دون الوقوع في مشكلة الأمثلة المحلية Local Optimum. حيث تبدأ الخوارزمية بواحد أو أكثر من الحلول الأولية، ثم يحاولون تحسين قيمة تابع الهدف مرة تلو الأخرى. يمكن إدراج كل من Tabu Search، Simulated Annealing، والخوارزميات الجينية Genetic Algorithms من بين هذه الأساليب.

قدم [8] تقنية عامة تستند إلى بحث تابو Tabu Search. اختبر الخوارزمية من خلال عدة حالات حقيقية، وتوصل إلى حلول وصفها ب "المرضية". طوروا [2] حزمة كمبيوتر تعتمد بشكل أساسي على إجراء بحث تابو. وذكروا أن الحزمة أسفرت عن نتائج مرضية من ناحية فترة الاختبار. كما اقترح [20] خوارزمية جينية لحل مشكلة الجدول الزمني للجامعة. وذكر أن النتائج التجريبية الأولى أظهرت أن نتائج الخوارزمية واعدة.

البرمجة المقيدة Constraint Programming هي أيضاً من المنهجيات الجديدة، يمكنها حل مجموعة متنوعة من المشكلات التجميعية، بما في ذلك الجدولة الزمنية، باستخدام نهج تحقيق القيود.

كما تم تطبيق منهجيات الشبكات العصبونية Neural Networks لحل مشكلة الجدولة الزمنية. قام [7] بتجربة استخدام الاستدلال القائم على الشبكات العصبونية لمشكلة الجدولة الزمنية لتوزيع المعلمين على الصفوف الدراسية.

3.2 أمثلة عن الأنظمة المشابهة

في هذه القسم تم عرض مجموعة من الأنظمة المتخصصة في إدارة الجداول الزمنية في المؤسسات التعليمية وبيان خصائص كل منها.

أ. نظام UniTime

الشكل (1) شعار نظام UniTime

وهو نظام جدولة تعليمي شامل يدعم تطوير الجداول الزمنية للمحاضرات والامتحانات [21]، وإدارة التغييرات في هذه الجداول الزمنية، ومشاركة الغرف مع الأحداث الأخرى، وجدولة الطلاب في صفوف فردية. كما إنه نظام موزع يسمح لمديري الجداول الزمنية للجامعات والأقسام المتعددة بتنسيق الجهود لبناء وتعديل جدول يلبي احتياجاتهم التنظيمية المتنوعة مع السماح بتقليل تعارض دورات الطلاب.

يشمل هذا النظام جدولة أوقات المحاضرات والامتحانات، وهو مخصص للجامعات التي تحوي عدة كليات معاً.

لا يأخذ هذا النظام قيود خاصة لضغط الجدول الزمني كما لا يدعم وجود أكثر من فئة دراسية لنفس مجموعة طلاب، كما لا يدعم اللغة العربية.

يتميز هذا النظام بأنه مجاني ومفتوح المصدر.

ب. نظام Prime Timetable

الشكل (2) شعار نظام Prime Timetable

وهو تطبيق ويب مدفوع (اشتراكات سنوية) لجدولة الحصص والمحاضرات يدعم أنظمة التشغيل المختلفة [22].

قابل للتخصيص ذو واجهات استخدام سهلة.

لا يدعم أضافة قيود خاصة بالمدرسين او المحاضرين، لا يدعم اللغة العربية.

اشتراك سنوي مدفوع قد يصل إلى $400 سنوياً.

ج. نظام Bullet TimeTabler

الشكل (3) شعار نظام Bullet TimeTabler

وهو تطبيق لسطح المكتب يعمل على MS Windows يضم عدد متنوع من القيود المختلفة، يحوي على واجهة مستخدم معقدة نسبياً، لا يدعم اللغة العربية، قادر على إدراج البيانات من ملفات خارجية وهو نظام مدفوع بنظام اشتراكات سنوية [23].

د. نظام FET

هو برنامج مجاني لجدولة الجداول الزمنية تلقائياً للمدارس أو للجامعات. يعمل على معظم أنظمة التشغيل، عادة ما يكون FET قادراً على حل جدول زمني معقد في مدة أقصاها 5-30 دقيقة [24]. للحصول على جداول زمنية أبسط، قد يستغرق الأمر وقتًا أقل. بالنسبة للجداول الزمنية الصعبة للغاية، قد يستغرق الأمر وقتًا أطول أي عدة ساعات.

يستخدم FET خوارزمية الكشفية heuristic. ووضع الأنشطة بدورها، بدءاً من أصعبها. إذا لم يتمكن من إيجاد حل، فإنه يوجهك إلى الأنشطة المستحيلة المحتملة، حتى تتمكن من تصحيح الأخطاء. تقوم الخوارزمية بتبديل الأنشطة بشكل متكرر إذا كان ذلك ممكنًا من أجل توفير مساحة لنشاط جديد، أو في الحالات القصوى، التراجع وترتيب التبديل في التقييم.

3

الفصل الثالث

الإطار النظري العام

- تمهيد
- مسألة الجدولة الزمنية
- بحوث العمليات
- أدوات تحليل وتصميم النظم

SWIFT

Timetable Management System

1.3 تمهيد

يتضمن الإطار النظري العام جميع المفاهيم النظرية التي لها علاقة بموضوع البحث. حيث يبدأ هذا الفصل بالحديث عن مسألة الجدولة الزمنية بأنواعها المختلفة ومناقشة أوجه التشابه والاختلاف فيما بينها، بالإضافة إلى التفصيل في شرح مسألة الجدولة الزمنية للمقررات الدراسية وذلك كون هذا النوع هو محور تركيز هذا البحث. بعد ذلك يتم الانتقال للحديث عن بحوث العمليات والتطبيقات المختلفة لهذا العلم، وعرض موجز للبرمجة الخطية والبرمجة الطبيعية التي تم الاعتماد عليها في بناء النموذج الرياضي في القسم العملي. يختتم هذا الفصل بلمحة شاملة عن دورة حياة تحليل وتصميم النظم والمراحل التي تمر بها بالإضافة إلى أبرز المنهجيات التي يتم الاعتماد عليها في وقتنا الحالي.

2.3 مسألة الجدولة الزمنية

تعتبر عملية وضع الجداول الزمنية للمحاضرات في المؤسسات التعليمية مثل الجامعات والمدارس من المهام الإدارية الأساسية التي لا بد أن تتكرر في كل فصل دراسي. وهي من العمليات المستهلكة للوقت والجهد بشكل كبير وخاصة في حال القيام بها بشكل يدوي، فهي تتطلب من الإدارة تحقيق مجموعة من القيود والتفضيلات التي قد تُفرض عليهم من قبل المدرسين أو بسبب محدودية الإمكانات المادية في المنشأة مثل عدد القاعات والمخابر وصالات الحاسوب وغير ذلك. وهكذا فإن صعوبة هذه المسألة تتناسب مع حجم المسألة وعدد المتطلبات المراد تحقيقها ودرجة تعقيد كل منها. نتيجة لذلك شغلت مسألة الجدولة الزمنية حيزاً كبيراً من اهتمام الباحثين في مجال بحوث العمليات والذكاء الاصطناعي في العقود الأخيرة الماضية هادفين إلى أتمتة حل هذه المسألة وبالتالي القيام بها بجهد أقل مع تحقيق أكبر قدر ممكن من المتطلبات.

يمكن تعريف مسألة الجدولة الزمنية على أنها مسألة تعيين مجموعة من الحصص الدراسية إلى عدد محدود من الفترات الزمنية [6]. وبشكل عام يمكن تصنيف مسائل الجدولة الزمنية في المؤسسات التعليمية ضمن ثلاثة أصناف رئيسية هي [1]:

- جدولة الامتحانات Examination Timetabling

جدولة امتحانات مجموعة من المقررات الدراسية مع تجنب التداخلات في امتحانات المقررات الدراسية المشتركة بين نفس الطلاب، مع زيادة الفراغات بين الامتحانات قدر الامكان بالنسبة لكل مجموعة من الطلاب.

- جدولة المقررات الدرسية Course Timetabling

إعداد الجدول الأسبوعي لمحاضرات مجموعة من المقررات الدراسية، مع عدم تداخل محاضرات المقررات التي يشترك فيها طلاب أو مدرسين، وهذا النوع هو محل اهتمام هذا البحث.

- جدولة المدارس School Timetabling

إعداد الجدول الأسبوعي لجميع الصفوف الدراسية في المدرسة، وتجنب اجتماع المعلمين بفصلين أو أكثر في نفس الوقت، أو اجتماع أكثر من معلم في فصل ما في نفس الوقت.

جدير بالذكر أن هذه التصنيفات ليست صارمة، بمعنى أننا في كثير من الحالات نواجه مسائل تقع بين هذه التصنيفات الثلاثة ويصعب تصنيفها ضمن نوع واحد من هذه الأصناف.

هناك تقاطع كبير بين خصائص مسألة جدولة الامتحانات وخصائص مسألة جدولة المقررات الدرسية، وفي كثير من الحالات من الصعوبة بمكان التمييز بين المسألتين. فجدولة الامتحانات ليست إلا عملية جدولة لمجموعة من الامتحانات (امتحان واحد لكل مقرر دراسي بدلاً من أن تكون عدة محاضرات) على فترات زمنية محددة. ومع ذلك هناك مجموعة من الخصائص لمسألة جدولة الامتحانات تميزها عن مسألة جدولة المقررات الدراسية، مثل:

- هناك امتحان واحد فقط لكل مقرر دراسي.
- قيود التضارب بين الامتحانات صارمة جداً فلا يمكن بشكل من الأشكال أن يكون للطالب امتحانين في نفس الوقت. أما في حالة المقررات الدرسية فيمكن قبول أنو يكون لدى أحد الطلاب تضارب في أوقات المحاضرات (في حالات محددة).
- هناك قيود مختلفة، على سبيل المثال يمكن أن يكون هناك امتحان واحد كحد أقصى في اليوم لكل طالب (وليس عدة امتحانات متتالية كما هو الحال في المحاضرات).
- قد يختلف عدد الفترات اليومية، أما في مسألة جدولة المقررات الدرسية فيكون عددها ثابت في جميع الأيام.
- يمكن أن يكون هناك أكثر من امتحان في قاعة واحدة في نفس الوقت.

تشترك جميع أنواع مسائل الجدولة الزمنية بمجموعة من الخصائص. يعد فهم هذه الخصائص أمراً أساسياً لمعالجة أي مشكلة جدولة مهما كان نوعها. لذلك سنقوم بتوضيح هذه الخصائص في القسم التالي.

1.2.3 خصائص مسألة الجدولة الزمنية

تبحث مسائل الجدولة الزمنية عن حلول مقبولة يمكن تطبيقها على أرض الواقع، ومن أجل ذلك على هذه الحلول أن تحقق مجموعة من القيود التي يمكن تصنيفها بصنفين أساسيين هما القيود الصلبة والقيود المرنة.

يقصد بالقيود الصلبة مجموعة القيود التي يجب أن تتحقق جميعها لكي يكون الحل مقبولاً. أما القيود المرنة هي تلك التي يُرغب بتحققها ولكنها لا تؤثر على صحة الحل وقابليته للتطبيق [11].

بعض الأمثلة عن القيود الصلبة Hard Constraints:

- المحاضرة يجب أن تحدث مرة ومرة واحدة فقط كل أسبوع.
- يدرس المدرس محاضرة واحدة على الأكثر في نفس الفترة الزمنية.
- تحوي القاعة التدريسية حصة دراسية واحدة على الاكثر في نفس الفترة الزمنية.
- لا يمكن أن يسند أكثر من حصة درسية واحدة لنفس مجموعة الطلاب في فترة زمنية واحدة.
- القاعة التدريسية يجب أن تكون مناسبة للحصة الدراسية سواء من حيث الحجم أو النوع.
- تسند الحصص الدراسية للمدرس فقط في الفترات المتاحة بالنسبة له.

بعض الأمثلة عن القيود المرنة Soft Constraints:

- محاضرات بعض المقررات الدراسية يفضل أن تكون في أوقات صباحية.
- محاضرات بعض المقررات الدراسية يفضل أن يتم جدولتها قبل محاضرات مقررات أخرى.
- من الأفضل الاعتماد على بعض القاعات التدريسية دون غيرها.
- تخفيض عدد الفراغات بين الحصص الدرسية للحد الأدنى (بالنسبة للمدرسين أو للطلاب).
- تخفيض عدد أيام عمل المدرسين.
- تخفيض عدد أيام دوام الطلاب.

إن القيود الصلبة هي التي تحدد فيما إذا كان الحل مقبول أو غير مقبول. فإن كان الحل مقبولاً فالقيود المرنة هي التي تحدد جودة هذا الحل.

بشكل عام يمكن التعامل مع مسألة الجدولة على أنها مسألة اختيار بديل (جدول زمني) من عدة بدائل متاحة (جميع الحالات الممكنة)، ولكن من الناحية التطبيقية فإن هذا صعب جداً في أغلب الحالات. لذلك جاءت الحاجة للنظر إلى هذه المسألة على أنها مسألة أمثلة، وتم اقتراح طرق عديدة لحل هذه المسألة منها طرق رياضية ومنها طرق استدلالية [26].

تم إثبات أن مسائل الجدولة الزمنية صعبة سواء من الناحية العملية التطبيقية أو من الناحية النظرية. قد يكون تعقيد المسألة غير كبير (من درجة $O(E\ logE)$ حيث E هو عدد العقد في بيان مضاعف) إلا أن المتطلبات الأخرى التي غالباً ما يُرغب بتحقيقها هي التي تزيد من درجة تعقيد المسألة بشكل كبير لتصبح من مرتبة NP-Complete أو NP-Hard [19].

إن هذا البحث يتمحور حول مسألة جدولة المقررات الدرسية في الجامعات، أما النوعين الآخرين فهما خارج نطاق هذا البحث. ونظراً لذلك سيتم تناول هذه المسألة بشيء من التفصيل في القسم التالي.

2.2.3 مسألة الجدولة الزمنية للمقررات الدراسية

يمكن النظر لمسألة الجدولة الزمنية للمقررات الدراسية على أنها مسألة تعيين متعددة الأبعاد. حيث يتم فيها تعيين مجموعة من المحاضرات (المحاضرة هي اجتماع بين مجموعة من الطلاب ومدرس) على القاعات الدراسية والأوقات [12].

يختلف تعريف المشكلة والمصطلحات المستخدمة في وصفها بين مؤسسة وأخرى إلا أن المفاهيم التالية شائعة الاستخدام بين معظم المؤسسات التعليمية:

المقرر الدراسي Course: يشير المقرر الدراسي الى موضوع معين يتم تدريسه من خلال محاضرة أو عدة محاضرات أسبوعية خلال السنة أو الفصل الدراسي. كما يمكن أن يشترك عدة مدرسين في إعطاء نفس المقرر.

الصف الدراسي Class: يشير الصف الدراسي إلى مجموعة من الطلاب يأخذون نفس المجموعة من المقررات الدراسية وعادةً ما يبقون معاً طوال الأسبوع.

البرنامج الدراسي Program: يشير إلى مجموعة من المقررات الدراسية الاجبارية ومجموعة من المقررات الدراسية الاختيارية التي يجب أن يأخذها الطلاب الراغبون بالحصول على شهادة معينة.

يمكن تقسيم مسألة الجدولة للمقررات الدراسية الى خمسة مسائل فرعية. بالطبع ليست بالضرورة أن تكون جميعها متعلقة بنفس الحالة:

1- جدولة المقررات الدراسية Course Timetabling:

تعيين المقررات الدراسية على الفترات الزمنية. قد تكون هناك عدة قيود جانبية مثل احترام هيكلية البرنامج الدراسي بالإضافة إلى توفر أو عدم توفر القاعات والمعلمين في أوقات محددة.

في بعض الحالات يجب أن يتم توزيع محاضرات بعض المقررات الدراسية على أوقات الأسبوع بشكل محدد. على سبيل المثال، تطلب بعض المؤسسات التعليمية أن يتم وضع جميع محاضرات مقرر دراسي في وقت واحد. أو على العكس، قد تطلب توزيع محاضرات مقرر دراسي على أكثر من يوم من أيام الأسبوع. كما يمكن أن يتم تحديد قيود تحدد الساعات الدراسية التي يمكن للمدرس أو الطلاب الحصول عليها في اليوم الواحد.

2- جدولة مدرس لصف Class-Teacher Timetabling:

نجد هذه المشكلة غالباً في المدارس حيث تكون وحدة الجدولة هي الصف الدراسي class وليس الطالب. حيث نفترض هنا أن عملية ربط المدرسين مع المقررات الدراسية والصفوف قد تمت مسبقاً. ولكن تبقى المشكلة بتحديد أزمنة كل حصة دراسية دون أن يحدث أي تعارض.

3- جدولة الطلاب Student Scheduling:

نواجه هذه المشكلة عندما يتم تدريس المقررات الدراسية في أقسام متعددة. بمجرد أن يختار الطلاب مقرراتهم الدراسية، يجب تصنيفهم في أقسام. الهدف هو توفير جداول ذات جودة جيدة (خالية من التعارض) للطلاب مع الموازنة بين أحجام فئات الطلاب وسعة القاعات الدراسية.

4- تعيين المعلمين Teacher Assignment:

تكمن المشكلة هنا في توزيع المعلمين على المقررات الدراسية مع تحقيق أكبر قدر ممكن من تفضيلاتهم.

5- تعيين القاعات الدراسية Classroom Assignment:

يجب تحديد قاعة دراسية محددة لكل محاضرة مع الأخذ بعين الاعتبار نوع القاعة المناسب والحجم والموقع وتفضيلات المنشأة وقيودها. بالطبع تتم عملية التعيين هذه بعد تعيين المحاضرات إلى فترات زمنية محددة. من الناحية المثالية، يجب تعيين الأوقات والقاعات في وقت واحد، لكن العديد من المؤسسات التعليمية تفصل بين العمليتين بهدف تبسيط المسألة وتخفيض فترة المعالجة.

3.3 بحوث العمليات

يقدم هذا القسم مقدمة مختصرة حول بحوث العمليات، ثم ينتقل إلى التعريف بالبرمجة الرياضية حيث يتناول كل من البرمجة الخطية والبرمجة الطبيعية بشكل موجز وواضح.

تسمح تقنيات بحوث العمليات باتخاذ ليس فقط قرارات جيدة، بل قرارات مثالية لمشكلات القرار العملية. وبالتالي فإن لبحوث العمليات طيف واسع من التطبيقات العملية مثل الخدمات اللوجستية، إدارة سلسلة التوريد، إدارة الإيرادات، جدولة أطقم العمل، جدولة الإنتاج، تصميم شبكات الشحن، الجدولة الزمنية، وتحديد مواقع المنشآت، وذلك على سبيل المثال لا الحصر. يستخدم الباحثون أساليب بحوث العمليات لتحليل المشكلة -عادة ما يتم ذلك باستخدام أدوات رياضية متقدمة- وتحديد أفضل مسار لحل هذه المشكلة.

إن النماذج الرياضية المكونة من المتغيرات والقيود من الممكن أن تعبر عن مشكلة القرار. حيث تمثل المتغيرات القرارات، والقيود تفرض شروطاً على القرارات. عادة تتكون المشكلة من القيود الصلبة والمرنة، حيث تشكل القيود الصلبة الشروط التي يجب أن يحققها الحل. فإذا كان الحل ينتهك أي قيد من القيود الصلبة فإنه يعتبر غير قابل للتنفيذ وبالتالي غير مقبول. تمثل القيود المرنة تفضيلات مرتبطة بخصائص الحل، أي ميزات الحل المرغوبة أو غير المرغوب بها. غالباً ما تتضمن المشكلة تابع هدف يقيس تأثير قرار معين، وعادةً ما يستخدم مفهوم الانتهاك المرجح Weighted violation للقيود المرنة. وليس غريباً أن يتكون تابع الهدف من عدة أهداف متضاربة [15].

1.3.3 البرمجة الخطية Linear Programming

تشير البرمجة الرياضية والأمثلة إلى اختيار الحل الأفضل من مجموعة البدائل المتاحة المرتبطة بمشكلة ما [16]. وفي مسائل الأمثلة نحاول عادة تعظيم أو تصغير تابع الهدف الخاص بالمشكلة، ويتم ذلك من خلال تغيير قيم متغيرات القرار التي تحدد بدورها سلوك النظام. تخضع متغيرات القرار لمجموعة من القيود والتي يجب أن تتحقق جميعها لكي يكون الحل مقبولاً.

البرمجة الخطية هي أحد فروع البرمجة الرياضية التي تعتمد على أدوات رياضية لتحديد الحل الأفضل من بين مجموعة بدائل. ما يميز البرنامج الخطي هو أن تابع الهدف ومجموعة القيود هي كثيرات حدود من الدرجة الأولى، بالإضافة إلى أن جميع المتغيرات تنتمي إلى مجموعة الأعداد الحقيقية.

يتكون البرنامج الخطي بشكل عام من أربعة أجزاء رئيسية:

- **متغيرات القرار Decision Variables**

متغير القرار هو المتغير الذي يتم تحديد قيمته من خلال النموذج وهو المؤثر الأساسي على أداء النظام.

- **تابع هدف Objective function**

هو كثير حدود من الدرجة الأولى يتكون من متغيرات القرار وقد يكون الهدف هو تعظيم هذا التابع Maximization أو تصغيره Minimization

- **مجموعة من القيود Constraints**

تقيد القيم التي تأخذها متغيرات القرار، وهي التي تحدد فضاء الحلول المقبولة Feasible region.

- **قيود اللاسلبية Non-negativity constraints**

مجموعة من القيود التي تمنع متغيرات القرار من أن تأخذ قيماً سالبة

وتكون الصيغة المعيارية للبرنامج الخطي كما يلي:

$$\begin{aligned} Max \quad & C^T X \\ & A X = b \\ & X \geq 0 \end{aligned}$$

إن الفضاء الذي تولده مجموعة القيود هو ما يسمى بفضاء الحلول المقبولة. وهنا نميز ثلاث حالات [14]:

1. إن كان فضاء الحلول خالي نقول إن المسألة غير قابلة للحل Infeasible.
2. إن لم يكن فضاء الحلول خالي فإننا نسمي الحل الذي يعظم قيمة دالة الهدف للحد الأقصى - بحالة ال Maximization - بالحل الأمثل Optimal solution.
3. إن لم يكن فضاء الحلول خالي ولم يكن هناك حل أمثلي نقول بأن المسألة غير محدودة Unbounded.

توضيحاً للمفاهيم التي تم ذكرها سنقوم بعرض مثال توضيحي. لنفترض أنه هناك بائع تجزئة يريد شراء مجموعة من السلع ومن ثم بيعها في متجره، وأن هذه السلع هي A وB، إن الربح المتوقع من بيع كل وحدة من السلعتين هو 10$ و8$ وحدات نقدية على التتالي. إلا أنه لا يمكن تحميل عربة النقل إلا ب$100\ kg$، وكذلك لا يمكن أن تحتوي أكثر من$16\ m^3$، فإذا علمنا أن أوزان السلع هو على التتالي $12\ kg$ و $8\ kg$، وأحجامها $1.2\ m^3$ و $1.6\ m^3$. فما هي الكمية المثلى التي يجب أن يشتريها البائع من كل سلعة لكي يحقق أكبر ربح ممكن دون أن يخترق قيود الحجم والوزن؟

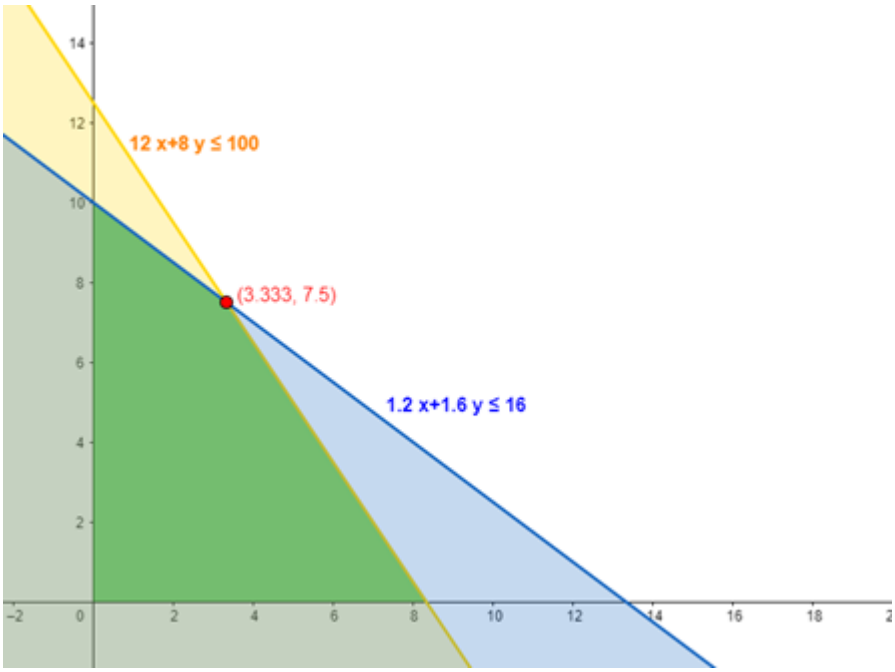
أسم السلعة	A	B
الربح المتوقع	10 \$	8 \$
وزن السلعة	$12\ kg$	$8\ kg$
حجم السلعة	$1.2\ m^3$	$1.6\ m^3$

الجدول (1) معطيات المسألة

نفترض أن X_A هو عدد الوحدات المشتراة من المنتج A و X_B هو عدد الوحدات المشتراة من المنتج B.
يمكن التعبير عن المسألة بالبرنامج الخطي التالي:

$$\begin{aligned} Max \quad & 10X_A + 8X_B \\ & 12X_A + 8X_B \leq 100 \\ & 1.2X_A + 1.6X_B \leq 16 \\ & X_A, X_B \geq 0 \end{aligned}$$

ومن خلال التمثيل البياني الموضح بالشكل أدناه نجد أن قيمة تابع الهدف عند الحل الأمثل هي $(Z = \frac{280}{3})$
وهذا هو أكبر ربح يمكن أن يحققه البائع ويكون الحل $X_B = \frac{15}{2}$ و $X_A = \frac{10}{3}$



الشكل (4) التمثيل البياني للمسألة

نلاحظ أن متغيرات القرار في المثال السابق أخذت قيماً كسرية عند الحل الأمثل، ولكن ذلك غير مقبول في حالتنا هذه وحتى في حالات أخرى كثيرة، فلا يمكن للبائع على سبيل المثال شراء 7.5 وحدات من المنتج B. وفي حالات كهذه تظهر الحاجة لما يسمى بالبرمجة الطبيعية Integer Programming.

هناك أنواع عديدة للبرمجة الرياضية التي يميز كل منها مجموعة من الخصائص نذكر منها:

- البرمجة الخطية Linear Programming

- البرمجة الخطية الطبيعية Integer Programming or Linear Integer Programming
تماما كالبرمجة الخطية ولكن المتغيرات تنتمي إلى مجموعة الأعداد الطبيعية عند الحل الأمثل.

- البرمجة الطبيعية البوليانية Binary Integer Programming
هي حالة خاصة من البرمجة الخطية الطبيعية حيث تكون جميع المتغيرات تأخذ القيمة واحد أو صفر {0,1}

- البرمجة الطبيعية المختلطة Mixed-integer Programming
حالة خاصة من البرمجة الخطية الطبيعية ولكن المتغيرات قد تكون حقيقية أو طبيعية (أو بوليانية)

- البرمجة الغير خطية Non-linear Programming
القيود وتابع الهدف كثيرات حدود من الدرجة الثانية فما فوق والمتغيرات تنتمي لمجموعة الأعداد الحقيقية Real Numbers.

- البرمجة الغير خطية الطبيعية Nonlinear Integer Programming
القيود وتابع الهدف كثيرات حدود من الدرجة الثانية فما فوق والمتغيرات تنتمي لمجموعة الأعداد الطبيعية Integer.

2.3.3 البرمجة الطبيعية المختلطة Mixed-Integer Programming

تمثل البرمجة الطبيعية IP طريقة شائعة وفعالة ضمن نطاق بحوث العمليات. نموذج البرمجة الطبيعية هو نموذج أمثلة رياضي حيث تنتمي جميع متغيراته لمجموعة الأعداد الطبيعية. يشمل نموذج الأمثلة عدد من القيود والمتغيرات وتابع الهدف. الهدف هو إيجاد حل يحقق القيود الصلبة ويقلل أو يزيد من تابع الهدف لأقصى حد ممكن، وذلك على حسب المشكلة. إذا كانت المشكلة تسمح لمجموعة فرعية من المتغيرات أن تكون مستمرة، فإن النموذج هو نموذج برمجة طبيعية مختلطة. البرمجة الطبيعية المختلطة هي أحد فروع البرمجة الطبيعية ويرمز لها MIP أو MILP.

من خلال تخفيف Relaxing الشروط التكاملية لجميع المتغيرات بحيث تكون متغيرات مستمرة، يصبح النموذج عبارة عن برنامج خطي LP، وبشكل أكثر دقة يصبح نموذجاً مخففا من البرنامج الطبيعي LP-Relaxation. عادة ما يكون حل LP-Relaxation أسهل بكثير من حل نموذج MIP الأصلي. يوفر حل النموذج المخفف حداً أعلى أو أدنى Upper\Lower Bound لقيمة تابع الهدف عند الحل الأمثل. وبالتالي

يمثل هذا الحد حداً نظرياً لأفضل حل ممكن للمشكلة. وبالتالي يسمح الحد بتقييم جودة حل معروف من خلال النظر إلى الفجوة بين قيمة الحل المعروف وقيمة الحد الأعلى أو الأدنى. الحل الذي تتساوى قيمته وقيمة الحد هو حل أمثل. وبالتالي يمكن للبرمجة الطبيعية المختلطة MIP إيجاد حل أمثلي للمشكلات وإثبات أمثلية هذا الحل، وبالتالي تنتمي البرمجة الطبيعية المختلطة MIP إلى عائلة الأساليب الدقيقة Exact methods [15].

يتم حل نماذج البرمجة الطبيعية المختلطة MIP باستخدام محركات حل MIP Solvers، والتي تستخدم بشكل مكثف مفهوم البرنامج الخطي المخفف LP-relaxations وفجوات الأمثلة Optimality gap. التقنية الأساسية التي تستخدمها محركات الحل MIP Solvers هي خوارزمية التفرع والحد Branch & Bound، والتي تحلل المشكلة إلى مشاكل فرعية ليتم حلها بشكل فردي. أي يقوم محرك الحل بحل ال LP-relaxation لكل مشكلة فرعية للحصول على حدود وحلول مقبولة، والتي يستخدمها لقطع (حذف) فروع شجرة البحث التي ليس من الممكن أن تعطي نتائج أفضل من الحد الذي تم الحصول عليه. أثناء حل نموذج البرمجة الطبيعية المختلطة MIP يتم تحسين الحد بانتظام، وحتى إذا لم يتم التوصل إلى الحل الأمثل، يظل الحد النهائي يوفر معلومات قيمة حول المشكلة والحلول المعروفة. جدير بالذكر أنه في الواقع العملي وفي كثير من الحالات قد يصل محرك الحل إلى الحل الأمثل بمدة قصيرة نسبياً إلا أنه يستغرق زمنا كبير جدا لإثبات أمثلية هذا الحل، بعبارة أخرى، يستغرق وقت طويل حتى يتطابق الحد الذي أوجده محرك الحل مع قيمة الحل.

هناك طيف واسع من المسائل التي يمكن حلها باستخدام البرمجة الطبيعية المختلطة نذكر منها:

- مسائل اختيار موقع المنشأة Facility location problems

هناك عدة أنواع لهذه المسألة، وغالبا ما يتم تصنيفها وفقا لتابع الهدف. يمكن تمييز 3 أصناف رئيسية لهذه المسألة وهي:

1. مسألة تغطية مجموعة من نقاط الطلب Set covering problems: ويقصد بها بناء أقل عدد ممكن من المنشآت لتغطية كامل الطلب.
2. مسائل التغطية العظمى Maximum covering problems: بناء عدد محدد من المنشآت لتغطية أكبر قدر ممكن من الطلب.
3. مسائل اختيار الموقع بناء على التكاليف الثابتة Fixed charge location problems: إيجاد التوازن بين العائد المتوقع من تغطية الطلب وبين التكلفة المترتبة على بناء المنشأة.

- مسائل جدولة الآلة Machine scheduling problems

بشكل عام نكون أمام مسألة جدولة الالة عندما يكون لدينا مجموعة من المهام التي يجب توزيعها على مجموعة من الآلات. ويمكن تصنيف هذا النوع من المسائل ضمن أربعة أنواع رئيسية:

1. Single machine serial production
2. Multiple parallel machines
3. Flow shop problems
4. Job shop problems

- مسائل اختيار المسار الأمثل Vehicle routing problems

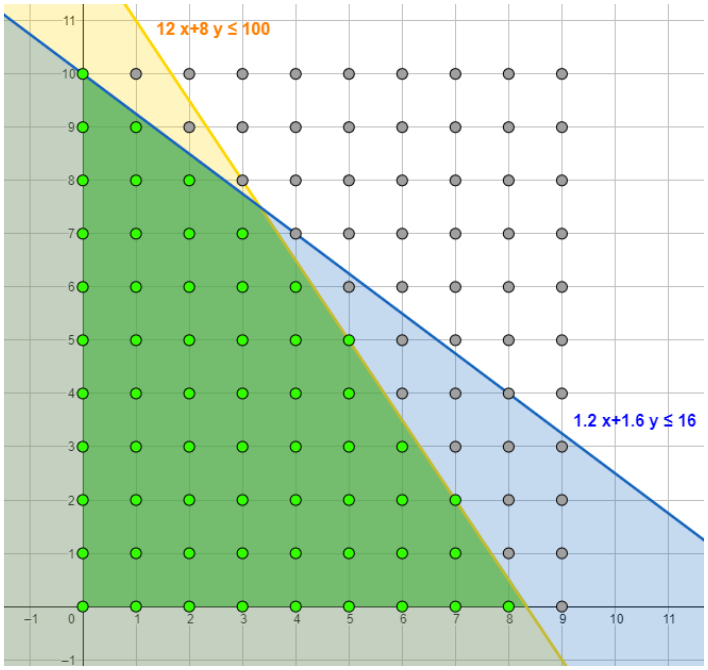
ومن أشهر هذه المسائل مسألة البائع الجوال Traveling salesperson problem

بالإضافة لمسائل كثير أخرى يمكن حلها باستخدام البرمجة الطبيعية مثل مسألة الجدولة الزمنية للمقررات الدرسية في الجامعات والتي تعتبر محور هذا البحث.

بالعودة إلى المثال السابق مع إضافة بعض القيود على طبيعة متغيرات القرار يصبح النموذج كالتالي:

$$\begin{aligned} Max \quad & 10X_A + 8X_B \\ & 12X_A + 8X_B \leq 100 \\ & 1.2X_A + 1.6X_B \leq 16 \\ & X_A, X_B \geq 0 \; and \; Integer \end{aligned}$$

نلاحظ أن الحل السابق ($X_B = \frac{15}{2}$ و $X_A = \frac{10}{3}$) لم يعد ضمن فضاء الحلول المقبولة. وأصبح الحل الأمثل في هذه الحالة هو ($X_B = 5$ و $X_A = 5$) وتكون قيمة تابع الهدف عند هذا الحل ($Z = 90$).



الشكل (5) التمثيل البياني الجديد للمسألة

4.3 أدوات تحليل وتصميم النظم

في هذه القسم تم عرض دراسة نظرية مختصرة حول دورة حياة تطوير النظم والمراحل التي تمر فيها. بالإضافة إلى مناقشة أهم منهجيات تحليل وتصميم النظم المنتشرة في وقتنا الحالي وبيان ميزات ومساوئ كل منها.

1.4.3 دورة حياة تطوير النظم Systems Development Life Cycle

تعرف دورة حياة تطوير النظم بأنها مجموعة الإجراءات التي تمكننا من الانتقال من النظام بوضعه القديم سواء أكان نظاماً الكترونياً أم تقليدياً إلى النظام الجديد سواء أكان نظاماً الكترونياً أم تقليدياً.

يسمى النظام بوضعه القديم قبل عملية التطوير ب As-Is System وهو النظام بوضعه الحالي والذي يتم استخدامه من قبل المستخدم؛ ويسمى النظام الجديد وهو النظام الذي يطمح المستخدم للوصول إليه من خلال عملية التطوير التي سوف تنفذ ب To-Be System وهو النظام المراد صنعه انطلاقاً من المتطلبات التي تم جمعها من المستخدم.

كما عرف الريس في بحثه [27] دورة حياة تطوير النظم بأنها مجموعة من الإجراءات التي تم تقسيمها إلى مراحل واضحة يتم من خلالها تطوير نظام تقليدي أو الكتروني يساهم في تحقيق أهداف ومتطلبات المستخدم بالجودة المطلوبة ويساهم في حل مشكلة ما قد يعاني منها النظام السابق أو المستخدم؛ أو يساهم في تحقيق منافع اقتصادية أو اجتماعية من خلال تطويره.

ومن التعريفات السابقة يمكننا تعريف دورة حياة تطوير النظام بأنه مجموعة الخطوات والمراحل التي من خلالها نقوم بتحويل نظام ما إلى نظام الكتروني يحقق الغايات والأهداف المطلوبة منه.

1.1.4.3 مراحل دورة حياة تطوير النظام

تمر دورة حياة تطوير النظام بأربعة مراحل رئيسية: وهي التخطيط، التحليل، التصميم والتنجيز [25]. حيث تتألف كل مرحلة من هذه المراحل من عدة خطوات تعتمد كل منها على مجموعة من التقنيات، وتنتج وثـائق مخصصة تعطي في مجموعها فهماً أوسع للنظام.

1. التخطيط Planning

مرحلة التخطيط هي العملية الأساسية لفهم سبب بناء نظام المعلومات وتحديد كيفية عمل فريق المشروع على بنائه. تتألف هذه المرحلة من مرحلتين:

المرحلة الأولى: أثناء بدء المشروع، يتم تحديد قيمة عمل النظام بالنسبة للمؤسسة (مثلاً: كيف سيخفض النظام الجديد التكاليف أو يزيد الإيرادات؟)، تأتي معظم الأفكار للأنظمة الجديدة من خارج منطقة نظام المعلومات (من قسم التسويق، قسم المحاسبة، إلخ) على شكل طلب نظام.

حيث يقدم طلب النظام ملخصاً موجزاً لاحتياجات العمل، ويوضح كيف سيخلق النظام الذي يدعم الحاجة قيمة تجارية. يعمل قسم نظم المعلومات مع الشخص أو القسم الذي يقوم بإنشاء الطلب (يسمى راعي المشروع) لإجراء تحليل الجدوى. يفحص تحليل الجدوى الجوانب الرئيسية للمشروع المقترح:

- الجدوى الفنية (هل يمكننا أن نبني النظام؟)
- الجدوى الاقتصادية (هل سيوفر قيمة تجارية؟)
- الجدوى التنظيمية (إذا قمنا ببنائه، فهل سيتم استخدامه؟)

يتم تقديم طلب النظام وتحليل الجدوى إلى لجنة الموافقة على أنظمة المعلومات (تسمى أحياناً لجنة التوجيه)، والتي تقرر ما إذا كان ينبغي تنفيذ المشروع.

المرحلة الثانية: بمجرد الموافقة على المشروع، تتم الآن بداية إدارة المشروع حيث يقوم مدير المشروع بإنشاء خطة عمل، وتعيين موظفين للمشروع، ووضع تقنيات لمساعدة فريق المشروع على التحكم في المشروع وتوجيهه. ويكون خرج هذه العملية هو خطة مشروع تصف كيف سيبدأ فريق المشروع في تطوير النظام الجديد.

2. التحليل Analysis

تجيب مرحلة التحليل على الأسئلة التالية:

- من سيستخدم النظام؟
- وماذا سيفعل النظام؟
- أين ومتى سيتم استخدامه؟

خلال هذه المرحلة، يقوم فريق المشروع بمراجعة النظام الحالي، وتحديد فرص التحسين، وتطوير مفهوم للنظام الجديد. تتكون هذه المرحلة من ثلاث خطوات:

الخطوة الأولى: يتم في هذه المرحلة تطوير استراتيجية تحليل لتوجيه جهود فريق المشروع. تتضمن هذه الإستراتيجية عادةً دراسة النظام الحالي ومشاكله، وتصور طرق لتصميم النظام الجديد.

الخطوة الثانية: تتم في هذه المرحلة جمع المتطلبات من خلال المقابلات أو ورش العمل الجماعية أو الاستبيانات. يؤدي تحليل هذه المعلومات جنباً إلى جنب مع مدخلات راعي المشروع والعديد من الأشخاص الآخرين إلى تطوير مفهوم لنظام جديد. ثم يتم استخدام مفهوم النظام كأساس لتطوير مجموعة من نماذج تحليل الأعمال التي تصف كيفية عمل الأعمال إذا تم تطوير النظام الجديد. تتضمن المجموعة نماذج تمثل البيانات والعمليات اللازمة لدعم عملية الأعمال الأساسية.

الخطوة الثالثة: يتم دمج التحليلات ومفهوم النظام والنماذج في مستند يسمى اقتراح النظام System Proposal، والذي يتم تقديمه إلى راعي المشروع وصناع القرار الرئيسيين الآخرين (على سبيل المثال، أعضاء لجنة الموافقة) الذين سيقررون ما إذا كان يجب الاستمرار في تطوير المشروع أم لا.

3. التصميم Design

تحدد مرحلة التصميم كيفية عمل النظام من حيث الأجهزة والبرمجيات والبنية التحتية للشبكة التي ستكون موجودة بالإضافة إلى واجهة الاستخدام والنماذج والتقارير التي سيتم استخدامها والبرامج المحددة وقواعد المعطيات والملفات المطلوبة. على الرغم من أن معظم القرارات الاستراتيجية حول النظام يتم اتخاذها في تطوير مفهوم النظام أثناء مرحلة التحليل، فإن الخطوات في مرحلة التصميم تحدد كيفية عمل النظام بشكل أكثر دقة.

تتكون مرحلة التصميم من أربع خطوات:

الخطوة الأولى: يجب في هذه الخطوة تحديد استراتيجية التصميم حيث توضح هذه الاستراتيجية ما إذا كان سيتم تطوير النظام من قبل المبرمجين التابعين للشركة أو سيتم الاستعانة بمصادر خارجية لتطويره، أو إذا كانت الشركة ستشتري حزمة برامج جاهزة.

الخطوة الثانية: تؤدي الخطوة السابقة إلى تطوير التصميم المعماري الأساسي للنظام الذي يصف الأجهزة والبرامج والبنية التحتية للشبكة التي سيتم استخدامها. في معظم الحالات، سيضيف النظام إلى البنية التحتية الموجودة بالفعل في المؤسسة أو يغيرها. يحدد تصميم الواجهة كيف سينتقل المستخدمون عبر النظام (مثلاً عن طريق القوائم والأزرار التي تظهر على الشاشة)، والنماذج والتقارير التي سيستخدمها النظام.

الخطوة الثالثة: تتم في هذه المرحلة تحديد نوع قاعدة المعطيات والملفات وأماكن تخزينها التي يتعامل معها النظام.

الخطوة الرابعة: يقوم فريق التحليل في هذه المرحلة بتطوير تصميم البرامج وتحديد البرامج التي يجب كتابتها وماذا سيفعله كل برنامج بالضبط.

هذه المجموعة من المخرجات (التصميم المعماري، تصميم الواجهة، مواصفات الملفات وقاعدة المعطيات، وتصميم البرنامج) هي مواصفات النظام التي يستخدمها فريق البرمجة للتنفيذ. في نهاية مرحلة التصميم يتم إعادة فحص ومراجعة تحليل الجدوى وخطة المشروع، ويتم اتخاذ قرار آخر من قبل راعي المشروع ولجنة الموافقة حول ما إذا كان سيتم إنهاء المشروع أو الاستمرار فيه.

4. التنجيز Implementation

المرحلة النهائية في دورة حياة تطوير النظم هي مرحلة التنجيز، والتي يتم خلالها بناء النظام بالفعل (أو شراؤه). تحظى هذه المرحلة عادةً بأكبر قدر من الاهتمام وذلك لأنها بالنسبة لمعظم الأنظمة هي أطول جزء منفرد من عملية التطوير والأكثر كلفة.

تتألف هذه المرحلة من ثلاث خطوات:

الخطوة الأولى: بناء النظام هو الخطوة الأولى. حيث يتم بناء النظام واختباره للتأكد من أنه يعمل حسب التصميم. نظراً لأن تكلفة إصلاح الأخطاء قد تكون هائلة، تعد مرحلة الاختبار أحد أهم خطوات التنفيذ.

الخطوة الثانية: تتم في هذه الخطوة تثبيت النظام. التثبيت هو العملية التي يتم من خلالها إيقاف تشغيل النظام القديم وتشغيل النظام الجديد. هناك العديد من الطرق التي يمكن استخدامها للتحويل من النظام القديم إلى النظام الجديد. من أهم جوانب التحويل خطة التدريب، التي تستخدم لتعليم المستخدمين كيفية استخدام النظام الجديد والمساعدة في إدارة التغييرات التي يسببها النظام الجديد.

الخطوة الثالثة: يضع فريق التحليل خطة دعم للنظام. تتضمن هذه الخطة عادة مراجعة رسمية أو غير رسمية لما بعد التنجيز، بالإضافة إلى طريقة منهجية لتحديد التغييرات الرئيسية والثانوية اللازمة للنظام.

2.4.3 منهجيات تحليل وتصميم النظم

كما ناقشنا في القسم السابق، توفر دورة حياة تطوير الأنظمة (SDLC) الأساس للعمليات المستخدمة لتطوير نظام المعلومات. يعبر مصطلح "المنهجية" على الطريقة الرسمية لتنفيذ مراحل دورة حياة تطوير الأنظمة (أي أنها قائمة بالخطوات والمخرجات). هناك العديد من منهجيات تطوير الأنظمة، وهي تختلف من حيث التقدم الذي يتم اتباعه خلال مراحل دورة حياة تطوير الأنظمة. تمتلك العديد من المنظمات منهجياتها الداخلية الخاصة التي تم تنقيحها على مر السنين، وهي تشرح بالضبط كيف يتم تنفيذ كل مرحلة من مراحل دورة حياة تطوير الأنظمة في تلك الشركة. سنعرض في هذا القسم العديد من المنهجيات السائدة التي تطورت عبر الزمن.

1.2.4.3 التصميم البنيوي Structures Design

تتبنى منهجيات التصميم البنيوي مقاربة صورية لدورة حياة تطوير النظم تعتمد سياسة الخطوة خطوة، بمعنى أن يجري الانتقال بشكل منطقي من مرحلة إلى المرحلة التي تليها. وتعتبر هذه المنهجيات الرائدة في إدخال النمذجة الصورية وتقنيات رسم المخططات لوصف إجرائيات العمل الأساسية في النظام والمعطيات التي تدعمها. وتنقسم إلى نوعين: التطوير الشلالي والتطوير على التوازي.

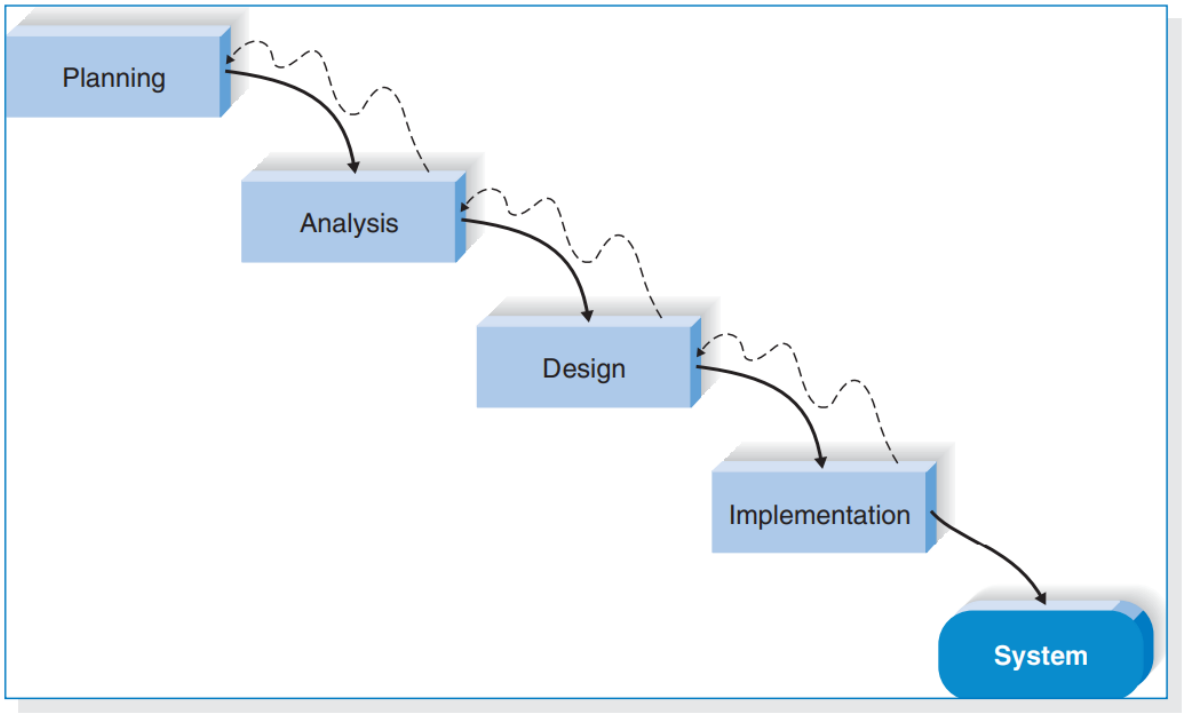
أ. التطوير الشلالي Waterfall-Development

ينتقل المحللون والمستخدمون في هذه المنهجية انتقالاً تتابعياً من مرحلة لأخرى .

عادةً ما تكون المخرجات الرئيسية لكل مرحلة ضخمة ويتم تقديمها إلى لجنة الموافقة وراعي المشروع للموافقة عليها مع انتقال المشروع من مرحلة إلى أخرى. بمجرد الموافقة على العمل المنتج في مرحلة واحدة، تنتهي المرحلة وتبدأ المرحلة التالية. مع تقدم المشروع من مرحلة إلى أخرى فإنه يتحرك للأمام بنفس طريقة الشلال. في حين أنه من الممكن العودة إلى الوراء خلال المراحل (على سبيل المثال، من التصميم إلى التحليل)، إلا أن ذلك صعب للغاية. (تخيل نفسك كسمك السلمون يحاول السباحة عكس التيار في شلال).

تتميز منهجيات تطوير الشلال بمزايا تحديد المتطلبات قبل وقت طويل من بدء البرمجة وتقييد التغييرات على المتطلبات أثناء تقدم المشروع. وتتمثل العيوب الرئيسية في أنه يجب تحديد التصميم تماماً قبل بدء البرمجة، ويمر وقت طويل بين الانتهاء من اقتراح النظام في مرحلة التحليل وتسليم النظام، ويتم التعامل مع الاختبار على أنه فكرة لاحقة تقريباً في مرحلة التنفيذ.

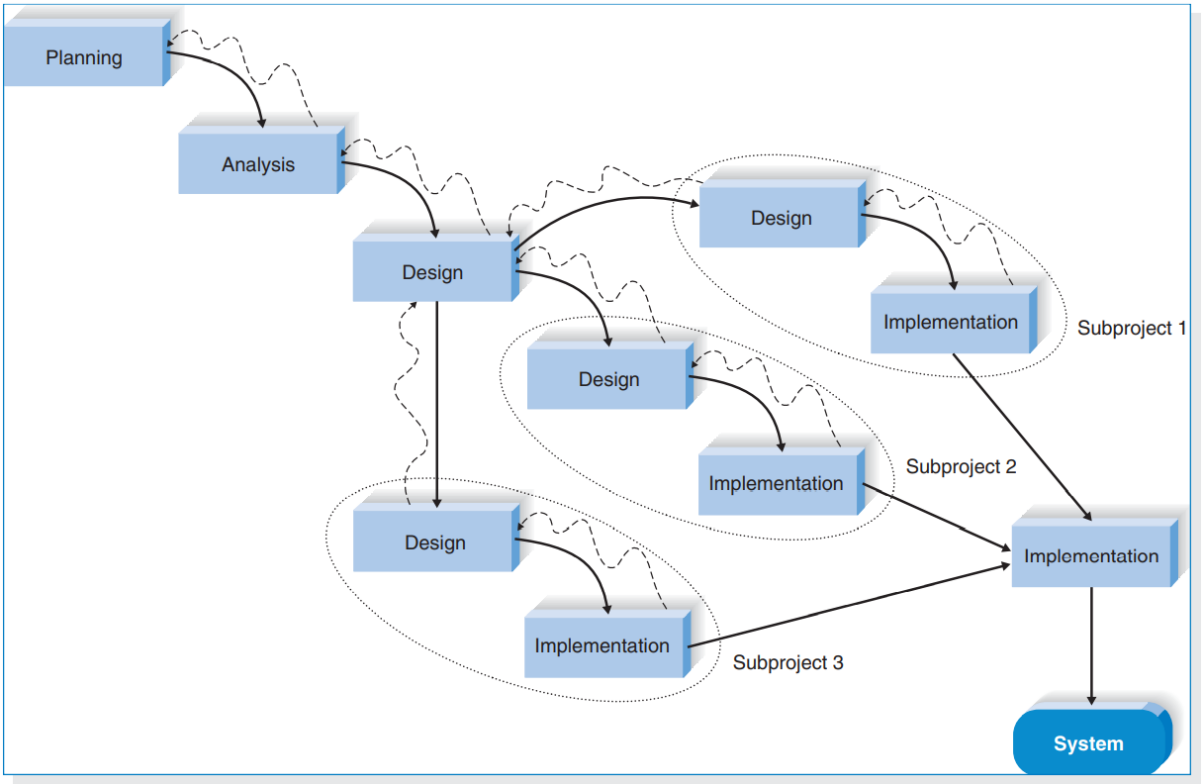
كما قد ينسى المستخدمون الغرض الأصلي من النظام، حيث انقضى وقت طويل بين الفكرة الأصلية والتنفيذ الفعلي. كما ان في بيئة الأعمال الديناميكية الحالية، قد يحتاج النظام الذي يلبي الشروط البيئية الحالية أثناء مرحلة التحليل إلى إعادة عمل لأشياء كثيرة لكي يتطابق مع البيئة عند تنفيذه. تتطلب إعادة العمل هذه العودة إلى المرحلة الأولى وإجراء التغييرات اللازمة خلال كل مرحلة من المراحل اللاحقة على التوالي. وقد تم بناء النظام المقترح في هذا البحث اعتمادا على هذه المنهجية.



الشكل (6) التطوير الشلالي

ب. التطوير على التوازي Parallel Development

تحاول المنهجيات التي تعتمد التطوير على التوازي أن تعالج موضوع الفجوة الزمنية الكبيرة بين لحظة طلب النظـام ولحظة تسليمه. فبدلاً من القيام بالتصميم كاملاً ثم الانتقال إلى التنجيز (كما هو الحال في التطوير الشلالي)، يوضع تصميم عـام للنظـام ككل، ثم يقسم المشروع إلى عدد من المشاريع الفرعية المستقلة التي يمكن تصميم كل منها وتنجيزه على التـوازي مـع المشاريع الفرعية الأخرى.



الشكل (7) التطوير على التوازي

2.2.4.3 التطوير السريع للتطبيقات Rapid Application Development

ظهرت المنهجيات المعتمدة على التطوير السريع للتطبيقات للتغلب على نقطتي الضعف المذكورتين آنفاً فـي منهجيـات التصميم البنيوي. لتحقيق هذا الهدف، تنسق المنهجيات المعتمدة على RAD بين مراحل دورة حياة تطوير النظام للحصول على أجزاء من النظام بسرعة وتضعها بين يدي المستخدم. إن حصول المستخدم على أجزاء من النظام في وقـت مبكـر يتيح له فهماً أفضل للنظام مما يجعله يقترح بعض التعديلات التي تجعل النظام أكثر تلبية لاحتياجاته.

تنصح معظم المنهجيات المعتمدة على التطوير السريع للتطبيقات أن يستخدم المحللون تقنيات مخصصة وأدوات حاسوبية خاصة لتسريع مراحل التحليل والتصميم والتنجيز، مثل أدوات هندسة البرمجيات بمعونـة الحاسـب CASE ولغات البرمجة المرئية كـ .NET MS Visual Basic. ومع ذلك تبقى هناك في المنهجيات المعتمدة على التطوير السريع للتطبيقات مشكلة خفية تكمن في إدارة توقعات الزبون. فمع زيادة

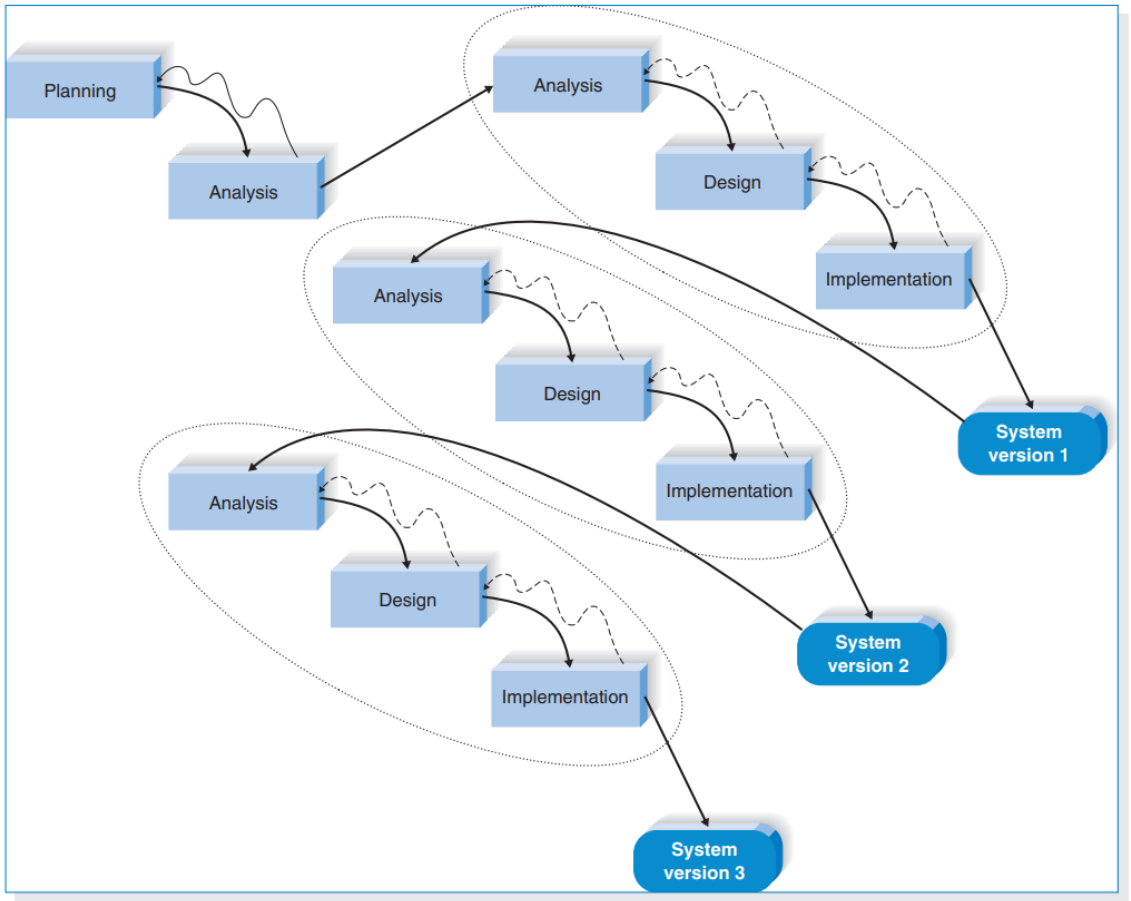
سرعة تطوير النظام يزداد فهم الزبون للتقانات المستخدمة وتزداد طلباته وتوقعاته من النظـام، ممـا يجعـل متطلبات النظام تتضخم بشكل كبير أثناء المشروع.

وتأخذ منهجية التطوير السريع للتطبيقات شكلين من أشكال التطوير وهما: التطوير التكراري والنمذجة الأولية.

أ. التطوير التكراري Iterative development

تعتمد هذه المنهجيات على تجزئة النظام الكلي إلى سلسلة من الإصدارات التي يجري تطويرها تتابعياً. ففي مرحلة التحليل يجري تحديد المفهوم الكلي للنظام، ثم يقوم فريق المشروع والمستخدمون والممول بتصنيف المتطلبات في سلسلة مـن الإصدارات المتتابعة. تشكل المتطلبات الأساسية والأكثر أهمية الإصدار الأول.

ننطلق من مجموعة المتطلبات هذه، وبعد تحليلها بدقة ننتقل إلى تصميها وتنجيزها فنحصل على الإصدار الأول من النظام ثم نكرر الأمر نفسه عدة مرات، بحيث أننا بعد انتهاء كل إصدار نبدأ العمل على الإصدار التالي الذي ينطلق من مجموعة المتطلبات السابقة مضافاً إليها الأفكار الجديدة التي يأتي بها المستخدم بعد تجربته مع الإصدار السابق.



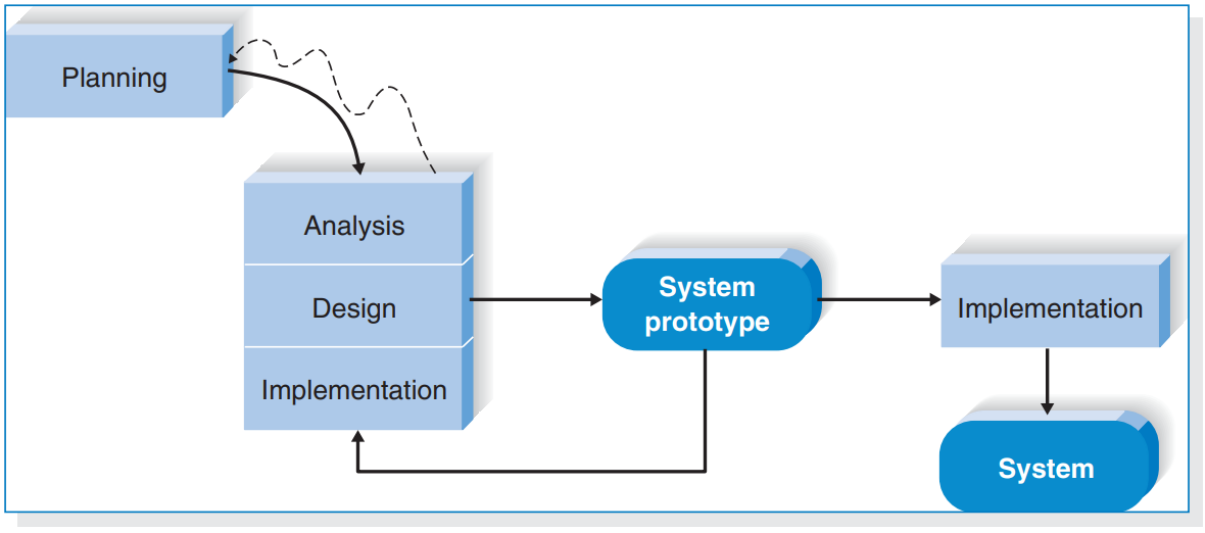
الشكل (8) التطوير التكراري

ب. النمذجة الأولية Prototyping

في هذه المنهجية، يجري العمل في مراحل التحليل والتصميم والتنجيز بشكل تسايري. تُؤدي هذه المراحل الثلاثة ضمن حلقة تكرارية إلى أن يتم إنجاز كامل النظام. في مثل هذه المنهجيات نبدأ العمل بإجراء تحليل وتصميم أساسيين، ثم نقوم مباشرة ببناء النموذج الأولي للنظام. إن النموذج الأولي برنامج "سريع وفج" يعتبر إصداراً أولاً مصغراً عن النظام، ويمتلك عدداً قليلاً من الوظائف والصفات المطلوبة من النظام، يشكل هذا الإصدار الجزء الأول الذي سيستخدمه الزبون، ويعرض عادة على المستخدم والممول لأخذ ملاحظاتهم وردود أفعالهم. بناء على هذه الملاحظات يجري العمل على نموذج فيه المزيد من الوظائف والصفات المطلوبة، وتكرر العملية السابقة إلى حين الحصول على النظام بكامل مواصفاته.

وتتميز المنهجيات المعتمدة على النموذج الأولي بأنها توفر للمستخدم نظاماً يمكنه التفاعل معه وإن لم يكن هذا النظام جاهزاً للاستخدام الفعلي .

أما مساوئ هذه المنهجيات فهي أنها تؤدي في غالب الأحيان وبسبب كثرة التعديلات التي تطرأ على النموذج الأولي إلى الحصول على تصميم سيء للنظام.

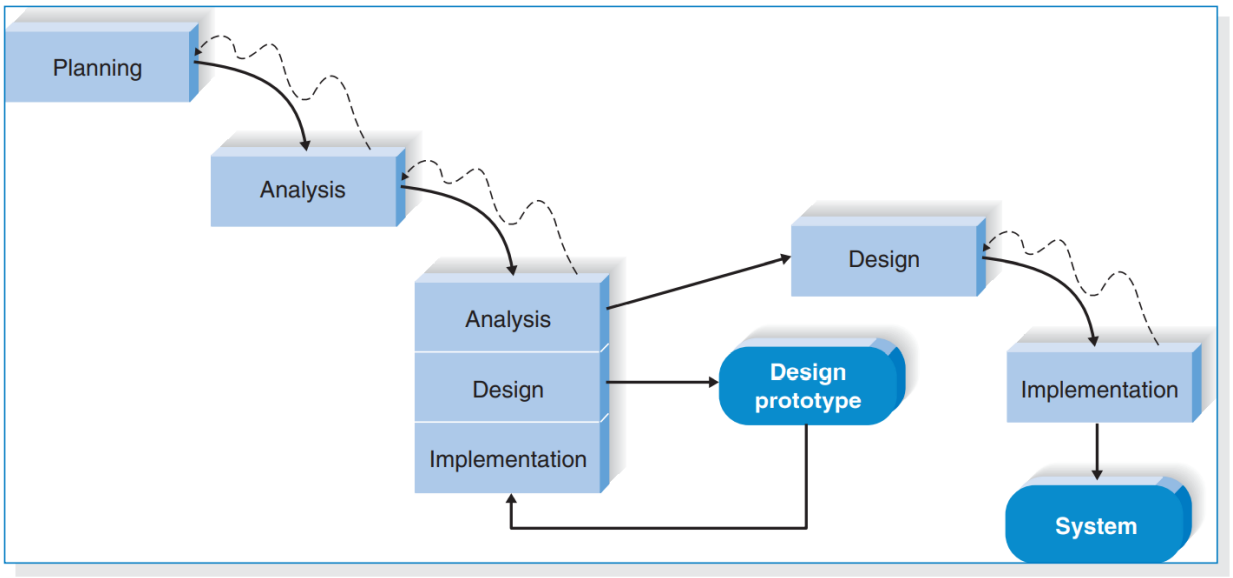


الشكل (9) النمذجة الأولية

ج. النمذجة الأولية مع رمي النموذج Throwaway Prototyping

تشبه هذه المنهجيات تلك التي أوردناها في الفقرة السابقة (المنهجيات المعتمدة على النموذج الأولي) في أنها تعتمد على صنع نماذج أولية، غير أنها تختلف عنها في أنه يجري صنع النماذج الأولية في موضع مختلف من دورة حياة تطوير النظم. تقوم هذه النماذج بدور مختلف عن مثيلاتها التي أوردناها في الفقرة السابقة كما يكون مظهرها مختلفاً كلياً. يقوم المحللون في هذه المنهجيات بإجراء تحليل عميق نسبياً يتم خلاله جمع المعلومات وتطوير أفكار حول مفهوم النظام.

قد تكون بعض خصائص النظام التي يطلبها المستخدم غير واضحة أو خيالية أو تمثل تحدياً تقنياً، وعليه يجري تجريب كل من هذه الطروحات عبر بناء نموذج أولي تصميمي Prototype Design.



الشكل (10) النمذجة الأولية مع رمي النموذج

3.2.4.3 التطوير الرشيق Agile Development

أخذت هذه المنهجيات بالظهور حديثاً. وهي تركز بشكل كبير على عملية البرمجة، وتمتلك عدداً قليلاً من القواعد والممارسات مما يجعل اتباعها سهلاً .تهدف هذه المنهجيات إلى الانسياب عبر دورة حياة التطوير البرمجي بإلغاء الكثير من الحمل الإضافي الذي ينتج عن عمليات النمذجة والتوثيق مما يؤدي إلى توفير الوقت الذي تستغرقه هذه العمليات. وبدلاً عن ذلك تركز المشاريع على تطوير بسيط وتكراري للتطبيقات. ومن الأمثلة على هذه المنهجيات نورد البرمجة الحدية Extreme Programming (XP).

أ. البرمجة الحدية Extreme Programming (XP)

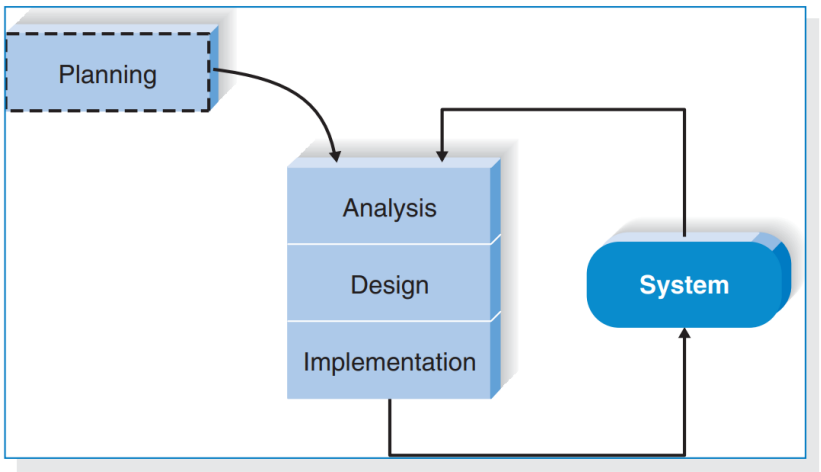
تقوم هذه المنهجية على القيم الأربعة التالية:

1. **التواصل:** يجب أن يقدم المطورون رد فعل سريع على طلبات الزبون وبشكل دائم.
2. **البساطة:** يجب أن يحافظ المطورون على المبدأ Keep it Simple & Stupid (KISS)
3. **رد الفعل:** يجب أن يقوم المطورون بتعديلات متزايدة Incremental حتى يكبر النظام تدريجياً، كما يجب أن يستوعبوا التعديلات ويعملوا على احتوائها لا مجرد أن يقبلوا بها.
4. **الشجاعة:** والذهنية المتوقدة التي يجب أن يتحلى بها المطورون.

كما تعتمد البرمجة الحدية المبادئ الأساسية التالية لصنع أنظمة ناجحة :

1. الاختبارات المستمرة.
2. الترميز البسيط.
3. التواصل الوثيق مع المستخدم لبناء النظم بناء سريعاً.

وبعد عملية تخطيط سريعة، تبدأ فرق العمل بإجراء التحليل والتصميم والتنجيز بشكل تكراري.



الشكل (11) البرمجة الحدية

3.4.3 اختيار منهجية التطوير المناسبة

ليس من السهل اختيار المنهجية المناسبة للتطوير، إذ لا توجد منهجية أجمع المطورون على أنها هي الأفضل كما أن لكل شركة تطوير معاييرها ومقاييسها النموذجية. سنسلط الضوء في هذه الفقرة على بعض النقاط التي يمكننا استخدامها كمعايير للمقارنة بين المنهجيات.

1. وضوح متطلبات المستخدم:

عندما يقدم لك المستخدم متطلبات غير واضحة عن النظام وعما يجب أن يفعله، يكون من الصعب فهم هذه المتطلبات بالحديث عنها أو بكتابة تقرير حولها. يحتاج المستخدم في هذه الحالة إلى التفاعل مع التقانة لفهم ما سيفعله النظام وكيف يمكن تطويع هذه التقانة لاحتياجات الزبون. في مثل هذه الحالات تكون المنهجيات المعتمدة على النمذجة الأولية وعلى النمذجة الأولية مع رمي النموذج هي الأكثر مناسبة لأنها تقدم نماذج أولية للمستخدمين تمكنهم من التفاعل معها في مرحلة مبكرة من دورة الحياة.

2. التآلف مع التكنولوجيا:

إذا كان النظام مصمماً دون أن يكون فريق المشروع متآلفاً مع التكنولوجيا الأساسية فيه، تزداد المخاطرة لأن الأدوات قد لا تكون قادرة على فعل المطلوب منها. في هذه الحالة يكون استخدام المنهجيات المعتمدة على النمذجة الأولية مع رمي النموذج هو الخيار الأفضل، في حين لا يكون استخدام النمذجة الأولية مناسباً.

3. تعقيد النظام:

تحتاج النظم المعقدة إلى تحليل وتصميم دقيقان. يمكن في هذه النظم استخدام النمذجة الأولية مع رمي النموذج، أو استخدام المنهجيات المعتمدة على التصميم البنيوي. أما استخدام التطوير على مراحل، فقد بينت التجربة أن فرق العمل الذي اعتمدته كانت تولي تحليل النظام المعقد اهتماماً أقل مما لو استخدمت منهجيات أخرى.

4. موثوقية النظام :

تعتبر موثوقية النظام عاملاً هاماً في تطوير النظم. وتشكل المنهجيات المعتمدة على النمذجة الأولية مع رمي النموذج الخيار الأفضل عندما تكون الموثوقية ذات أولوية عالية. أما استخدام النمذجة الأولية فلا ينصح به هنا لأنه تنقصه الدقة والتأني في مرحلتي التحليل والتصميم.

5. الخطط الزمنية قصيرة:

تناسب هذه المشاريع المنهجيات المعتمدة على التطوير السريع للتطبيقات RAD. كما أن النمذجة الأولية والتطوير على مراحل يشكلان خيارين ممتازين لمثل هذه المشاريع. أما التطوير الشلالي فهو الخيار الأسوأ ويجب الابتعاد عنه.

6. متابعة الخطة الزمنية :

لا توفر جميع المنهجيات القدرة على متابعة الخطط الزمنية والتحقق من مدى التقيد بها بدرجة واحدة. ونظـراً إلـى أن التصميم البنيوي يترك التصميم والتنجيز للمراحل الأخيرة، فإنه يخشى من عدم التمكن من المتابعة. تنقل منهجيات RAD الكثير من قرارات التصميم إلى البدايات مما يسمح بالتعرف إلى مواطن المخاطرة العالية والتصدي لها مبكراً.

يلخص الجدول التالي هذه المعايير للمفاضلة بين المنهجيات:

Usefulness in Developing Systems	Waterfall	Parallel	V-Model	Iterative	System Prototyping	Throwaway Prototyping	Agile Development
with unclear user requirements	Poor	Poor	Poor	Good	Excellent	Excellent	Excellent
with unfamiliar technology	Poor	Poor	Poor	Good	Poor	Excellent	Poor
that are complex	Good	Good	Good	Good	Poor	Excellent	Poor
that are reliable	Good	Good	Excellent	Good	Poor	Excellent	Good
with short time schedule	Poor	Good	Poor	Excellent	Excellent	Good	Excellent
with schedule visibility	Poor	Poor	Poor	Excellent	Excellent	Good	Good

الجدول (2) مقارنة منهجيات التطوير

4

الفصل الرابع

الإطار العملي

- تمهيد
- تحليل النظام
- تصميم النظام
- تنجيز النظام

SWIFT

Timetable Management System

1.4 تمهيد

في هذا الفصل بيان لجميع المراحل التي مر بها النظام من الصفر حتى البناء الكامل. حيث قسم إلى ثلاثة أقسام رئيسية هي تحليل النظام، تصميم النظام وتنجيز النظام.

يتم في مرحلة تحليل النظام دراسة كل ما من شأنه أن يساعد الباحث في تحديد متطلبات النظام، مثل التعرف على المعهد العالي لإدارة الأعمال ونظام الجدولة الحالي فيه وصياغة مسألة الجدولة الزمنية ومن ثم تحديد متطلبات النظام وتصميم قاعدة المعطيات بناء على تحليل حالات الاستخدام وتحديد ودراسة العلاقات بين الكيانات المختلفة.

في مرحلة التصميم يتم تحديد كيف سيعمل النظام، أي ما هي الالية والخوارزميات التي سيعتمد عليها، لذلك في هذه المرحلة يبدأ الفصل بعرض فكرة الحل المقترح والذي يتكون من جزئين، أولهما نموذج البرمجة الطبيعية والذي سينتج عن حله تعيين كل محاضرة لفترة زمنية محددة وثانيهما خوارزمية تخصيص القاعات الدراسية.

مرحلة التنجيز هي المرحلة الأخيرة ويمكن تسميتها بمرحلة البرمجة، في هذه المرحلة يتم عرض الأدوات البرمجية المستخدمة ومناقشة كل منها مع عرض واجهات الاستخدام التي يتم من خلالها التعامل مع النظام بالإضافة لمجموعة واسعة من الاختبارات التي تم إجراؤها على بيانات واقعية بغرض بيان كفاءة عمل النظام.

2.4 تحليل النظام

يبدأ هذا القسم بالتعريف بجهة التطبيق وهي المعهد العالي لإدارة الأعمال HIBA، وتسليط الضوء على نظام جدولة المقررات الدراسية الحالي في المعهد، الأمر الذي سيمهد لعملية تحديد المتطلبات وفهم المعالم الأساسية للمشكلة. ينتقل القسم بعد ذلك إلى دراسة المشكلة بشيء من التفصيل من خلال بيان المكونات الأساسية لمسألة الجدولة الزمنية وطبيعة الارتباطات بين هذه المكونات ومن ثم الحصول على صياغة موجزة وواضحة للمسألة. بناء على ما سبق يتم استنتاج متطلبات النظام الأساسية واقتراح حل مناسب يحقق جميع هذه المتطلبات. ثم ينتقل إلى تحليل حالات الاستخدام الأساسية في النظام من خلال تحديد الفاعلين الأساسيين وكيفية تفاعلهم مع النظام المقترح، وإعداد مخططات حالات الاستخدام. بعد ذلك ينتقل إلى مرحلة تصميم قاعدة المعطيات وسيتم ذلك من خلال تحديد الكيانات الأساسية في النظام وتعريف العلاقات فيما بينها وإنشاء مخطط العلاقات بين الكيانات ERD، ومن ثم تصميم جداول قاعدة المعطيات لتكون جاهزة في مرحلة البرمجة.

1.2.4 مقدمة عن المعهد العالي لإدارة الأعمال

أحدث المعهد العالي لإدارة الأعمال بتاريخ 2001/6/23 وهو مؤسسة تعليمية متخصصة في علوم الإدارة.[28] يهدف المعهد إلى تنمية الموارد البشرية وتأهيل الطلاب من مختلف الاختصاصات في مجالات علوم إدارة الأعمال على مختلف المستويات وفي جميع القطاعات العام والخاص والمشترك.

رؤية المعهد Vision: يتطلع المعهد إلى التميز في علوم الإدارة والأبحاث على المستوى الإقليمي والدولي، والارتباط باتفاقيات علمية وثقافية مع مؤسسات أجنبية ومحلية، وإعداد أفضل الكوادر المتخصصة في الإدارة إعداداً علمياً وعملياً، وتنمية رأس المال الفكري المبدع والمبادر عن طريق الحوار والتكامل الثقافي لتلبية حاجات قطاع الأعمال الحالية والمستقبلية.

رسالة المعهد Mission: تخريج كوادر إدارية متخصصة على مستويات الإجازة الجامعية والدراسات العليا لرفد قطاع الأعمال بما يتناسب مع متطلبات سوق العمل وتوسيع نطاق نشر المعرفة الإدارية في القطاعات الاقتصادية المختلفة، والإسهام في خدمة المجتمع من خلال تبني مبدأ المسؤولية الاجتماعية للتنمية البشرية واكتساب القيم الاجتماعية والوطنية والإنسانية وتكريس أخلاقيات الأعمال.

يتكون المعهد من الأقسام العلمية التالية:

- قسم إدارة العمليات والمعلومات
- قسم التسويق
- قسم الإدارة المالية والمصرفية
- قسم إدارة الموارد البشرية
- قسم العلوم الأساسية والرديفة

يمنح المعهد الشهادات التالية:

- الشهادة التحضيرية في الإدارة
- الإجازة في علوم الإدارة
- الماجستير في علوم الإدارة
- الدكتوراه في علوم الإدارة
- دبلوم التأهيل والتخصص في إدارة الأعمال
- الماجستير في إدارة الأعمال

1.1.2.4 تحليل نظام الجدولة الحالي في المعهد

تختلف أنظمة المقررات الدراسية بين المؤسسات التعليمية بشكل كبير، لذلك تم إجراء بعض المقابلات مع الإداريين المعنيين في المعهد من أجل جمع المعلومات المطلوبة والحصول على فهم كامل حول المشكلة. فيما يلي ملخص يشرح خصائص هذا النظام في المعهد العالي لإدارة الأعمال بشيء من التفصيل.

بداية من أجل فهم سبب التنوع في خصائص المحاضرات يمكننا تمييز مرحلتين دراسيتين أساسيتين، المرحلة الأولى هي السنوات الدراسية الأولى والثانية والثالثة (الحلقة الأولى)، أما المرحلة الثانية فهي السنوات الدراسية الرابعة والخامسة (الحلقة الثانية). محل اهتمام هذا البحث هو حل مشكلة الجدولة للحلقتين الأولى والثانية نظراً لأنها الأكثر تعقيدا وتشمل الحالات الأخرى مثل جدولة محاضرات الدراسات العليا. لذلك سنقتصر على شرح نظام الجدولة في المعهد للحلقتين الأولى والثانية.

إن الجدول الأسبوعي للمحاضرات يتكون من مجموعة من الفترات الزمنية والتي هي عبارة عن تقاطع أيام الأسبوع مع الفترات اليومية. يفترض النظام الحالي وكذلك النظام المقترح في هذا البحث أن فترات المحاضرات متساوية. يبين الشكل التالي مثال عن جدول محاضرات للسنة الرابعة اختصاص إدارة العمليات والمعلومات.

	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس
08:30 - 10:30	إدارة الأعمال الدولية د.محمد الخضر + أ.ربى الصعيدي قاعة 8		إدارة الإنتاج د.رعد الصرن قاعة 1	منهجيات تحليل وتصميم النظم د. كادان جمعة قاعة 1	نظرية القرارات د.طلال عبود قاعة 9
10:30 - 12:30	إدارة الأعمال الدولية د.محمد الخضر + أ.ربى الصعيدي قاعة 8		إدارة الإنتاج د.رعد الصرن قاعة 1	منهجيات تحليل وتصميم النظم د. كادان جمعة قاعة 1	نظرية القرارات د.طلال عبود قاعة 9
12:30 - 02:30	إدارة الجودة د. رعد الصرن + أ.بزار اشريفة قاعة 12		نماذج متقدمة في بحوث العمليات د. وائل خنسة قاعة 1	نماذج متقدمة في بحوث العمليات د. وائل خنسة قاعة 1	
02:30 - 04:30	إدارة الجودة د. رعد الصرن + أ.بزار اشريفة قاعة 12				

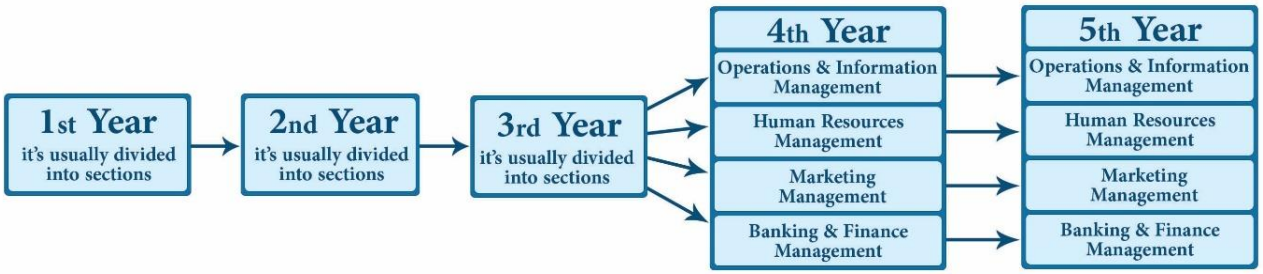
الشكل (12) جدول المحاضرات الأسبوعي للسنة الرابعة

نلاحظ من الشكل السابق أن جدول المحاضرات الأسبوعي في المعهد هو عبارة عن تقاطع مجموعة من الفترات مع أيام الأسبوع، بحيث أن جميع الفترات متساوية المدة. ينتج عن ذلك مجموعة من الفسح الزمنية *timeslots* التي يجب توزيع المحاضرات عليها. كما نلاحظ أنه قد يشترك في إعطاء المقرر أكثر من مدرس وفي هذه الحالة يخصص لكل مدرس عدد محدد من المحاضرات الأسبوعية.

بالنسبة للاختلاف في طبيعة المحاضرات بين الحلقتين الأولى والثانية فيمكن تلخيص ذلك فيما يلي:

الحلقة الأولى: يتلقى طلاب كل سنة دراسية في هذه المرحلة نفس المقررات الدراسية، إلا أن العدد الكبير من الطلاب يفرض على المعهد تقسيم كل سنة دراسية إلى عدة فئات حتى يكون العدد معقولاً لتلقي المحاضرات، وبالتالي فإن المحاضرة سوف تتكرر بعدد فئات السنة الدراسية.

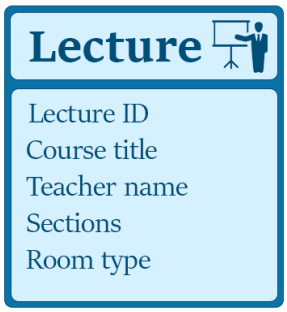
الحلقة الثانية: في هذه المرحلة يختار الطالب واحداً من التخصصات الأكاديمية المتاحة في المعهد (إدارة التسويق، إدارة الموارد البشرية، إدارة مالية ومصرفية، إدارة العمليات والمعلومات) وبالتالي ينتج لدينا أربع فئات من الطلاب من كل سنة دراسية. ويكون لدى الطلاب صنفين من المقررات الدراسية، الصنف الأول هو مجموعة مقررات مشتركة يتلقاها جميع طلاب السنة الدراسية (من كل الاختصاصات) وهنا نكون أمام نفس الحالة في المرحلة الأولى إلا أننا غالباً لا نواجه مشكلة في العدد وبالتالي لا تظهر الحاجة إلى تكرار المحاضرة أكثر من مرة وإنما يتلقى جميع الطلاب المحاضرة في آن معاً. أما الصنف الثاني فهو مقررات الاختصاص، حيث يتلقى طلاب كل اختصاص من الاختصاصات الأربعة مجموعة من المقررات التي لا يتلقاها سواه.



الشكل (13) المراحل الدراسية في المعهد

وبالتالي قد أصبح هناك ثلاثة أنواع من المحاضرات:

1. **محاضرات مشتركة لكل فئة على حدة**
   وبالتالي تكرر المحاضرة لكل فئة (كما هو الحال عندما يكون عدد الطلاب كبير)
2. **محاضرات مشتركة لجميع فئات السنة الدراسية مع بعضها البعض**
   عندما يكون عدد الطلاب غير كبير ولكن التقسيم إلى فئات نتج بسبب التخصصات المتعددة
3. **محاضرة اختصاص**



الشكل (14) صيغة موحدة للمحاضرة

سيتم تحويل جميع المحاضرات إلى صيغة موحدة لتسهيل عملية التعامل معها عند الوصول إلى مرحلة النمذجة الرياضية. بحيث يكون لكل محاضرة عنوان (هو نفسه اسم المقرر) ومدرس محدد ومجموعة محددة من الفئات تمثل حضور المحاضرة (قد تتكون هذه المجموعة من فئة واحدة أو أكثر) ونوع محدد من القاعات يناسبها (صالة حاسوب، قاعة كبيرة، قاعة صغيرة، الخ...). وكل محاضرة يجب أن تحدث مرة ومرة واحدة فقط كل أسبوع.

إن تعريف المحاضرات بصيغة موحدة يسهل عملية تحديد مجموعة القيود التي تخضع لها عملية الجدولة، مثل عدم تعارض المحاضرات مع بعضها البعض سواء من حيث المدرس المسؤول عن إلقاء المحاضرة أو فئات الطلاب الذين يمثلون مجموعة الحضور.

بالإضافة لما سبق، ومن خلال المقابلات مع الإداريين المعنيين والبحث عن المتطلبات الوظيفية التي يرغبون بتوفرها، تبين إلى أن هناك ميل نحو ضغط الجدول الزمني بالنسبة للمدرسين الخارجيين، حيث يفضل المدرسون الخارجيون ضغط جدول محاضراتهم قدر الإمكان بهدف تقليل عدد أيام قدومهم إلى المعهد بالإضافة الى تقليل الفراغات بين محاضراتهم. وبالمثل بالنسبة للطلاب، حيث يواجه معظم الطلاب صعوبة في القدوم إلى المعهد بسبب سوء حال المواصلات الناتج عن المشاكل الاقتصادية والسياسية التي تواجهها البلاد، وبالتالي يرغبون بتقليل عدد أيام حضورهم. إن كل ما سبق سوف يؤخذ بعين الاعتبار عند وضع متطلبات النظام.

2.2.4 صياغة المسألة وتحديد المتطلبات

في هذا القسم سنقوم بتناول مسألة الجدولة الزمنية للمقررات الدراسية بشيء من التفصيل من خلال مناقشة مكوناتها الأساسية ومن ثم الحصول على صياغة واضحة لهذه المسألة تساعدنا في تحديد متطلبات النظام المقترح.

باختصار تتكون مسألة الجدولة الزمنية في المعهد العالي لإدارة الأعمال من مجموعة من **المقررات الدراسية** C. ولكل مقرر دراسي مدرس واحد على الأقل وبالتالي لدينا مجموعة من **المدرسين** T. يخصص كل مقرر دراسي لمجموعة واحدة على الأقل من الطلاب (سنصطلح أسم "فئة" على مجموعة الطلاب الذين يشتركون بنفس المقررات الدرسية) وبالتالي لدينا مجموعة من **الفئات** S. ينتج عن كل مقرر درسي مجموعة من **المحاضرات الأسبوعية (الحصص الدراسية)** L يجب أن نضع كل منها في فترة زمنية محددة، أي لدينا مجموعة من **الفترات الاسبوعية** P. وأخيراً يجب أن يخصص لكل محاضرة من المحاضرات قاعة تدريسية، أي هناك مجموعة من **القاعات التدريسية** R.

وبالتالي تكون المكونات الأساسية للمسألة كما يلي:

مجموعة من المقررات الدرسية $C = \{c_1, c_2, \dots\dots\dots, c_n\}$

مجموعة المحاضرات الأسبوعية $L = \{l_{1,1}, l_{1,2}, \dots\dots\dots, l_{n,m}\}$

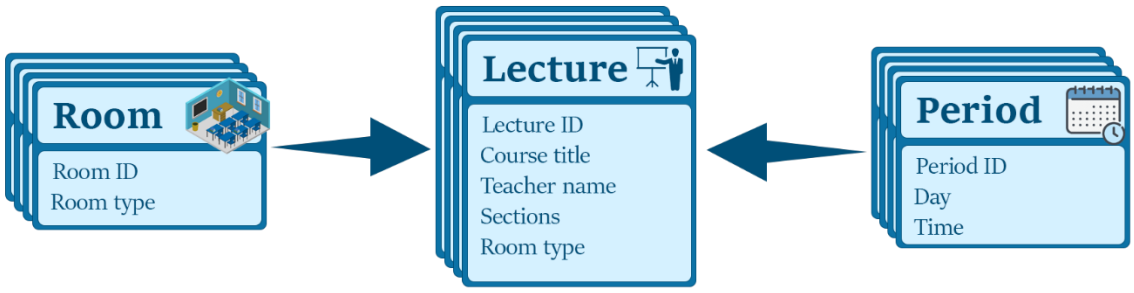
مجموعة من المدرسين $T = \{t_1, t_2, \dots\dots\dots, t_i\}$

مجموعة من الفئات $S = \{s_1, s_2, \dots\dots\dots, s_j\}$

مجموعة من القاعات التدريسية $R = \{r_1, r_2, \dots\dots\dots, r_k\}$

الفترات الزمنية $P = \{p_1, p_2, \dots\dots\dots, p_l\}$

وهكذا تكون مسألة الجدولة الزمنية بشكل عام هي عبارة عن تخصيص مجموعة المحاضرات L (وما يرتبط بها من مدرسين وفئات ونوعية محددة من القاعات) إلى فترات زمنية محددة وقاعات تدريسية محددة.



الشكل (15) مسألة الجدولة كمسألة تعيين

إلا أن هناك ضرورة لأخذ المتطلبات التي تم جمعها من الإدارة بعين الاعتبار لكي يكون الحل قابل للتطبيق ومرضي بالنسبة لجميع أصحاب المصلحة. فيما يلي مجموعة المتطلبات:

- تخصيص فترة زمنية محدد لكل محاضرة.
- تخصيص قاعة تدريسية محددة لكل محاضرة مع الأخذ بعين الاعتبار نوع القاعة المطلوب.
- إنشاء جداول أسبوعية للمحاضرات (محققة لكل الشروط)
- عرض الجداول الأسبوعية
- إمكانية إضافة أو حذف أو تعديل المقررات \ المدرسين \ السنوات الدراسية \ الفئات \ أنواع القاعات الدرسية \ القاعات الدرسية.
- إمكانية إضافة مقررات يشترك في تدريسها أكثر من مدرس.
- إمكانية إضافة محاضرات لفئة واحدة.
- إمكانية إضافة محاضرات مشتركة لمجموعة من الفئات مع بعضهم البعض.
- إمكانية تحديد أيام الدوام وأيام العطلة.
- إمكانية تحديد عدد الفترات اليومية ووقت بداية المحاضرات.
- إمكانية تحديد مدة المحاضرة \ الحصة الدرسية.

- إمكانية اختيار مجموعة جزئية من مجموعة المحاضرات لجدولتها.
- إمكانية تحديد الأيام \ الفترات المتاحة لكل مدرس.
- إمكانية إضافة قيود ضغط الجدول الزمني لبعض المدرسين دون غيرهم.
- أمكانية تحديد أولوية لبعض المدرسين من ناحية ضغط جدولهم الزمني.
- إمكانية إيقاف \ تفعيل خاصية ضغط الجدول الزمني للمدرسين \ للفئات.
- إمكانية وضع حد أقصى لعدد المحاضرات اليومية لمدرس ما.
- إمكانية وضع حد لعدد القاعات التدريسية المتاحة من نوع ما في فترة زمنية محددة.
- إمكانية استخدام الحلول القديمة.
- إمكانية التعديل على الحلول (تغيير الفترة، تغيير نوع القاعة التدريسية، تغيير القاعة التدريسية)
- إمكانية التأكد من عدم وجود أية تعارض في الحل بعد التعديل عليه.
- إمكانية حفظ الجداول الزمنية للمحاضرات في ملفات Excel.

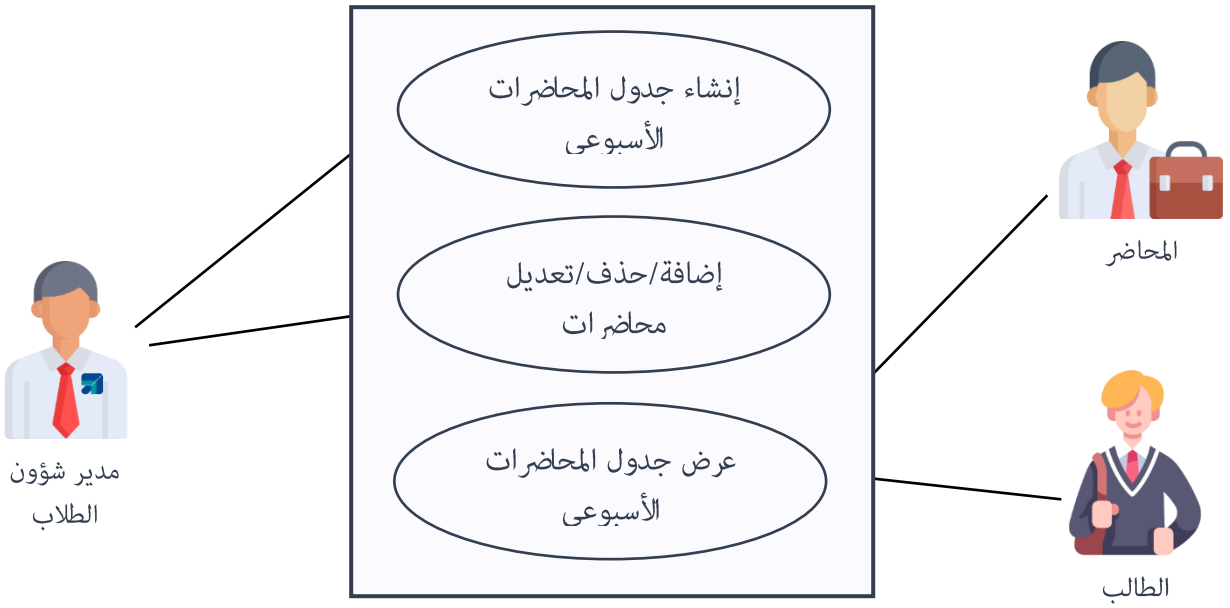
3.2.4 تحليل حالات الاستخدام

بعد أن تم التوصل للمتطلبات التي يرغب بتوافرها في النظام، سيتم في هذا القسم تحليل حالات الاستخدام من خلال تحديد الفاعلين الرئيسيين ومن ثم دراسة حالات الاستخدام من خلال توصيفها وعرض الخطوات التي تمر فيها كل حالة.

تم البدء بتحديد الفاعلين الرئيسيين في النظام، والذين يمثلون جميع من يتعامل مع النظام بشكل مباشر أو غير مباشر. تم تحديد الأطراف الفاعلة في النظام كما يلي:

1. مدير شؤون الطلاب
2. الطلاب
3. المدرسين

بعد تحديد الفاعلين الرئيسيين تم تحديد حالات الاستخدام الأساسية في النظام Use Cases. ومن أجل فهم النظام بشكل عام يتم عرض المخطط العام لحالات الاستخدام والذي يحتوي أهم الخدمات التي سيقدمها النظام للفاعلين.



الشكل (16) المخطط العام لحالات الاستخدام

فيما يلي توصيف كل حالة من الحالات الثلاث المذكورة حيث يتضمن توصيف الحالة أسم الحالة، رقم الحالة، درجة الأهمية، الفاعل، المحرض، نوع المحرض، وصف الحالة، الخطوات الرئيسية، وحالة النهاية الناجحة أو الفاشلة.

اسم حالة الاستخدام:	إضافة/حذف/تعديل محاضرات	
رقم الحالة:	1	
درجة الأهمية:	متوسطة.	
الفاعل:	مدير شؤون الطلاب.	
المحرض:	يطلب مدير شؤون الطلاب إضافة/حذف/تعديل محاضرات.	
نوع المحرض:	خارجي.	
وصف الحالة:	توصف هذه الحالة كيف يمكن لمدير شؤون الطلاب إضافة/حذف/تعديل محاضرة من قاعدة المعطيات.	

الخطوات الرئيسية للحالة:

1. يقوم مدير شؤون الطلاب باختيار اسم المقرر.
2. يقوم مدير شؤون الطلاب باختيار السنة الدراسية.
3. يقوم مدير شؤون الطلاب باختيار عدد المسارات المطلوب.
4. يقوم مدير شؤون الطلاب باختيار اسم المدرس لكل مسار.
5. يقوم مدير شؤون الطلاب بتحديد نوع القاعة المناسب لكل مسار.
6. يقوم مدير شؤون الطلاب باختيار مجموعة الحضور (الفئات) لكل مسار.
7. يقوم مدير شؤون الطلاب بتحديد عدد المحاضرات الأسبوعية لكل مسار.
8. يقوم مدير شؤون الطلاب بطلب إضافة المحاضرة.

حالة النهاية الناجحة:	يقوم النظام بإضافة المحاضرة إلى قاعدة المعطيات.
حالة النهاية الفاشلة:	يعرض النظام رسالة خطأ بسبب نقص بعض البيانات المطلوبة.

اسم حالة الاستخدام:	جدولة مجموعة من المحاضرات.	
رقم الحالة:	2	
درجة الأهمية:	عالية.	
الفاعل:	مدير شؤون الطلاب.	
المحرض:	يطلب مدير شؤون الطلاب جدولة مجموعة من المحاضرات.	
نوع المحرض:	خارجي.	
وصف الحالة:	توصف هذه الحالة كيف يمكن لمدير شؤون الطلاب جدولة مجموعة من المحاضرات.	

الخطوات الرئيسية للحالة:

بعد أن قام مدير شؤون الطلاب بإدخال جميع المعطيات المطلوبة والواجب توافرها لعمل النظام، أصبح النظام جاهزاً للبدء بعملية الجدولة.

1. تحديد أيام الدوام، وقت بداية أول محاضرة، مدة المحاضرة وعدد المحاضرات اليومية.
2. تحديد مجموعة المقررات المراد جدولة محاضراتها.
3. إدخال الأوقات المتاحة والغير متاحة لكل مدرس.
4. تحديد تكلفة يوم العمل وتكلفة وقت التعطل لكل مدرس.
5. اختيار القيود المرنة المراد تفعيلها.
6. تحديد تكلفة يوم الحضور وتكلفة وقت التعطل للفئات.
7. تحديد الحد الأقصى لعدد المحاضرات التي يمكن للمدرس أن يلقيها في اليوم. (اختياري)
8. تحديد عدد القاعات من نوع محدد والمتاحة في فترة زمنية معينة. (اختياري)
9. طلب بناء النموذج.
10. يقوم النظام بسحب ومعالجة المعطيات المطلوبة من قاعدة المعطيات وبناء النموذج.
11. طلب تشغيل محرك الحل (Solver).

يحفظ النظام الحل.

حالة النهاية الناجحة:	يقوم النظام بحفظ حل مقبول. (مقبول أمثلي)
حالة النهاية الفاشلة:	إدخال معطيات متعارضة للنظام أو غير قابلة للحل مما يؤدي لعرض النظام لرسالة غلط.

اسم حالة الاستخدام:	عرض جدول المحاضرات الأسبوعي.	
رقم الحالة:	3	
درجة الأهمية:	عالية.	
الفاعل:	مدير شؤون الطلاب.	
المحرض:	يطلب مدير شؤون الطلاب بعرض جدول المحاضرات الأسبوعي.	
نوع المحرض:	خارجي.	
وصف الحالة:	توصف هذه الحالة كيف يمكن لمدير شؤون الطلاب عرض حل قديم أو جديد. (جدول المحاضرات الأسبوعي)	

الخطوات الرئيسية للحالة:

1. يقوم مدير شؤون الطلاب باختيار الحل المراد عرضه.
2. يقوم النظام بتشغيل خوارزمية تعيين القاعات.
3. يقوم النظام بعرض مجموعة المحاضرات وقاعاتها.
4. في حال عدم كفاية القاعات:
   يقوم مدير شؤون الطلاب باختيار قاعات من غير نوع للمحاضرات التي لم يتبق لها قاعات.
   أو يقوم مدير شؤون الطلاب بإضافة قاعات جديدة إلى قاعدة المعطيات.
5. يقوم مدير شؤون الطلاب بتغيير قاعات مجموعة من المحاضرات. (اختياري)
6. يقوم النظام بالتأكد من عدم تعارض المحاضرات من ناحية القاعات.
7. يقوم النظام بعرض الحل على شكل جداول أسبوعية مصنفة حسب الفئات، السنوات والمدرسين.
8. يقوم مدير شؤون الطلاب بنقل محاضرات من فترة لأخرى. (اختياري)
   يقوم النظام بالتأكد من صحة الحل وعدم وجود تناقض من ناحية القاعات والفئات والمدرسين.
9. يقوم مدير شؤون الطلاب بطلب تصدير الحل (جدول المحاضرات الأسبوعي) بصيغة ملفات Excel. (اختياري)
   يقوم النظام بحفظ جدول المحاضرات الأسبوعي على شكل ملفات Excel.

حالة النهاية الناجحة:	يقوم النظام بعرض الجداول الأسبوعية للمحاضرات.
حالة النهاية الفاشلة:	في حال وجود تعارض في الحل يقوم النظام بعرض رسالة خطأ توضح مكان الخطأ.

4.2.4 تصميم قاعدة المعطيات

يعد التصميم الجيد لقاعدة المعطيات أمراً مهماً لضمان اتساق البيانات وعدم تكرارها، بالإضافة لسرعة تنفيذ الاستعلامات وبالتالي الحصول على نظام عالي الأداء. كما يسهل التصميم الجيد لقاعدة المعطيات عملية الوصول للبيانات واسترجاعها وتجنب الأخطاء أثناء مرحلة التطوير وبالتالي توفير الوقت.

في هذه المرحلة قمنا بدايةً بتحديد البيانات التي سيحتاجها النظام وسيتعامل معها، وذلك من خلال تحديد وتحليل الكيانات الأساسية وخصائص كل منها، ومن ثم تعريف العلاقات ما بين الكيانات وخصائص كل من هذه العلاقات، كما تم إعداد مخطط العلاقات بين الكيانات ERD.

1.4.2.4 الكائنات ومخطط العلاقات بين الكيانات

من خلال تحديد متطلبات النظام ومراعاة حالات الاستخدام التي تم عرضها وشرحها في القسم السابق، تم التوصل للكيانات الأساسية في النظام وتحديد خصائص كل منها. يعرض الجدول التالي أسم كل كيان وخصائص كل من هذه الكيانات، حيث تم توصيف المفتاح الرئيسي للكيان بوضع خط تحت أسم الصفة التي تشكل هذا المفتاح.

جدول الكيانات Entities		
أسم الكيان	**خصائص الكيان**	**نوع الكيان**
المقرر الدراسي	رقم المقرر، عنوان المقرر	كيان مستقل
المحاضرة	رقم المحاضرة	كيان تابع
المدرس	رقم المدرس، أسم المدرس، تكلفة يوم عمل المدرس، تكلفة فترة تعطل المدرس	كيان مستقل
نوع القاعة	رقم النوع، توصيف النوع	كيان مستقل
القاعة	رقم القاعة، أسم القاعة	كيان تابع
السنة الدراسية	رقم السنة الدراسية، أسم السنة الدراسية	كيان مستقل
الفئة	رقم الفئة، أسم الفئة، تكلفة يوم حضور الفئة، تكلفة فترة تعطل الفئة	كيان تابع
الفترة	رقم الفترة، وقت بداية الفترة، وقت نهاية الفترة	كيان مستقل
اليوم	رقم اليوم، أسم اليوم	كيان مستقل

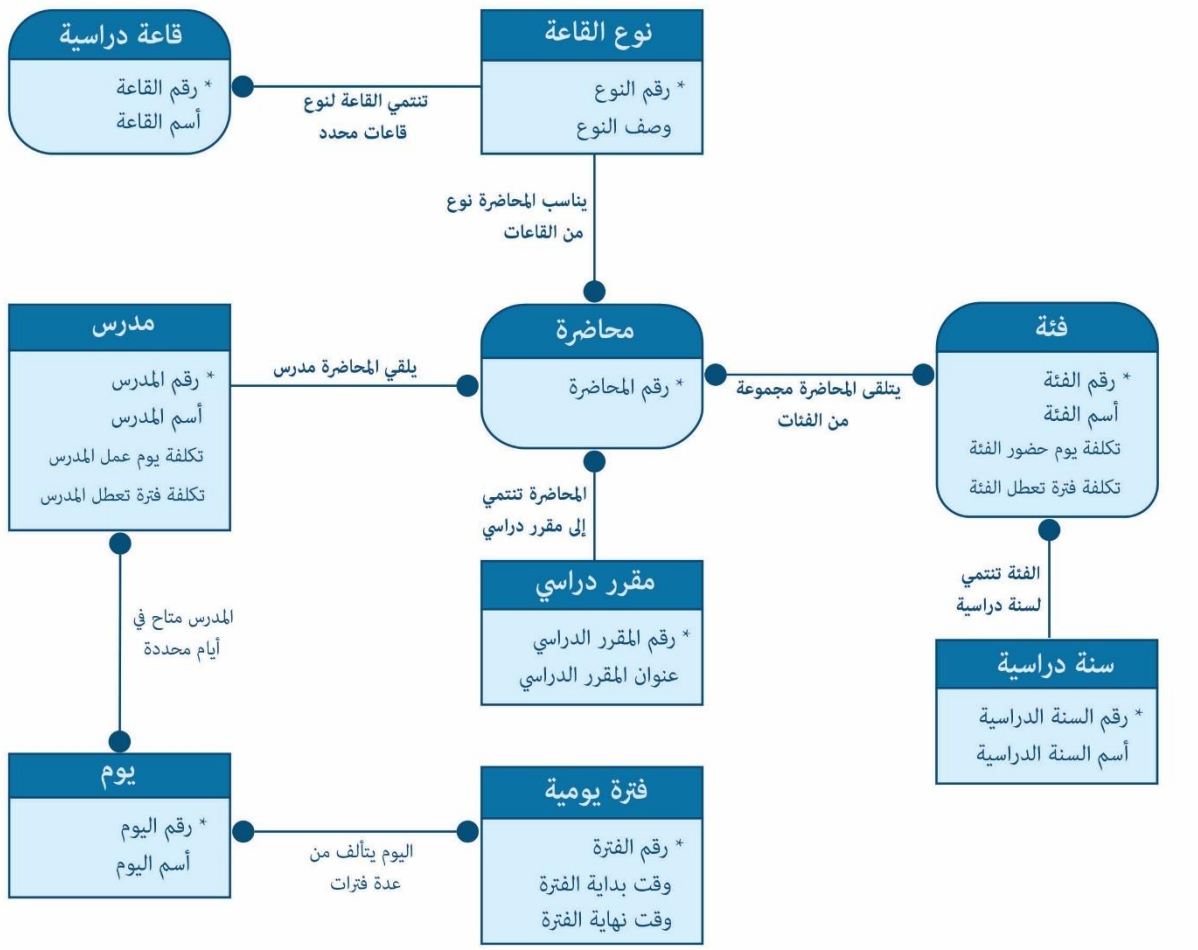
الجدول (3) الكيانات

بعد تحديد الكيانات الأساسية ودراسة خصائص كل منها، قمنا بدراسة العلاقة بين هذه الكيانات الأمر الذي يوضح بشكل أكبر ما هو المقصود بكل كيان. يوضح الجدول التالي العلاقات بين الكيانات ونوع كل من هذه العلاقات وشرح مختصر عن كل منها.

جدول العلاقات Relations				
الكيان 1	**الكيان 2**	**اسم العلاقة**	**نوع العلاقة**	**شرح مختصر**
المحاضرة	**المقرر الدراسي**	المحاضرة تنتمي إلى مقرر دراسي	1-n	كل محاضرة لمقرر دراسي واحد، ولكن يمكن أن يكون هناك أكثر من محاضرة للمقرر الدراسي
المحاضرة	**المدرس**	يلقي المحاضرة مدرس	1-n	لكل محاضرة مدرس واحد، والمدرس يمكن أن يلقي أكثر من محاضرة
المحاضرة	**نوع القاعة**	يناسب المحاضرة نوع من القاعات	1-n	كل محاضرة يناسبها نوع معين من القاعات
المحاضرة	**الفئة**	يتلقى المحاضرة مجموعة من الفئات	n-m	المحاضرة تعطى لمجموعة فئات، والفئة لديها أكثر من محاضرة
القاعة	**نوع القاعة**	تنتمي القاعة لنوع قاعات محدد	1-n	كل قاعة لها نوع محدد (قاعة كبيرة، صالة حاسوب...)، ويمكن أن يكون هناك أكثر من قاعة من نفس النوع
السنة الدراسية	**الفئة**	الفئة تنتمي لسنة دراسية	n-1	كل سنة دراسية تحوي عدة فئات، وكل فئة تنتمي لسنة دراسية واحدة
المدرس	**اليوم**	المدرس متاح في أيام محددة	n-m	لكل مدرس مجموعة من الأيام المتاحة بالنسبة له
اليوم	**الفترة**	اليوم يتألف من عدة فترات	n-m	كل يوم يحتوي عدة فترات، وكل فترة تنتمي لأكثر من يوم

الجدول (4) العلاقات بين الكيانات

بعد توضيح العلاقات بين الكيانات وتحديد نوع كل من هذه العلاقات، تم تصميم مخطط العلاقات بين الكيانات الذي يعرض الكيانات والعلاقات فيما بينها. يوضح الشكل التالي مخطط العلاقات بين الكيانات ERD:



الشكل (17) مخطط العلاقات بين الكيانات ERD

2.4.2.4 تصميم ووضع الجداول

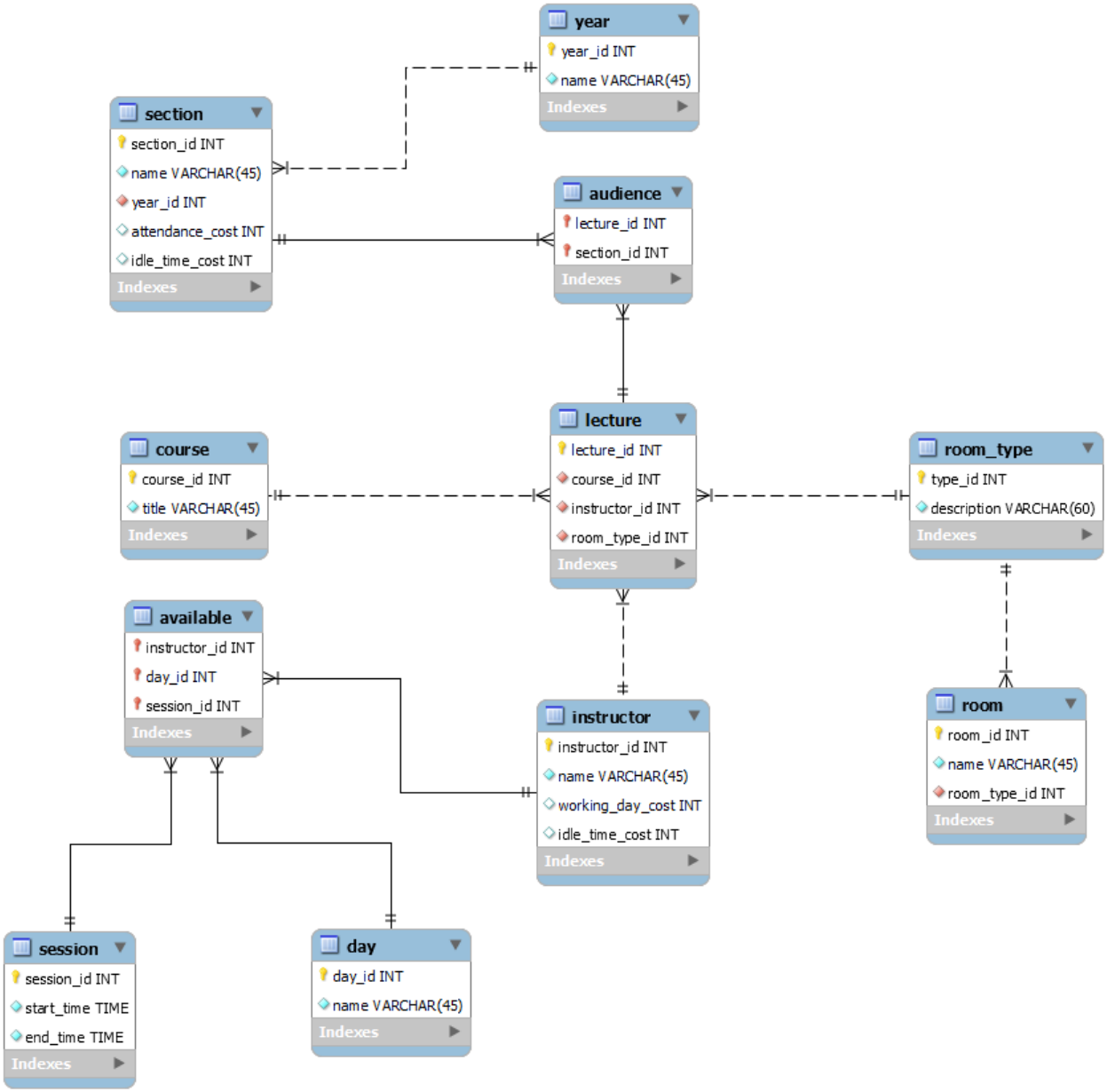
بعد أن قمنا بتعريف الكيانات الأساسية وبيان العلاقات فيما بينها أصبح بإمكاننا وضع الجداول التي تكون بدورها قابلة للبرمجة. تم اعتماد هذه القواعد لتحديد وتصميم الجداول المطلوبة:

- تحويل كل كيان رئيسي إلى جدول ووضع المفتاح الرئيسي الخاص به.
- تحويل كل علاقة عديد إلى عديد m-n إلى جدول وسيط مع إضافة خصائص العلاقة إلى هذا الجدول وإضافة مفتاح أساسي لهذا الأخير مكون من المفتاحين الأساسيين للجدولين اللذان يتم الربط بينهما.

الجداول Tables		
أسم الجدول	**الحقول**	**مصدر الجدول**
الفترات اليومية session	رقم الفترة، وقت البداية، وقت النهاية	كيان
الأيام day	رقم اليوم، أسم اليوم	كيان
الأوقات المتاحة للمدرس Available	*رقم المدرس، رقم اليوم، رقم الفترة*	علاقة m-n
أنواع القاعات room type	رقم نوع القاعة، وصف النوع	كيان
القاعات room	رقم القاعة، أسم القاعة، *رقم نوع القاعة*	كيان
المدرس instructor	رقم المدرس، أسم المدرس، تكلفة يوم عمل المدرس، تكلفة فترة تعطل المدرس	كيان
المقررات الدراسية course	رقم المقرر الدراسي، أسم المقرر	كيان
السنوات الدراسية year	رقم السنة الدراسية، أسم السنة الدراسية	كيان
الفئات section	رقم الفئة، أسم الفئة، *رقم السنة الدراسية*، تكلفة يوم حضور الفئة، تكلفة فترة تعطل الفئة	كيان
المحاضرات lecture	رقم المحاضرة، *رقم المقرر الدراسي، رقم المدرس، رقم نوع القاعة*	كيان
الحضور audience	*رقم المحاضرة، رقم الفئة*	علاقة m-n

الجدول (5) تفاصيل الجداول

يمثل الشكل التالي النموذج المحسن لعلاقات الكيانات Enhanced entity-relationship model:



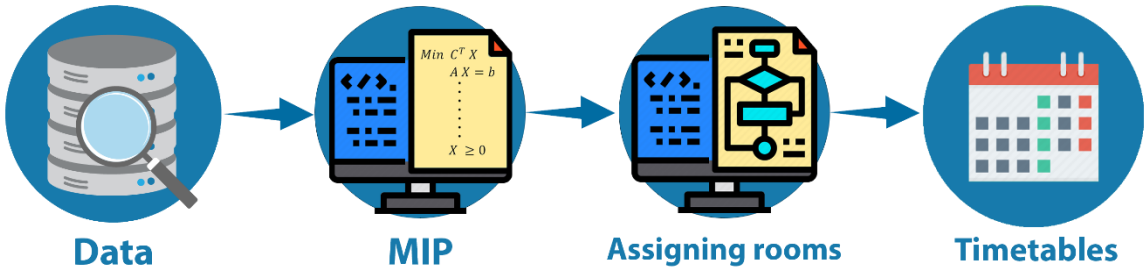
الشكل (18) النموذج المحسن لعلاقات الكيانات

3.4 تصميم النظام

يبدأ هذا القسم بشرح فكرة الحل المقترح وبيان كيف سيقوم النظام بحل مسألة الجدولة الزمنية في المعهد. حيث تتألف عملية الحل من قسمين، في المرحلة الأولى سيتم بناء نموذج البرمجة الطبيعية المختلطة الذي سينتج عن حله تعيين كل محاضرة لفترة زمنية محددة. سيتم بناء النموذج الرياضي وشرح كل مكوناته في قسم مستقل. في المرحلة الثانية سيتم بناء خوارزمية لتعيين القاعات الدراسية لكل محاضرة، أيضا يعرض الفصل هذه الخوارزمية بشكل مبسط بقسم مستقل.

1.3.4 الحل المقترح

كما نعلم إن مسألة الجدولة الزمنية تتضمن تخصيص فترة زمنية محددة وقاعة تدريسية محددة لكل محاضرة درسية. تمت تجزئة عملية حل المسألة إلى مرحلتين يعملان بشكل متكامل مع بعضهما البعض وذلك بهدف تحسين كفاءة عمل النموذج وتخفيض الفترة الزمنية اللازمة لإعداد الحل. المرحلة الأولى هي مرحلة بناء النموذج الرياضي وإيجاد حل وهنا قمنا باستخدام البرمجة الطبيعية المختلطة MIP، ومحرك الحل مفتوح المصدر CBC Solver الذي يعتمد على خوارزمية Branch & Cut، وسيكون هذا النموذج مسؤولاً عن تخصيص فترة زمنية محددة لكل محاضرة من المحاضرات الدرسية (اخذاً بعين الاعتبار مجموعة من المتطلبات والقيود) بالإضافة إلى ضمان عدم وجود نقص في القاعات الدرسية من أجل كل فترة زمنية من الفترات الأسبوعية. أما المرحلة الثانية فتم تطوير خوارزمية تقوم بتخصيص قاعة تدريسية لكل محاضرة درسية (اخذة بعين الاعتبار نوع القاعة التدريسية المناسب).



الشكل (19) خطوات الحل المقترح

تم بناء قاعدة المعطيات من خلال نظام إدارة قواعد المعطيات MySQL. ومن ثم معالجة المعطيات وتهيئتها للاستخدام في عملية بناء النموذج الرياضي باستخدام لغة بايثون Python. ثم تبدأ المرحلة الأولى وهي مسؤولة عن جدولة المحاضرات زمنياً، باستخدام البرمجة الطبيعية المختلطة MIP. تتضمن هذا المرحلة تحقيق جميع المتطلبات الزمنية في عملية الجدولة وهي:

- عدم تعارض المحاضرات مع بعضها البعض من حيث المدرسين أو فئات الطلاب.
- تحقيق قيود ضغط الجدول الزمني بالنسبة للمدرسين أو الفئات (حسب رغبة المستخدم).
- ضمان عدم وضع محاضرات لمدرس ما في الفترات الغير متاحة بالنسبة له.
- ضمان عدم تجاوز الحد الأعلى للقاعات المتاحة في كل فترة زمنية.
- ضمان عدم تجاوز الحد الأعلى لعدد المحاضرات اليومية الممكن إعطاؤها من قبل كل مدرس.

بعد إيجاد حل مقبول أمثلي (أو مقبول غير أمثلي) للبرنامج الطبيعي المختلط تتم ترجمة الحل الذي يمثل مجموعة من المحاضرات التي يرتبط كل منها بفترة زمنية محددة، بعد ذلك ننتقل للمرحلة الثانية من الحل وهي المسؤولة عن تخصيص قاعة تدريسية محددة لكل محاضرة، مع ضمان تطابق نوع القاعة المخصصة مع النوع المطلوب للمحاضرة. وينتج عن هذه المرحلة حل مقبول محقق للمتطلبات يبقى لدينا مهمة ترجمته وعرضه على شكل جداول وهذا ما تم باستخدام مجموعة من الأدوات التي سيتم الحديث عنها في القسم الخاص بعرض واجهات الاستخدام.

1.1.3.4 النموذج الرياضي

قمنا في هذا الفصل بصياغة البرنامج الطبيعي المختلط ابتداء من التعريف بالرموز المستخدمة من مجموعات ومجموعات جزئية ومعاملات وشرح خصائص كل منها، ثم قمنا بشرح القيود الصلبة والمرنة للنموذج وتابع الهدف الخاص به، حيث تم الاعتماد على النموذج الذي أشار إليه [3] في وضع القيود المسؤولة عن ضغط الجدول الزمني، ومن ثم التعريف بالمتغيرات المستخدمة في النموذج، وأخيراً صياغة النموذج بشكل رياضي وشرح الصيغ الرياضية المستخدمة وبيان الغرض من كل منها.

1.1.1.3.4 المجموعات والمعامِلات Sets & Parameters

قبل البدء بعرض القيود وصياغتها ضمن البرنامج الرياضي سنقوم بعرض قائمة المجموعات والمجموعات الجزئية والمعاملات أو الثوابت المستخدمة في النموذج الرياضي.
يوضح الجدول التالي المجموعات الأساسية والمجموعات الجزئية:

المجموعات Sets	
D (Days)	مجموعة تتكون من أيام الأسبوع.
P (Periods)	الفترات اليومية، حيث يقسم كل يوم d الى $\|P\|$ فترة.
TS (Timeslots)	الفترات الأسبوعية $D \times P = \{(d,p)\|d \in D \; and \; p \in P\}$
U	هي مجموعة الثنائيات $P \times P = \{(m,n)\|m,n \in p : n > m+1\}$ أي مجموعة الفترات التي يكون فيما بينها فترة أو أكثر
I (Instructors)	مجموعة المدرسين
S (Sections)	فئات الطلاب (الفئة هي مجموعة الطلاب الذين لديهم نفس المقررات الدراسية)
RT (Room Types)	مجموعة أصناف القاعات التدريسية (مثلا: كبيرة، صغيرة، قاعة حواسيب)
L (Lectures)	هي مجموعة المحاضرات الأسبوعية بحيث تكون كل محاضرة $l \in L$: • تحدث مرة واحدة أسبوعياً • يقوم بها مدرس واحد $i \in I$ • يحضرها مجموعة جزئية من مجموعة الفئات S (من الممكن أن تكون المحاضرة مشتركة بين عدة فئات) • تحدث في نوع محدد من القاعات $rt \in RT$

الجدول (6) المجموعات الأساسية

المجموعات الجزئية Subsets	
$L_i \subseteq L$	مجموعة المحاضرات التي يدرسها المدرس $i \in I$
$L_s \subseteq L$	مجموعة المحاضرات التي المخصصة للفئة $s \in S$
$L_{rt} \subseteq L$	مجموعة المحاضرات التي تناسبها القاعات التدريسية من النوع $rt \in RT$

الجدول (7) المجموعات الجزئية

المعاملات Parameters	
ω_i	تكلفة فترة التعطل بالنسبة للمدرس $i \in I$
γ_i	تكلفة يوم العمل بالنسبة للمدرس $i \in I$ (يعتبر يوم عمل كل يوم يكون للمدرس i محاضرة على الأقل)
φ_i	تكلفة فترة التعطل بالنسبة للفئة $s \in S$
β_i	تكلفة يوم الدوام بالنسبة للفئة $s \in S$ (يعتبر يوم دوام كل يوم يكون للفئة s محاضرة على الأقل)
$A_{i\,ts}$	1: اذا كان المدرس $i \in I$ متاح في الفترة $ts \in TS$ 0: غير ذلك
$CBL_{(l_1,l_2)}$	يعبر هذا المعامل عن التعارض بين المحاضرات Conflict Between Lectures بناء على حضور كل محاضرة. 1: لا يوجد أي فئة مشتركة $s \in S$ بين حضور المحاضرة l_1 وحضور المحاضرة l_2 (لا تعارض بين المحاضرتين من حيث الحضور) 0: غير ذلك
$Limit_i^i$	الحد الأعلى لعدد المحاضرات التي يمكن للمدرس $i \in I$ أن يلقيها في اليوم الواحد
$Limit_{rt}^{rt}$	الحد الأعلى لعدد القاعات المتاحة من النوع $rt \in RT$

الجدول (8) المعاملات

2.1.1.3.4 تابع الهدف والقيود الصلبة والمرنة

يجب أن يحقق الجدول الزمني للمحاضرات في المعهد العالي لإدارة الأعمال بعض القيود ليتم اعتباره جدولاً زمنياً جيداً. سنميز بين نوعين من القيود، القيود الصلبة والقيود المرنة. القيود الصلبة هي القيود التي يجب أن يحققها النموذج للحصول على حل مقبول أو جدول زمني صحيح. بينما تحدد القيود المرنة المتطلبات التي يُرغب بتحقيقها ولكن يمكن انتهاكها في حل ممكن، أي هي التي تحدد جودة الحل.

فيما يلي نناقش مجموعة القيود الصلبة التي تم تحديدها من خلال المقابلات التي جرت مع الإدارة:

H1 كل محاضرة من المحاضرات $l \in L$ يجب أن تحدث مرة ومرة واحدة فقط كل أسبوع.

H2 يمكن للمدرس $i \in I$ أن يلقي محاضرة واحدة على الأكثر في كل فترة زمنية $(d, p) \in TS$

H3 يمكن للمدرس $i \in I$ أن يلقي محاضرة فقط في الفترات المتاحة بالنسبة له.

H4 يمكن أن يكون لكل فئة $s \in S$ محاضرة واحدة على الأكثر في الفترة الزمنية الواحدة.

H5 عدد المحاضرات التي تناسبها القاعات التدريسية من النوع $rt \in RT$ لا يمكن أن يتجاوز الحد الأعلى لعدد القاعات $(Limit_{rt}^{rt})$ من نفس النوع في الفترة الزمنية الواحدة.

H6 عدد المحاضرات التي سيلقيها المدرس $i \in I$ في كل يوم $d \in D$ لا يمكن أن تتجاوز قدرته اليومية على القاء المحاضرات $(Limit_i^i)$.

نلاحظ أننا لم نضف قيد لضمان عدم تعارض المحاضرات في القاعات التدريسية، وذلك لأننا أسندنا مهمة تعيين القاعات التدريسية للمحاضرات لخوارزمية مستقلة سنقوم بعرضها في القسم التالي. إلا أننا قد أضفنا قيداً (H5) يضمن عدم وجود نقص في القاعات التدريسية. وبالتالي يضمن هذا القيد أن يكون الحل الذي سنحصل عليه - بعد حل النموذج الرياضي - دخلاً مقبولاً بالنسبة لخوارزمية تعيين القاعات.

كما ذكرنا من قبل، فإن هذه القيود الصلبة أساسية لتوفير جدول زمني مقبول. أما القيود المرنة فهي تلك الشروط التي من شأنها أن تحسن من جودة الجدول الزمني ولكنها لا تؤثر على صحة الحل. في هذا النموذج، القيود المرنة هي التي تؤثر على قيمة تابع الهدف وتوجه الحل ليحقق أكبر قدر ممكن من ضغط الجدول الزمني بالنسبة للمدرسين والطلاب. فيما يلي مجموعة القيود المرنة:

S1 كل مدرس من المدرسين لديه أقل عدد ممكن من أيام العمل.

S2 كل مدرس من المدرسين لديه أقل عدد ممكن من أوقات التعطل.

S3 كل فئة من الفئات لديها أقل عدد ممكن من أيام الدوام.

S4 كل فئة من الفئات لديها أقل عدد ممكن من أوقات التعطل.

يعتبر يوم عمل بالنسبة لمدرس ما إذا كان لديه محاضرة واحدة على الأقل في هذا اليوم.

يعتبر يوم دوام لفئة ما إذا كان لديها محاضرة واحدة على الأقل في هذا اليوم.

فترة التعطل هي الفترة التي يكون لدى المدرس أو الفئة وقت فراغ بين محاضرتين.

3.1.1.3.4 النموذج الرياضي

نبدأ أولا بتعريف متغيرات القرار والمتغيرات المساعدة كما هو موضح في الجدول التالي:

Decision Variables:	
$X_{l,\,ts} \in \{1,0\} = \begin{cases}1\\0\end{cases}$	*1*: إذا تم تعيين المحاضرة l في المدة الزمنية ts *0*: غير ذلك
Auxiliary Variables:	
$Y_{i\,d} \in \{1,0\} = \begin{cases}1\\0\end{cases}$	*1*: إذا كان للمدرس i في اليوم d محاضرة واحدة على الأقل *0*: غير ذلك
$Z_{i\,d\,p} \in \{1,0\} = \begin{cases}1\\0\end{cases}$	*1*: إذا كانت الفترة الزمنية *(day, period)* تمثل فترة تعطل بين فترتين مشغولتين بالنسبة للمدرس i *0*: غير ذلك
$W_i \in \mathbb{Z}^+$	اجمالي فترات التعطل بالنسبة للمدرس i
$L_{s\,d} \in \{1,0\} = \begin{cases}1\\0\end{cases}$	*1*: إذا كان للفئة (مجموعة الطلاب) s في اليوم d محاضرة واحدة على الأقل *0*: غير ذلك
$D_{s\,d\,p} \in \{1,0\} = \begin{cases}1\\0\end{cases}$	*1*: إذا كانت الفترة الزمنية *(day, period)* تمثل فترة تعطل بين فترتين مشغولتين بالنسبة للفئة s *0*: غير ذلك
$M_s \in \mathbb{Z}^+$	اجمالي فترات التعطل بالنسبة للفئة s

الجدول (9) متغيرات القرار والمتغيرات المساعدة

بناء على ما سبق، نحصل على الصيغة الرياضية للبرنامج الطبيعي المختلط.

$$\textbf{Minimize} \sum_{i \in I} \omega_i W_i + \sum_{i \in I} \sum_{d \in D} \gamma_i Y_{i\,d} + \sum_{s \in S} \varphi_s M_s + \sum_{s \in S} \sum_{d \in D} \beta_s L_{s\,d}$$

$\sum_{ts \in TS} X_{l,\,ts} = 1$	$\forall\, l \in L$	(1)
$X_{l,\,ts} = 0$	$\forall\, i \in I, \quad \forall\, l \in L_i,$ $\forall\, ts \in TS, \quad if\ A_{i\,ts} = 0$	(2)
$\sum_{l \in L_i} X_{l,\,ts} \leq 1$	$\forall\, ts \in TS, \quad \forall\, i \in I$	(3)
$X_{l_1\,ts} + X_{l_2\,ts} \leq 1 + CBL_{(l_1,l_2)}$	$\forall\, ts \in TS,$ $\forall\, l_1, l_2 \in L\ if\ l_1 < l_2$	(4)
$\sum_{l \in L_i} X_{l\,ts} \leq Y_{i\,d}$	$\forall\, i \in I, \quad \forall\, ts = (d, p) \in TS$	(5)
$Z_{i\,d\,p} \geq -1 + \sum_{l \in L_i} (X_{l\,(d,m)} + X_{l\,(d,n)} - X_{l\,(d,p)})$	$\forall\, i \in I, \ \forall\, (d, p) \in TS,$ $\forall\, (m, n) \in U : m < p < n$	(6)
$W_i = \sum_{(d,p) \in TS} Z_{i\,d\,p}$	$\forall\, i \in I$	(7)
$Z_{i\,d\,p} \leq Y_{i\,d}$	$\forall\, i \in I, \quad \forall\, (d, p) \in S$	(8)
$\sum_{l \in L_s} X_{l\,ts} \leq L_{s\,d}$	$\forall\, s \in S, \quad \forall\, ts = (d, p) \in TS$	(9)
$D_{s\,d\,p} \geq -1 + \sum_{l \in L_s} (X_{l\,(d,m)} + X_{l\,(d,n)} - X_{l\,(d,p)})$	$\forall\, s \in S, \ \forall\, (d, p) \in TS,$ $\forall\, (m, n) \in U : m < p < n$	(10)
$M_s = \sum_{(d,p) \in TS} D_{s\,d\,p}$	$\forall\, s \in S$	(11)
$D_{s\,d\,p} \leq L_{s\,d}$	$\forall\, s \in S, \quad \forall\, (d, p) \in TS$	(12)

$\sum_{l \in L_i} \sum_{p \in P} X_{l\,(d,p)} \leq Limit_i^i$	$\forall\, i \in I, \quad \forall\, d \in D$	(13)
$\sum_{l \in L_{rt}} X_{l\,ts} \leq Limit_{rt}^{rt}$	$\forall\, ts \in TS, \quad \forall\, rt \in RT$	(14)

$X_{l,\,ts} \in \{1,0\}$	$\forall\, l \in L,$	$\forall\, ts \in TS$
$Z_{i\,d\,p}^{i} \in \{1,0\}$	$\forall\, i \in I,$	$\forall\, (d,p) \in TS$
$Y_{i\,d} \in \{1,0\}$	$\forall\, i \in I,$	$\forall\, d \in D$
$W_i \in \mathbb{Z}^+$	$\forall\, i \in I$	
$Z_{s\,d\,p}^{s} \in \{1,0\}$	$\forall\, s \in S,$	$\forall\, (d,p) \in TS$
$L_{s\,d} \in \{1,0\}$	$\forall\, s \in S,$	$\forall\, d \in D$
$M_s \in \mathbb{Z}^+$	$\forall\, s \in S$	

4.1.1.3.4 شرح النموذج

- تضمن مجموعة القيود (1) أن كل المحاضرات ضمن مجموعة المحاضرات L سوف تتم جدولتها.
- تضمن مجموعة القيود (2) عدم تعيين المحاضرات في الفترات التي يكون مدرس هذه المحاضرات غير متاح.
- تضمن مجموعة القيود (3) ألا يتم تعيين أكثر من محاضرة واحدة للمدرس في الفترة الزمنية الواحدة.
- تضمن مجموعة القيود (4) ألا تجتمع محاضرتان تشتركان في فئة من الطلاب أو أكثر في فترة زمنية واحدة.
- تحدد مجموعة القيود (5) عدد أيام عمل كل مدرس.
- تحدد مجموعة القيود (6) و (7) عدد فترات التعطل لكل مدرس.

في مجموعة القيود (6) عندما يتم تفعيل المتغيرين $X_{l\,(d,m)}$ و $X_{l\,(d,n)}$ (أي عندما يأخذان القيمة 1) والفترة الزمنية (d,p) فارغة، حتماً سيأخذ المتغير $Z_{i\,d\,p}^{i}$ القيمة 1. أما مجموعة القيود (7) فستقوم بحساب قيمة W_i في دالة الهدف.

- تؤكد مجموعة القيود (8) أن المتغيرات الخاصة بأوقات تعطل كل مدرس حتماً ستكون قيمتها صفر في حال كان اليوم الذي تنتمي إليه هذه الفترة يوم عطلة بالنسبة لهذا المدرس. لا يغير هذه القيد من النتيجة النهائية إلا أن استخدامه قد حسن كفاءة الحل.
- تحدد مجموعة القيود (9) عدد أيام دوام كل فئة طلاب.

- تحدد مجموعة القيود (10) و (11) عدد فترات التعطل لكل فئة طلاب.

في مجموعة القيود (10) عندما يتم تفعيل المتغيرين $X_{l\,(d,m)}$ و $X_{l\,(d,n)}$ (أي عندما يأخذان القيمة 1) والفترة الزمنية (d,p) فارغة، حتماً سيأخذ المتغير $Z^{s}_{s\,d\,p}$ القيمة 1. أما مجموعة القيود (11) فستقوم بحساب قيمة M_s في دالة الهدف.

- تؤكد مجموعة القيود (12) أن المتغيرات الخاصة بأوقات تعطل كل فئة حتماً ستكون قيمتها صفر في حال كان اليوم الذي تنتمي إليه هذه الفترة يوم عطلة بالنسبة لهذه الفئة.
- تضمن مجموعة القيود (13) ألا يتجاوز عدد محاضرات كل مدرس في كل يوم الحد الأعلى لقدرة المدرس.
- تضمن مجموعة القيود (14) ألا يتجاوز عدد المحاضرات - التي تتطلب نفس النوع من القاعات - في كل فترة عدد القاعات المتاحة في هذه الفترة.

2.1.3.4 خوارزمية تعيين القاعات

كما ذكرنا سابقا، إن حل النموذج الرياضي الذي سبق توضيحه في القسم السابق يمثل الدخل الرئيسي لخوارزمية تعيين القاعات بالإضافة لبعض البيانات الأخرى مثل القاعات المتاحة في المعهد ونوع كل منها، والفترات الأسبوعية. أما بالنسبة لخرج هذه الخوارزمية فهو حل مقبول محقق لجميع المتطلبات يبقى لدينا مهمة ترجمته وعرضه على شكل جداول أسبوعية.

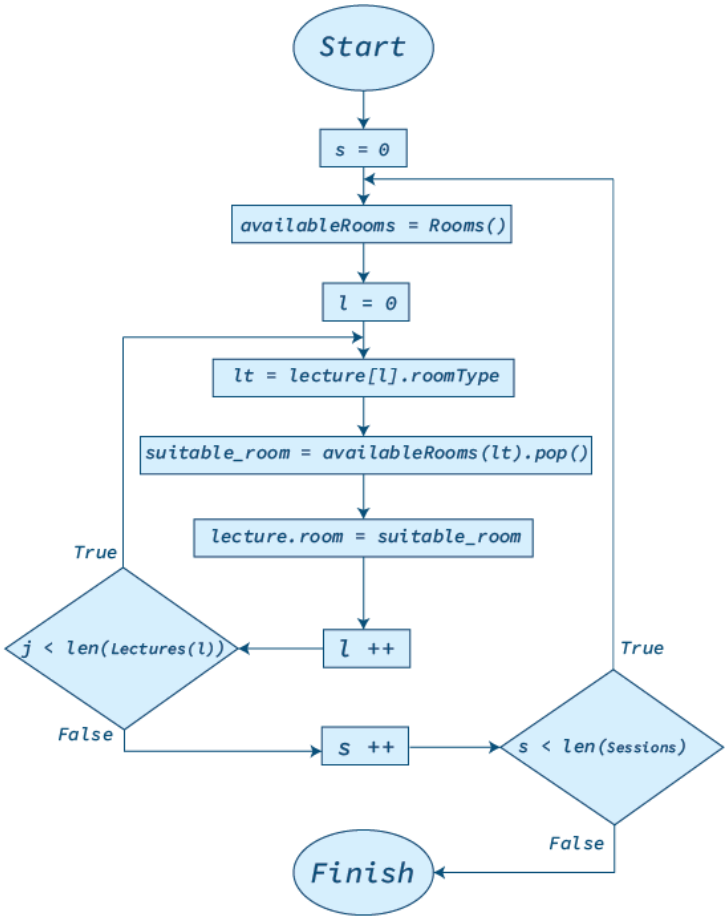
إن المدخلات الأساسية للخوارزمية هي أربعة قوائم موضحة بالجدول التالي:

قائمة الفترات الأسبوعية S	$weekly\ sessions\ [1, 2, 3, \dots \dots m]$
قائمة أنواع القاعات T	$room\ types = [1, 2, 3, \dots \dots . n]$
قائمة المحاضرات L	$lecture = [(id, roomTypeID, sessionId) \dots\dots\dots\dots\dots]$
قائمة القاعات الدراسية R	$room = [(id, roomTypeID) \dots\dots\dots\dots\dots]$

الجدول (10) مدخلات خوارزمية تعيين القاعات

يتم أولا تصنيف البيانات التي سبق ذكرها. حيث يتم تصنيف المحاضرات حسب الفترة الزمنية لها، ويتم تصنيف القاعات حسب نوعها.

بعد معالجة البيانات السابقة وإعدادها بالشكل المناسب يتم تشغيل الخوارزمية والتي تم توضيح كيفية عملها بشكل مبسط من خلال الشكل التالي:



الشكل (20) مخطط تدفق خوارزمية تعيين القاعات

إن الدالة Rooms() هي دالة ترجع قائمة مصنفة بالقاعات المتاحة في كل فترة زمنية في كل مرة يتم فيها استدعائها. أي أن وظيفة هذه الدالة هي إعادة ملء قائمة القاعات عند الانتقال من فترة زمنية إلى فترة زمنية أخرى. الدالة pop تقوم بإرجاع اخر قيمة في القائمة List ثم تقوم بحذفها من القائمة. أما الدالة len() ترجع عدد العناصر في قائمة ما.

تقوم الخوارزمية من أجل كل فترة زمنية بالمرور على جميع المحاضرات في هذه الفترة، وتقوم بتعيين قاعة دراسية مناسبة لكل محاضرة تمر عليها. يوضح الكود التالي المكتوب بلغة بايثون كيفية عمل الخوارزمية.

```
#    S: List of Weekly Sessions
# RT(): Method that return categorized available Rooms by RoomType
#   LS: Categorized lectures by sessions
#   LT: Lecture RoomType
#   LR: Lecture Room
for s in S:  # for every session in Sessions List
    Copy_R = RT()
    for l in LS[s]:  # L[s] is a list of lectures in session 's'
        lt = LT[l]  # 'lt' is the right type of room for lecture 'l'
        LR[l] = Copy_R[lt].pop()  # assign room from R[lt] to lecture 'l',
and delete this room from R[lt] list
```

4.4 تنجيز النظام

مرحلة التنجيز هي المرحلة الأخيرة من مراحل تطوير النظام وهي المرحلة التي يتم من خلالها برمجة النظام باستخدام مجموعة من الأدوات اعتماداً على النتائج التي توصلنا إليها في مرحلة التصميم. في هذا القسم سيتم عرض الأدوات المستخدمة في تطوير نظام TSwift ابتداء من نظام إدارة قواعد المعطيات وصولاً إلى لغة البرمجة ومجموعة المكتبات المستخدمة. بالإضافة على عرض واجهات الاستخدام وشرحها ومناقشة مجموعة من الاختبارات التي تم إجراؤها على النظام.

1.4.4 الأدوات المستخدمة

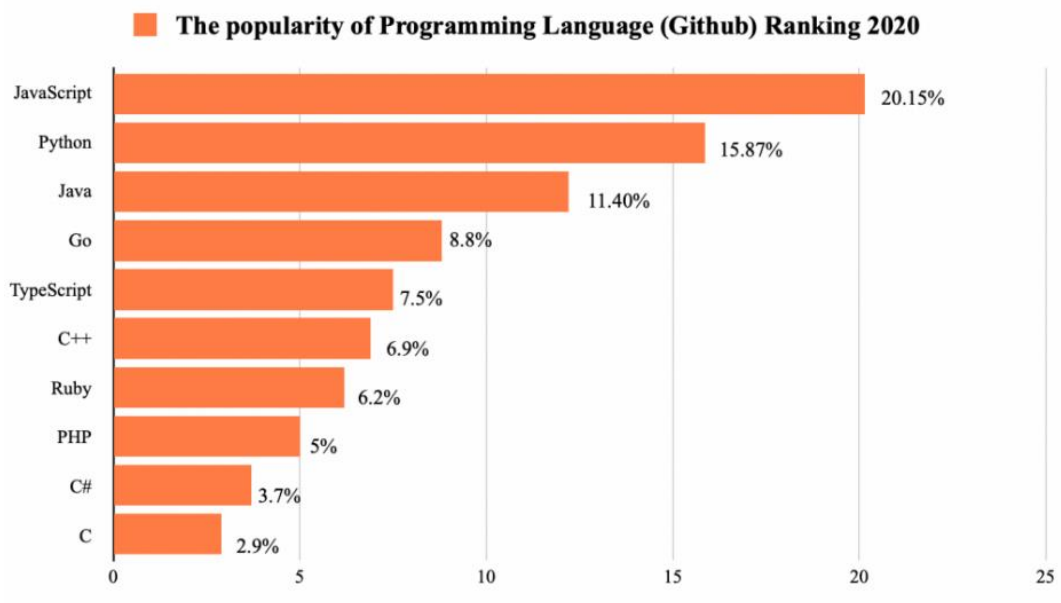
يعرض هذا القسم مجموعة الأدوات المستخدمة في برمجة النظام. حيث يبدأ بالحديث عن لغة البرمجة المستخدمة وهي لغة بايثون Python وخصائصها وميزاتها ومبررات استخدامها بالإضافة إلى مجموعة المكتبات المستخدمة والمبنية باستخدام هذه اللغة. بالإضافة إلى نظام إدارة قواعد المعطيات MySQL والذي تم استخدامه في بناء قاعدة المعطيات الخاصة بالنظام.

1.1.4.4 لغة بايثون Python

الشكل (21) شعار لغة بايثون

تمت عملية برمجة وتطوير النظام باستخدام لغة بايثون Python، وهي لغة برمجة عالية المستوي High Level Language تم تطويرها من قبل المبرمج الهولندي Guido Van Rossum وإطلاقها في عام 1990. وهي تتميز عن أغلب لغات البرمجة الأخرى بسهولة كتابة وقراءة الكود بالإضافة إلى عدم الحاجة لكتابة الكثير من الأسطر لتطبيق أمر بسيط كما هو الحال في لغات البرمجة الأخرى. تدعم لغة بايثون أيضاً البرمجة غرضية التوجه OOP، مما يجعلها لغة برمجة متعددة الاستخدام في العديد من المجالات. فيمكن من خلالها بناء تطبيقات الويندوز بسهولة وهذا يعد أحد المبررات الأساسية لاختيار هذه اللغة. حيث تم الاعتماد على البرمجة غرضية التوجه في بناء كامل النظام.

لا شك أن النظام الذي تم تطويره قابل للتحديث وإضافة الميزات في المستقبل، وبالتالي فإن الاعتماد على لغة شائعة الاستخدام بشكل كبير أمر أساسي لكي يتحقق هذا الغرض. ففي فترة قليلة من الزمن أصبحت لغة بايثون تحتل درجة عالية من اهتمام المبرمجين حول العالم حيث احتلت المرتبة الثانية كأشهر لغة برمجة عالمياً حسب تصنيف GitHub. ويرجع هذا السبب لوجود مجموعة ضخمة من المكتبات الخاصة بهذه اللغة والتي توفر الكثير من الوقت والجهد، حيث انها تساعد المبرمجين المحترفين منهم والمبتدئين على القيام بالمشاريع البرمجية بكل سهولة. وكل هذا يمكن استخدامه بشكل مجاني فهي لغة مفتوحة المصدر غير تابعة لأي شركة.



الشكل (22) تصنيف لغات البرمجة حسب شهرتها لعام 2020 حسب Github

ما يميز لغة Python والذي دفعنا لاستخدامها هو:

أ. لغة سهلة في الكتابة والقراءة
ب. لغة مجانية مفتوحة المصدر
ت. لها العديد من المكتبات التي تسهل العمل في جميع المجالات
ث. تعمل على جميع أنظمة التشغيل
ج. لغة شائعة الاستخدام بشكل كبير

تستخدم لغة بايثون في العديد من المجالات مثل تطوير تطبيقات سطح المكتب، برمجة مواقع الويب وحتى برمجة تطبيقات الهواتف النقالة والعديد من الاستخدامات الأخرى.

2.1.4.4 واجهة المستخدم الرسومية المستخدمة (GUI)

تم بناء واجهات النظام الرسومية باستخدام مكتبة Tkinter وهي واجهة المستخدم الرسومية الرسمية للغة بايثون وتأتي بشكل افتراضي مع حزم تثبيت Python.

يأتي اسم Tkinter من واجهة Tk وهي عبارة عن حزمة Tcl مخصصة لإنشاء أدوات عناصر واجهة المستخدم وهي مكتبة مجانية ومفتوحة المصدر، توفر Tk مكتبة من العناصر الأساسية لعناصر واجهة المستخدم الرسومية لبناء واجهة مستخدم رسومية (GUI) في العديد من لغات البرمجة. كما توفر Tk عدداً من عناصر واجهة المستخدم المطلوبة بشكل شائع لتطوير تطبيقات سطح المكتب، مثل الأزرار والقوائم ومربعات النصوص والإطارات وما إلى ذلك. تم تدعيم Tk لتعمل على جميع أنظمة التشغيل.

الشكل (23) شعار حزمة Tcl\Tk

3.1.4.4 أدوات النمذجة الرياضية

يناقش هذا الفصل مجموعة الأدوات البرمجية التي تم استخدامها في عملية النمذجة الرياضية وكذلك حل النموذج الرياضي.

1.3.1.4.4 مكتبة Numeric Python (NumPy)

الشكل (24) شعار مكتبة NumPy

هي الحزمة الأساسية للحوسبة العلمية في لغة بايثون. هي مكتبة توفر إمكانية إنشاء مصفوفات متعددة الأبعاد وغير ذلك من الأنواع المختلفة من المصفوفات، بالإضافة إلى توفير مجموعة متنوعة من العمليات التي يمكن إجراؤها على هذه المصفوفات مثل تغيير شكل المصفوفة (أبعادها) والتصنيف والاختيار والجبر الخطي وعمليات إحصائية وغير ذلك. وهي أيضا مكتبة مجانية ومفتوحة المصدر.

بالحقيقة إنه يمكن استخدام القوائم lists من لغة بايثون مباشرة والتعامل معها وكأنها مصفوفات إلا أن ذلك سيكون بطيء للغاية، فالتعامل مع المصفوفات من مكتبة NumPy أسرع بما يقارب 50 مرة. تمت كتابة هذه المكتبة باستخدام لغة Python ولكن بعض الأجزاء التي تتطلب سرعة معالجة كبيرة تمت كتابتها باستخدام C و C++

تم استخدام هذه المكتبة في نظام TSwift لتهيئة مدخلات البرنامج الرياضي على شكل مصفوفات. بالإضافة إلى أن أداة النمذجة المستخدمة (Python MIP) تعتمد على مكتبة NumPy في تنظيم متغيرات النموذج ووضعها في مصفوفات.

2.3.1.4.4 أداة النمذجة Python-MIP

كما ذكرنا سابقاً تحوي Python على كم هائل من المكتبات التي يمكن إعادة استخدامها، ولغرض بناء النموذج الرياضي برمجياً تمت الاستعانة بحزمة Python-MIP وهي عبارة عن مجموعة من الأدوات التي تستخدم في نمذجة وحل البرامج الخطية الصحيحة المختلطة MIPs باستخدام محركات حل (Solvers) خارجية مثل Gurobi وCBC. تم استخدام هذه المكتبة في نظام TSwift لبناء نموذج البرمجة الطبيعية المختلط MIP.

ما يميز MIP-Python عن بقية حزم النمذجة هو:

1. السرعة.
2. سهولة ووضوح النمذجة.
3. تدعم لأكثر من محرك حل Solver
4. تدعم عدد كبير من أنواع القيود.

3.3.1.4.4 أداة الحل CBC Solver

أثناء البحث عن أداة لحل نموذجنا صادفنا العديد من الخيارات المتاحة فنجد بعض الأدوات مجانية وبعضها مدفوع، كما وجدنا بعض محركات الحل Solvers غير متوافقين مع أداة النمذجة، لذا وقع خيارنا على CBC Solver.

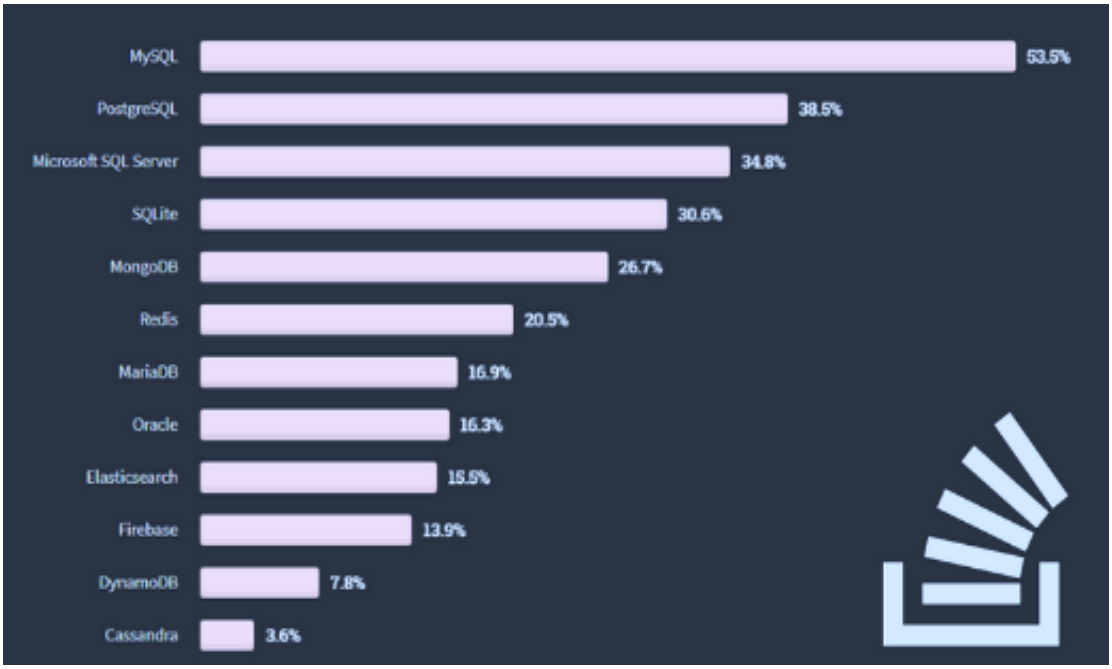
CBC (Coin-or branch and cut) وهو عبارة عن أداة حل للبرمجة الطبيعية المختلطة وهو مجاني ومفتوح المصدر مكتوب بلغة C. يمكن استخدامه كمكتبة قابلة للاستدعاء أو باستخدام ملف تنفيذي مستقل. حيث يمكن استخدامه بعدة طرق من خلال أنظمة وحزم النمذجة المختلفة، يتميز بسرعة في الحل وقابليته لحل عدد لا متناهي من القيود.

4.1.4.4 نظام إدارة قاعدة المعطيات

نظراً للكم الكبير للبيانات التي يتعامل معها التطبيق (محاضرات، مقررات، قاعات، مدرسين، ...) قمنا بإنشاء قاعدة بيانات علائقية Relational Database وإدارتها باستخدام نظام MySQL وهو نظام لإدارة قواعد البيانات العلائقية (RDBMS) تم تطويره بواسطة Oracle والذي يعتمد على لغة الاستعلام المهيكلة SQL.

الشكل (25) شعار نظام MySQL

تعتبر أهم سمات نظام قواعد البيانات MySQL السرعة والثبات، الشيء الذي يفسر كثرة استخدامه من قبل المطورين والمدراء والمستخدمين حول العالم حسب استطلاع اجراه موقع Stack Overflow وهو تجمع للمبرمجين من كافة الاختصاصات والجنسيات.



الشكل (26) تصنيف أنظمة إدارة قواعد المعطيات حسب Stack Overflow

وفيما يلي بعض مميزات هذا النظام التي دفعتنا لاستخدامه:

1. سرعة MySQL

يعتبر الوقت المستغرق لتنفيذ استعلام وإرجاع النتائج للمستعلم من اهم مميزات نظام إدارة قواعد البيانات العلائقية MySQL، وذلك يعود لكون MySQL تستخدم بنية متعددة المهام مثل الفهرسة والعقد، والاستعلامات المخبئة في الذاكرة، الشيء الدي ينتج عنه الأداء العالي بدون الحاجة لأي برمجة مخصصة من قبل المستخدم.

2. الأمن في MySQL

يعتبر الأمن من النقاط القوية في MySQL بحيث يأتي بنظام معقد للتحكم بالوصول و نظام صلاحيات ليمنع المستخدمين غير المصرحين من الوصول إلى قاعدة المعطيات، هذا النظام طبق على شكل خمس طبقات من الصلاحيات بشكل هرمي، مما يمكن مدراء MySQL من حماية الوصول إلى البيانات الحساسة، ويمكن حد المستخدمين ليؤدوا العمليات على قواعد بيانات معينة أو حقول معينة فقط، و تسمح MySQL أيضا بإمكانية التحكم في أنواع الاستعلامات التي يمكن للمستخدم أن يشغلها على مستوى قاعدة المعطيات أو الجدول أو الحقل.

3. سهولة استخدام MySQL

تعتبر سهولة الاستخدام من النقاط المهمة التي ركزت عليها MySQL، إن الواجهة الأساسية لمخدم MySQL هي واجهة سطرية بسيطة، والتي تتكون من واجهتين رسوميتين وهما MySQL Control Center وMySQL Administrator.

4. دعم مختلف لغات البرمجة

تقدم MySQL واجهة برمجية لمختلف لغات البرمجة لتمكنك من كتابة تطبيقات قواعد البيانات باللغة التي تختارها فهي تدعم PHP وJava وC++ وPerl وPython وTcl وغيرها لتعطي المطورين الحرية القصوى في تصميم التطبيقات التي تعتمد على MySQL.

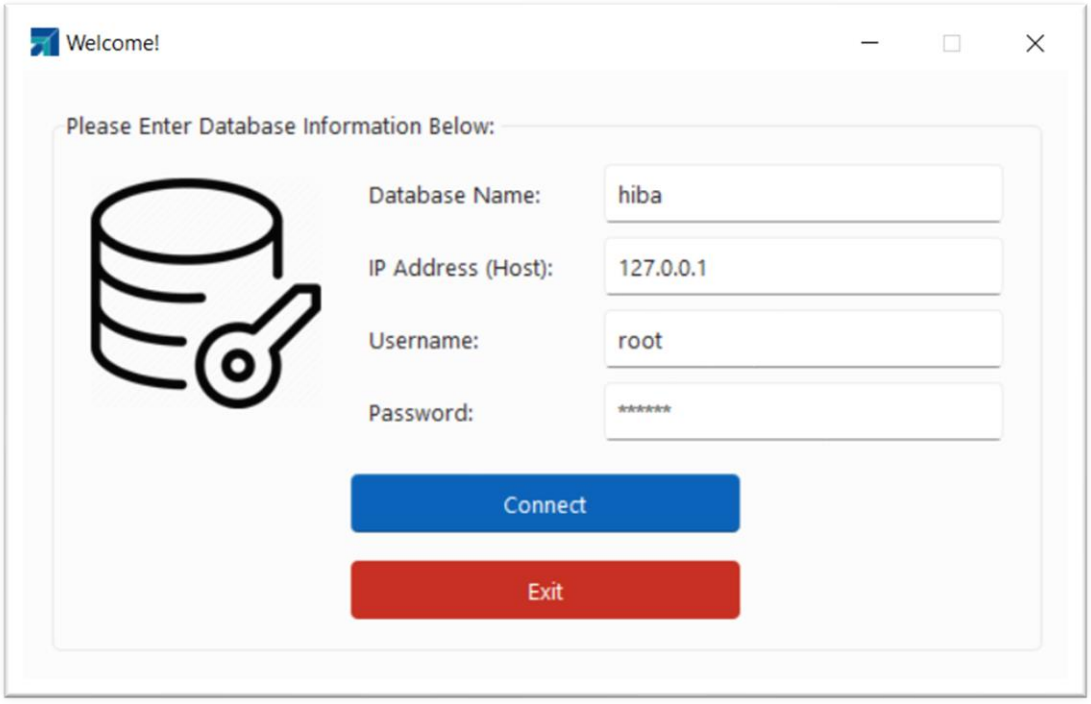
2.4.4 تطبيق النظام

من خلال هذه القسم سيتم عرض واجهات الاستخدام التي سيتم من خلالها التعامل مع النظام. ومن ثم إجراء مجموعة من الاختبارات ومناقشتها بهدف بيان كفاءة عمل النظام.

1.2.4.4 عرض واجهات الاستخدام

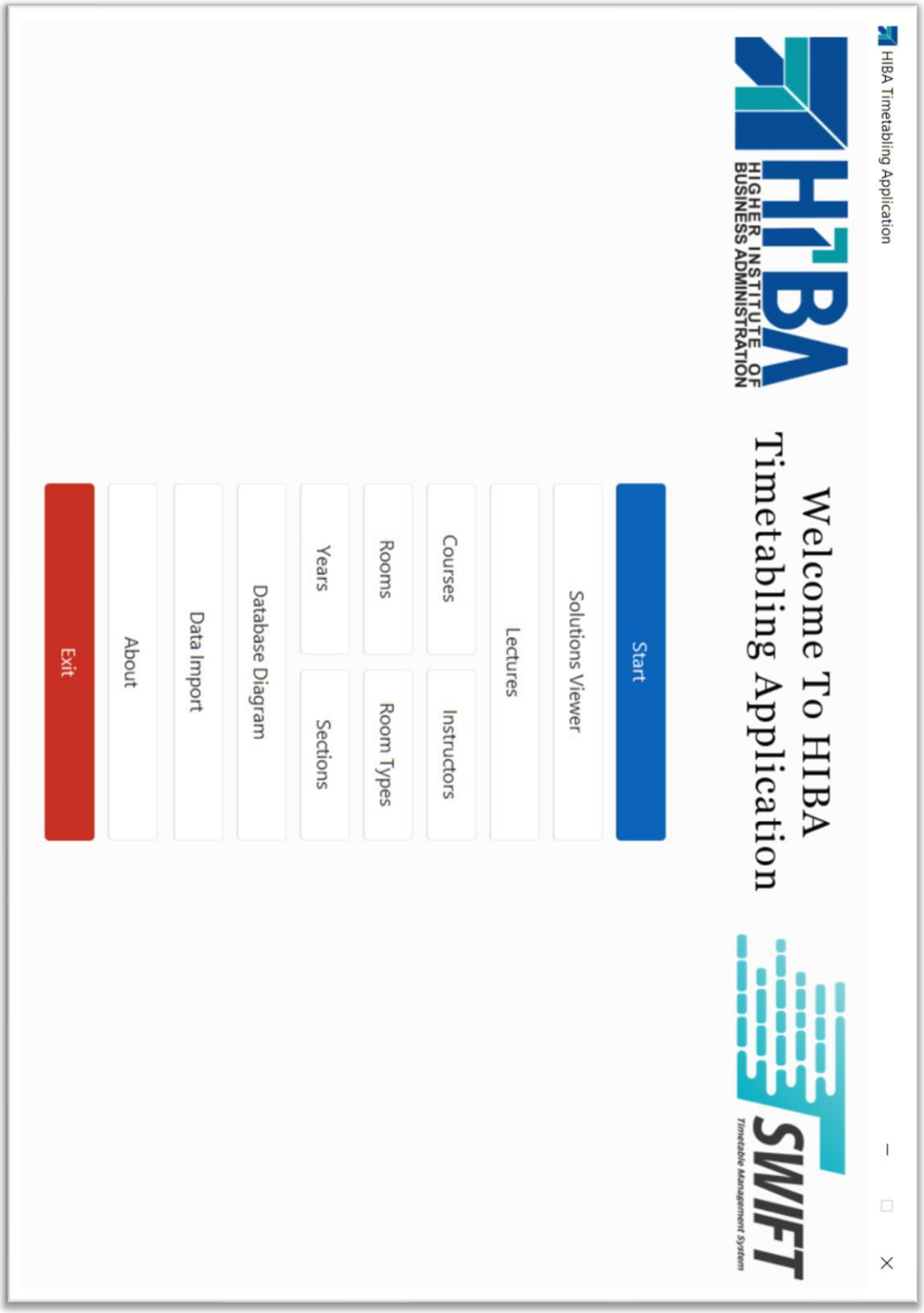
تم تصميم مجموعة من واجهات الاستخدام خصيصاً لإدخال البيانات إلى قاعدة المعطيات بسرعة وسهولة. سنبدأ بعرض هذه الواجهات مع توضيح خصائص ومكونات كل منها.

يتم أولا الاتصال بقاعدة المعطيات من خلال إدخال اسم قاعدة المعطيات واسم المستخدم وكلمة المرور كما هو موضح في الشكل (27).



الشكل (27) واجهة الاتصال مع قاعدة المعطيات

بعد أن يتم الاتصال بقاعدة المعطيات تظهر الواجهة الرئيسية (الشكل 28) والتي يتم من خلالها تحديد ما هو المطلوب إجراؤه وسيتم الانتقال إلى كل الواجهات الأخرى عن طريقها.

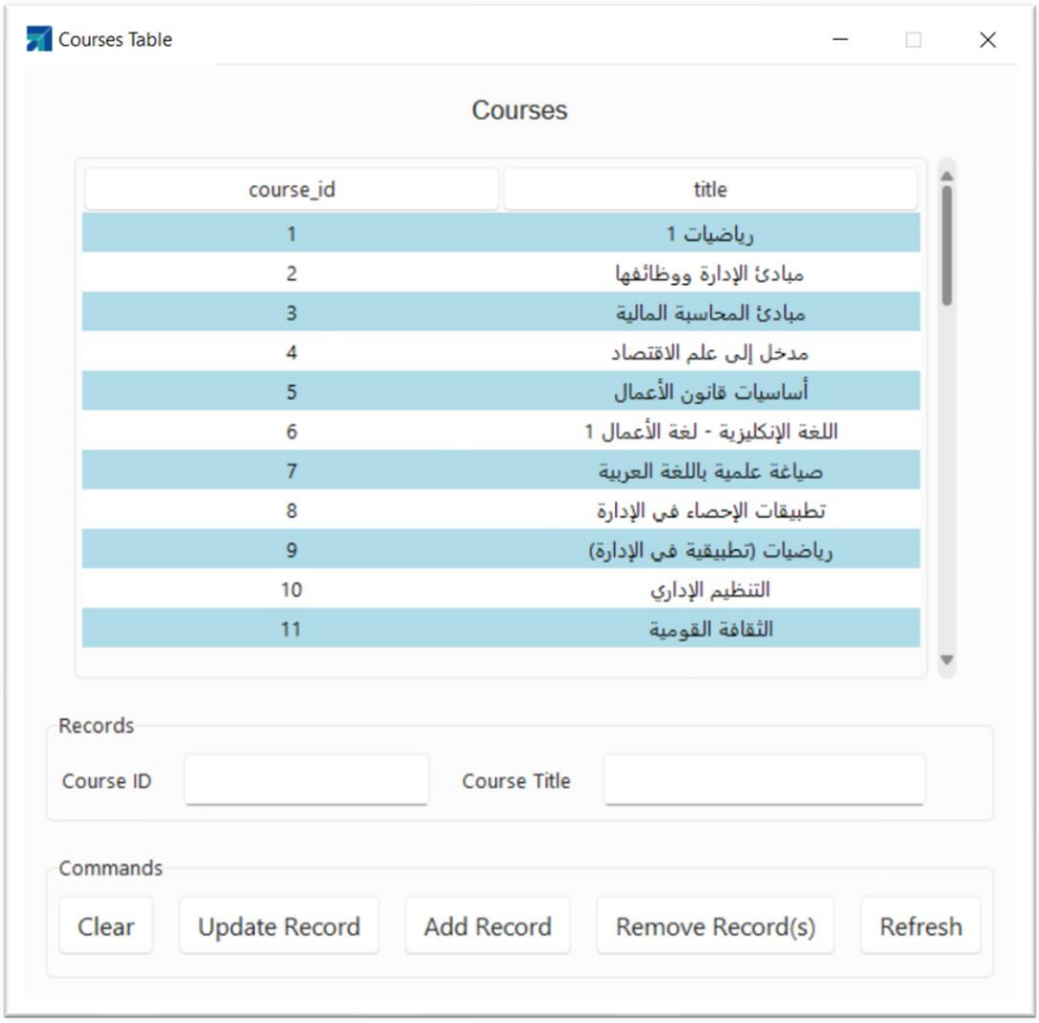


الشكل (28) واجهة الاستخدام الرئيسية

من خلال الواجهة Data import يمكن أن تتم عملية إضافة جميع البيانات بسرعة وسهولة عن طريق استيرادها من ملف Excel (الشكل 29). حيث يتم استيراد البيانات الموجودة في هذا الملف وإضافتها إلى قاعدة المعطيات.

الشكل (29) واجهة استيراد البيانات

إن لم يكن ملف ال Excel المراد سحب البيانات منه موجوداً يمكن إضافة البيانات بشكل يدوي من خلال مجموعة من الواجهات الأخرى. فعلى سبيل المثال يمكن إضافة مقرر بالدخول إلى واجهة Courses ومن ثم أضافة اسم المقرر والضغط على Add record (الشكل 30). كما يمكن حذف أو تعديل البيانات السابقة من خلال Update record وRemove Record.



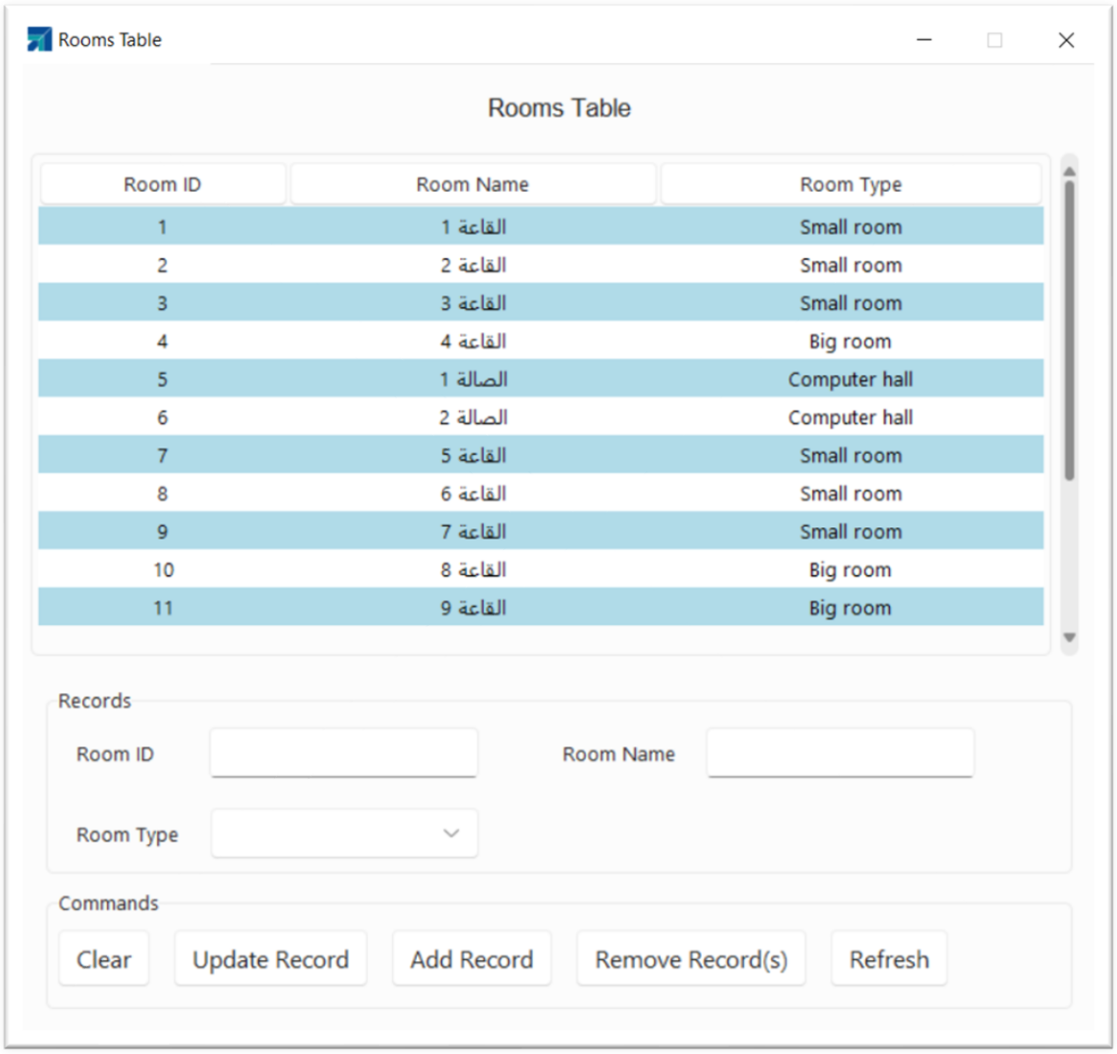
الشكل (30) واجهة إدخال المقررات

وبالمثل يمكن إضافة أنواع جديدة من القاعات التدريسية والمدرسين والسنوات الدراسية كما هو موضح في الشكل (31) و (32)

الشكل (31) واجهة إدخال المدرسين

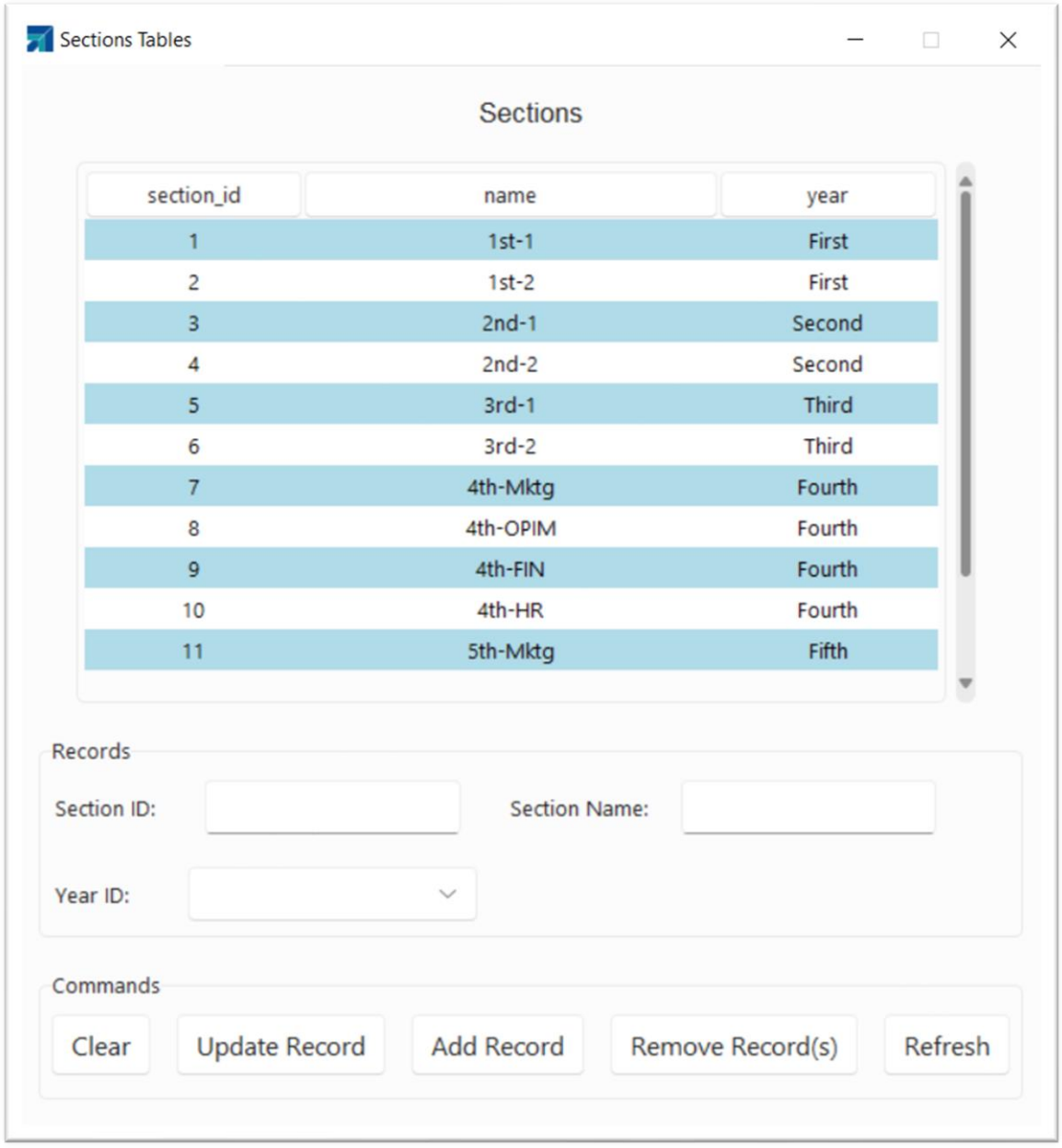
الشكل (32) واجهة إدخال السنوات الدراسية

إن كل البيانات السابقة ليست مرتبطة ببيانات أخرى (لا تحتوي على مفاتيح ثانوية) وبالتالي يجب البدء في إدخالها قبل الانتقال إلى البيانات الأخرى. بعد ذلك يمكن إدخال القاعات التدريسية الموجودة في المعهد مع تحديد أسم ونوع كل منها وذلك من خلال الواجهة الموضحة في الشكل (33).



الشكل (33) واجهة إدخال القاعات الدراسية

نلاحظ في هذه الحالة وجود مفتاح ثانوي مرتبط بجدول أنواع القاعات Room Types لذلك يجب إدخال القاعات التدريسية بعد إدخال الأنواع المتاحة. بالمثل لدينا فئات الطلاب Sections حيث يرتبط بجدول السنوات الدراسية Years.



الشكل (34) واجهة إدخال فئات الطلاب

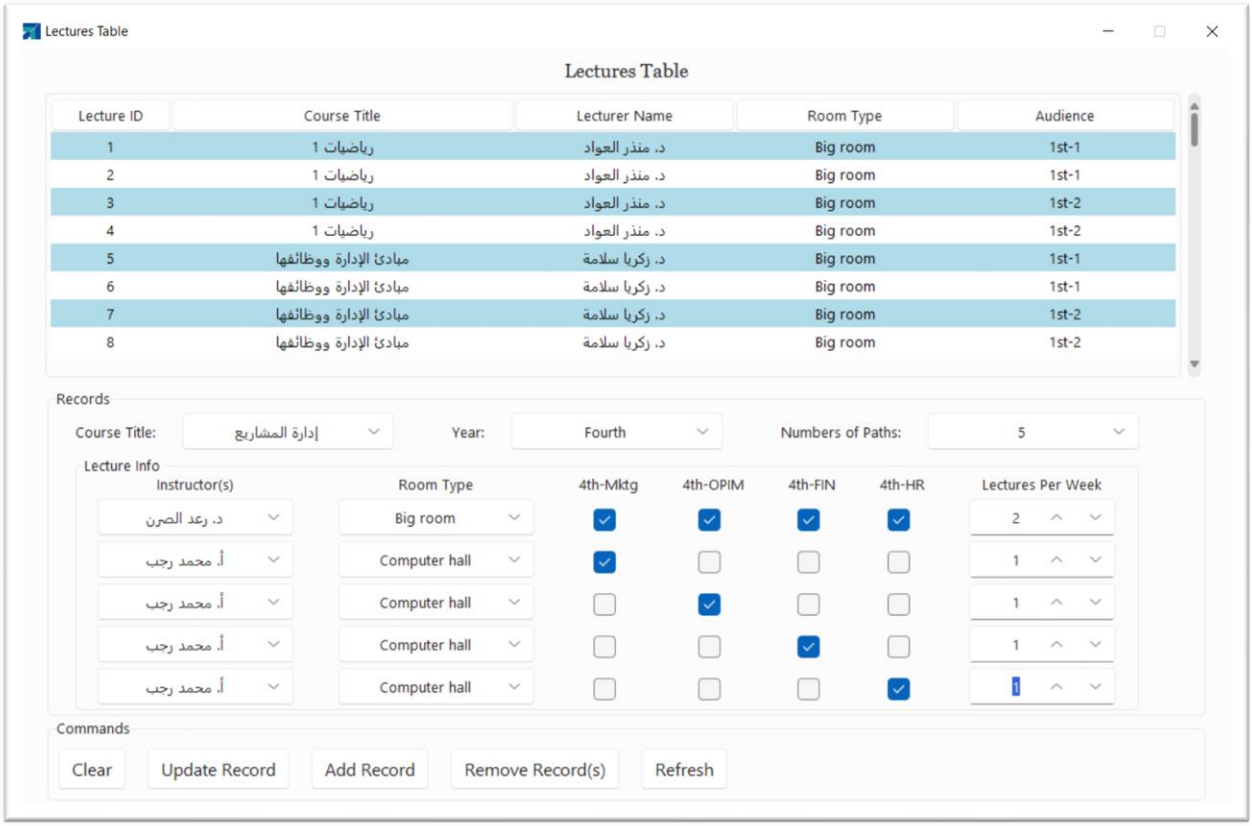
في هذه المرحلة يكون المستخدم قد أدخل جميع البيانات المطلوبة لإدخال المحاضرات الدراسية إلى قاعدة المعطيات، وهي أكثر المراحل استهلاكاً للوقت إلا أنه قد تم تصميم واجهة استخدام مرنة تسهل هذه العملية بشكل كبير وتقلل من الوقت المستغرق في إتمام العملية. تظهر الصورة التالية كيف تبدو واجهة إدخال المحاضرات Lectures (الشكل 35).

الشكل (35) واجهة إدخال المحاضرات 1

نلاحظ أن جدول المحاضرات يتكون من مجموعة من المفاتيح الثانوية بالإضافة إلى أن لكل محاضرة مجموعة من الفئات اللاتي يشكلون مجموعة الحضور (علاقة كثير لكثير كما تم التوضيح في القسم الخاص بتصميم قاعدة المعطيات).

يقوم المستخدم أولاً بتحديد أسم المقرر والسنة الدراسية، بناء على السنة الدراسية تظهر لديه مجموعة الفئات التي تنتمي لهذه السنة ويمكنه أن يختار جميعها أو جزء منها، وبالتالي تتاح له إمكانية إنشاء محاضرات مشتركة بين جميع الفئات أو محاضرات اختصاص مخصصة لفئات محددة. بناء على المتطلبات وخصائص المقرر يختار المستخدم عدد المسارات الذي يريده.

من أجل توضيح الفكرة تم عرض هذا المثال الشامل لنوعي المحاضرات المذكور أعلاه في الشكل (36). حيث تم اختيار مقرر إدارة المشاريع وتم اختيار السنة الرابعة (نلاحظ أنه لا يمكن للمستخدم إلا اختيار الفئات التي تنتمي للسنة الرابعة)، كما تم إنشاء 5 مسارات مختلفة بناء على رغبة المستخدم. في كل مسار من هذا المسارات على المستخدم أن يحدد أسم المدرس ونوع القاعة المطلوب ومجموعة الفئات التي تشكل الحضور وعدد المحاضرات الأسبوعية. في مثالنا الحالي، المسار الأول يمثل محاضرتين مشتركتين بين جميع اختصاصات السنة الرابعة وتحديد نوع القاعة لكي يكون كبيرا ويستوعب عدد الطلاب الكبير. أما في المسارات الأربعة التالية تم الاكتفاء بقاعات حاسوب ومحاضرة واحدة أسبوعية لكل اختصاص، حيث تشكل هذه المحاضرات محاضرات الجانب العملي من المقرر.



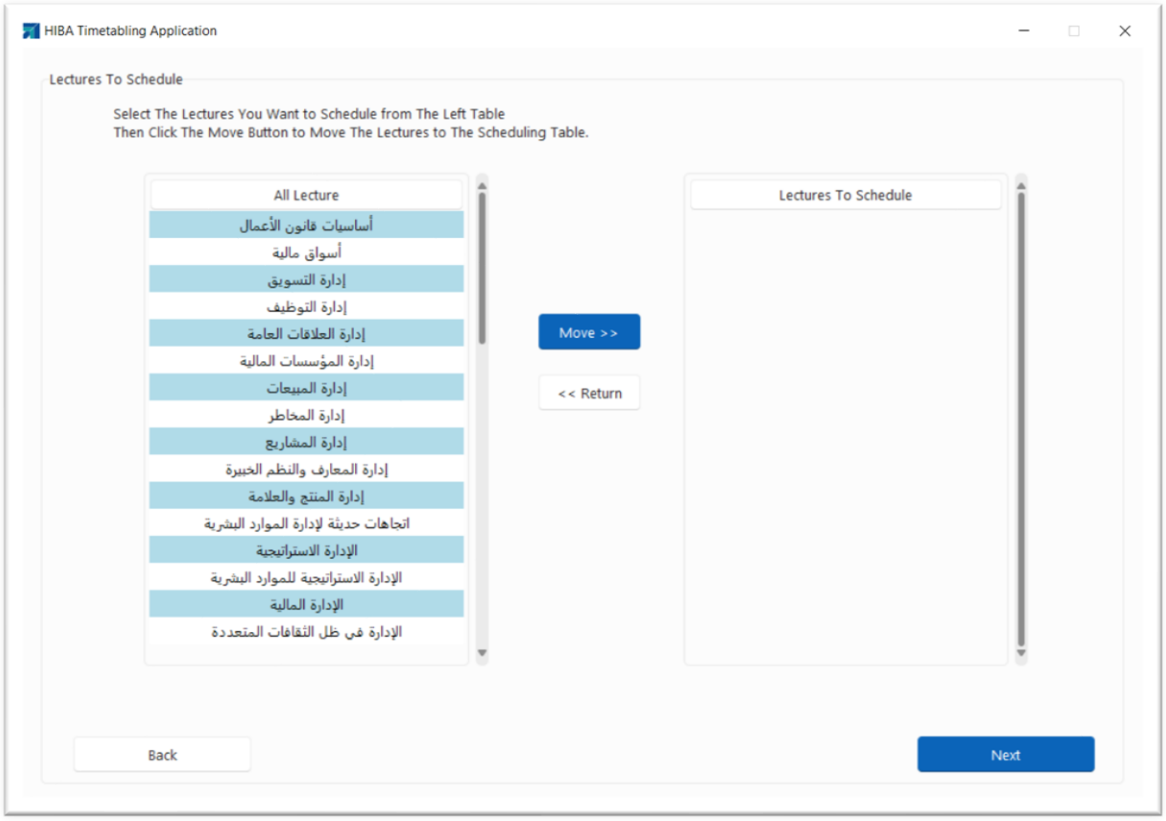
الشكل (36) واجهة إدخال المحاضرات 2

بعد الانتهاء من إدخال البيانات السابقة يمكن للمستخدم الانتقال إلى مرحلة إنشاء حل وذلك من خلال واجهة الاستخدام Start. عند الضغط على زر Start يظهر للمستخدم واجهة تتيح له إمكانية تحديد أيام الدوام الأسبوعية وأيام العطل وتحديد عدد الفترات اليومية وفترة كل محاضرة ووقت بداية المحاضرة الأولى (الشكل 37).

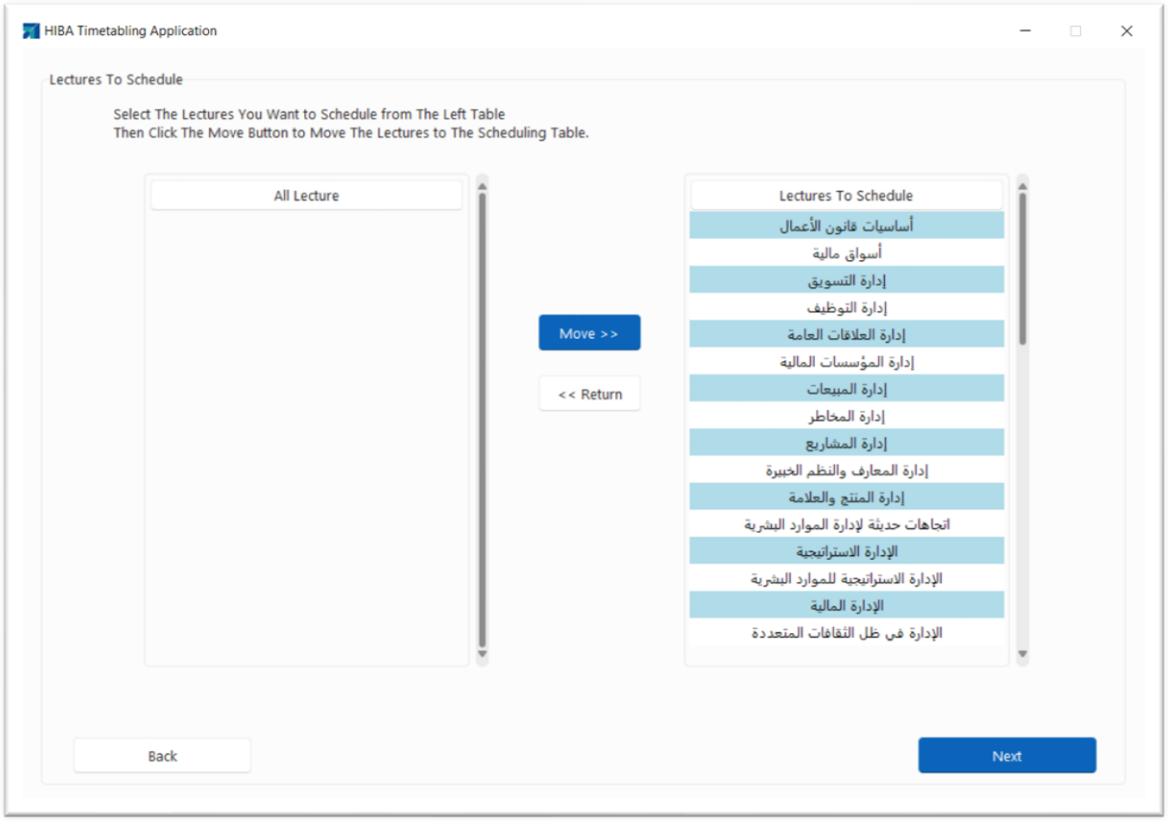
الشكل (37) واجهة تحديد الأوقات

في الواجهة أعلاه قام المستخدم بتحديد أيام الدوام من الأحد الى الخميس كما حدد موعد بداية أول محاضرة في تمام الساعة 8:30 وحدد مدة المحاضرة ساعة ونصف وعدد الفترات اليومية بـ 4.

عند الضغط على زر Next يظهر للمستخدم واجهة لاختيار مجموعة المقررات التي يُراد جدولة محاضراتها (الشكل 38). حيث يمكن للمستخدم جدولة كل المحاضرة أو مجموعة جزئية منها. في الحالة التالية قام المستخدم باختيار جميع المقررات (الشكل 39).

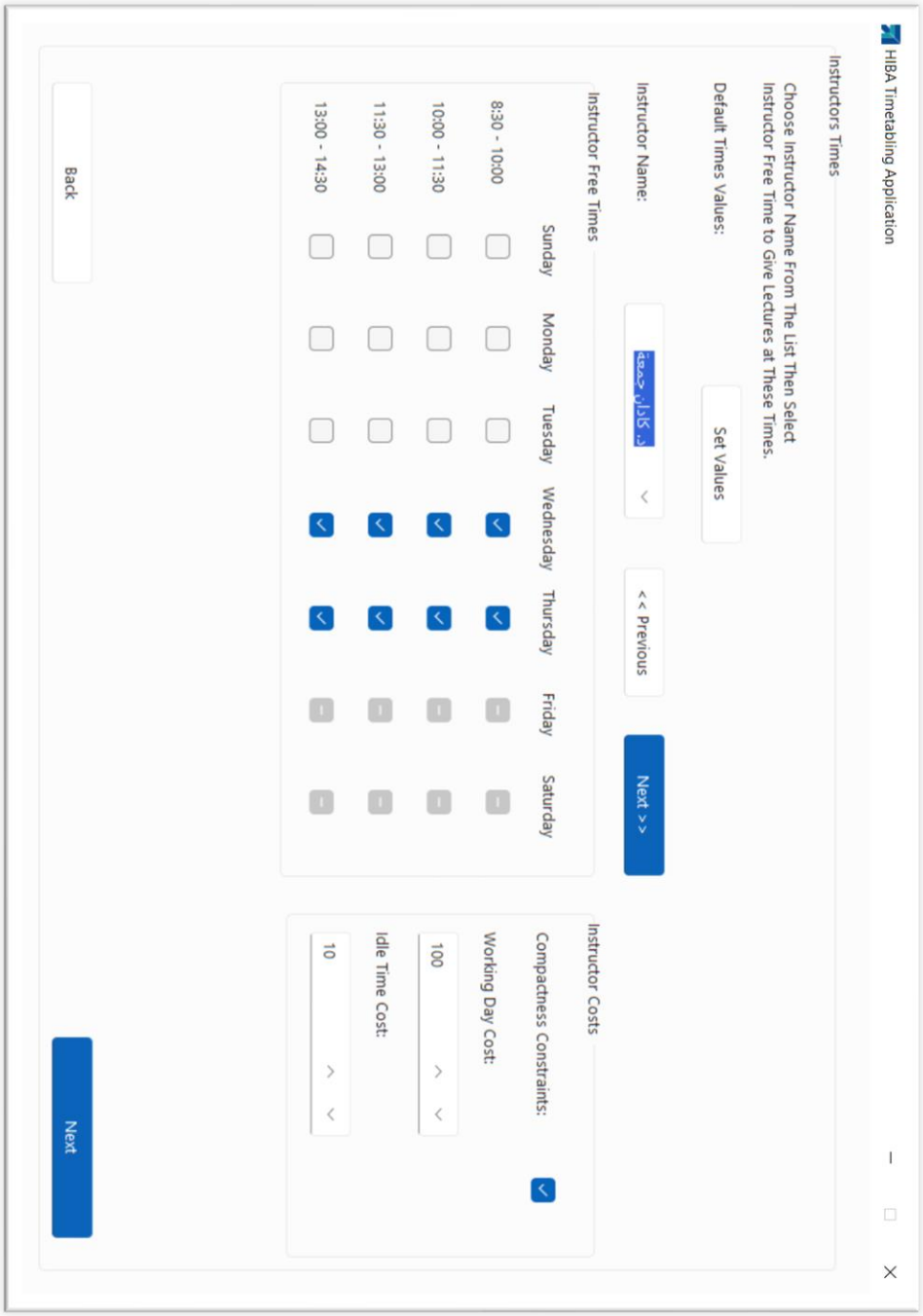


الشكل (38) واجهة اختيار المحاضرات 1



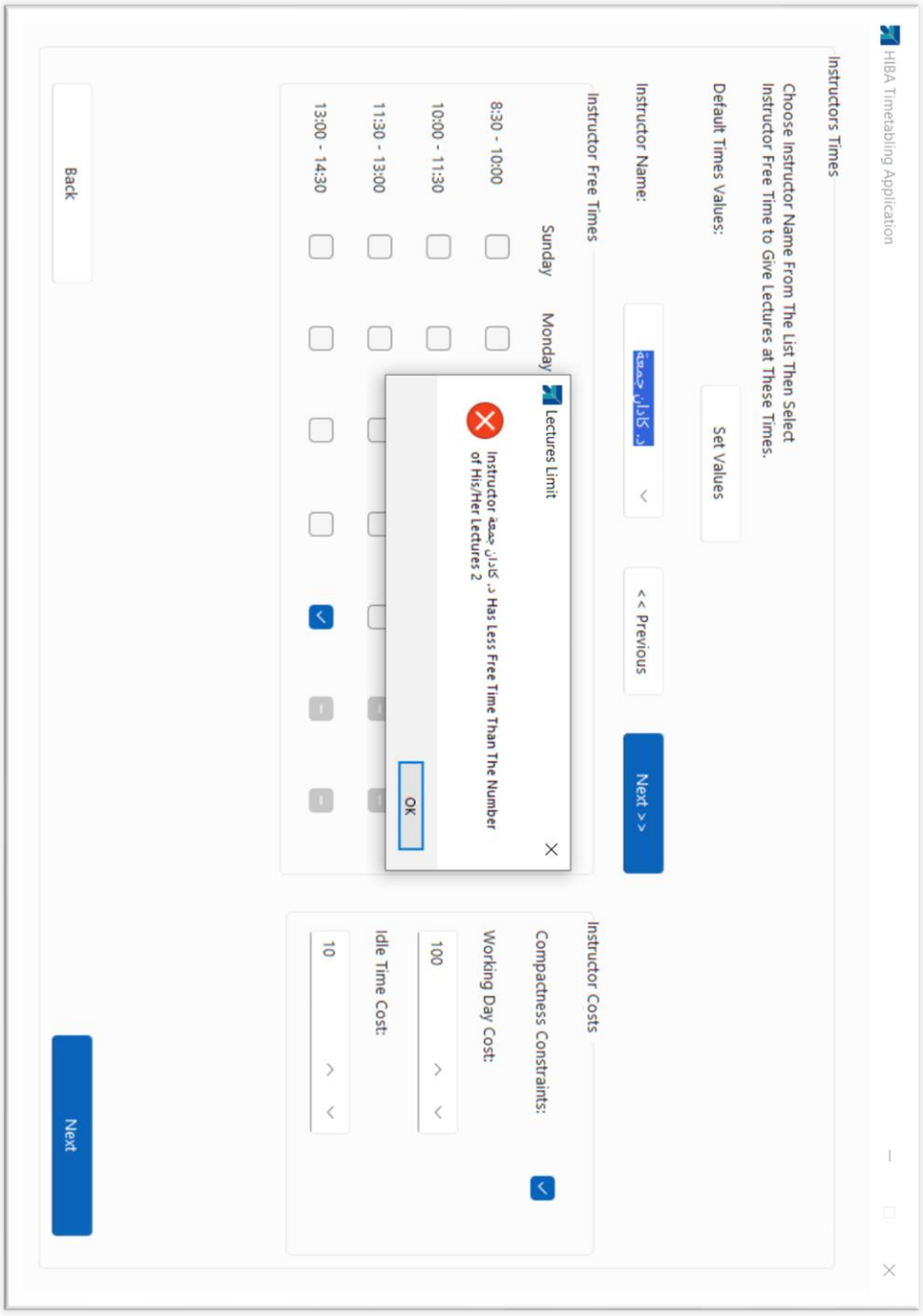
الشكل (39) واجهة اختيار المحاضرات 2

بعد ذلك يتنقل المستخدم إلى الواجهة الخاصة بمتطلبات المدرسين حيث يحدد الفترات المتاحة لكل مدرس ويحدد تكلفة يوم عمل المدرس وتكلفة فترة التعطل الخاصة به، وبالتالي يمكن للمستخدم تفضيل مجموعة من المدرسين عن غيرهم في مسألة ضغط الجدول الزمني وتظهر الحاجة لذلك حينما يكون هناك مدرسين يأتون من خارج المعهد لإعطاء المحاضرات وبالتالي يتم تفضيلهم عن المدرسين المتواجدين في المعهد بشكل دائم. كما يمكن إيقاف قيود ضغط الجدول الزمني أساساً حينما لا يكون هناك حاجة لذلك من قبل المدرس. في المثال التالي تم تحديد الفترات المتاحة بالنسبة لأحد المدرسين وتحديد تكاليف يوم العمل وتكاليف وقت التعطل ومن ثم الضغط على Next (الشكل 40).



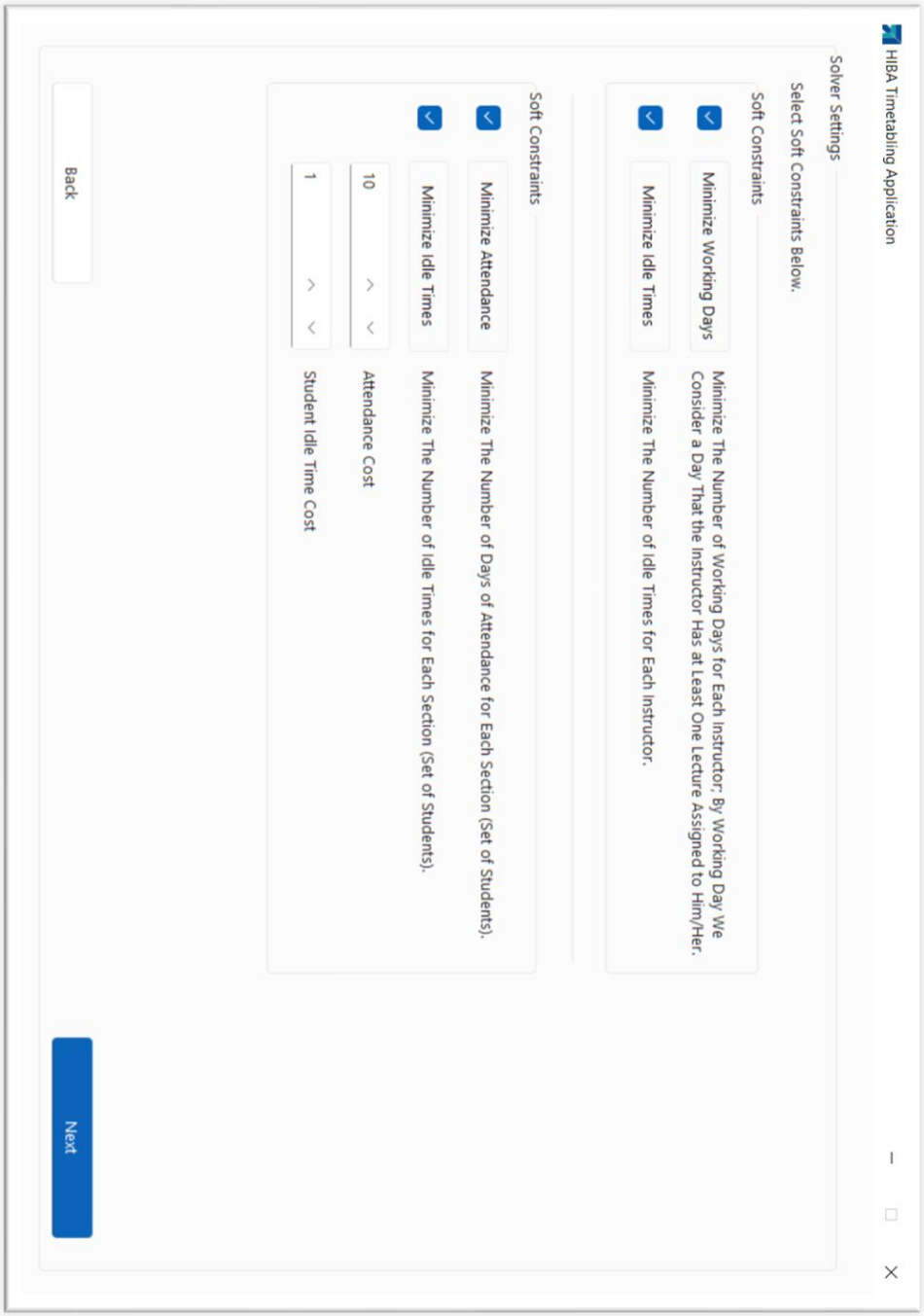
الشكل (40) واجهة خصائص المدرسين 1

في حال قام المستخدم بتحديد أوقات متاحة لمدرس ما أقل من عدد محاضرات هذا المدرس يعطي النظام رسالة خطأ ويمنع المستخدم من الانتقال إلى المرحلة التالية كما هو موضح في الشكل (41).



الشكل (41) واجهة خصائص المدرسين 2

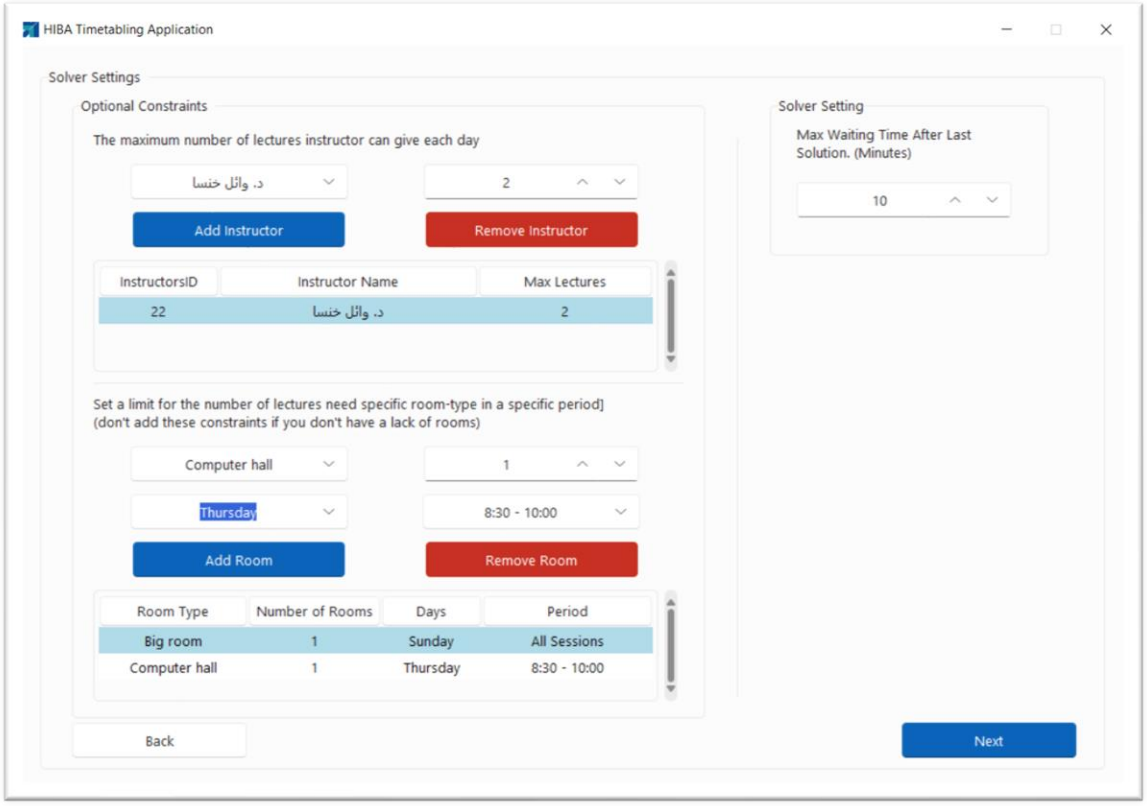
بعد ذلك تظهر واجهة خاصة بتحديد القيود المرنة المراد إضافتها على النموذج الرياضي، وتتألف من أربع خيارات. الخيار الأول مسؤول عن تشغيل القيود التي تقوم بتقليل عدد أيام عمل المدرسين إلى أصغر حد ممكن. الخيار الثاني مسؤول عن تشغيل القيود التي تقوم بتقليل عدد أوقات تعطل المدرسين. الخيار الثالث يقلل عدد حضور فئات الطلاب إلى المعهد، أما الخيار الرابع فيقلل عدد أوقات تعطل فئات الطلاب. بالإضافة إلى تحديد تكلفة حضور الطلاب إلى المعهد وتكلفة وقت التعطل الخاصة بهم. إلا أنه في هذه الحالة – وعلى خلاف واجهة إعدادات المدرسين – يتم اختيار تكاليف موحدة لكل الفئات وذلك لعدم وجود حاجة لتفضيل فئة عن فئة أخرى في مسألة ضغط الجدول الزمني.



الشكل (42) واجهة اختيار القيود المرنة

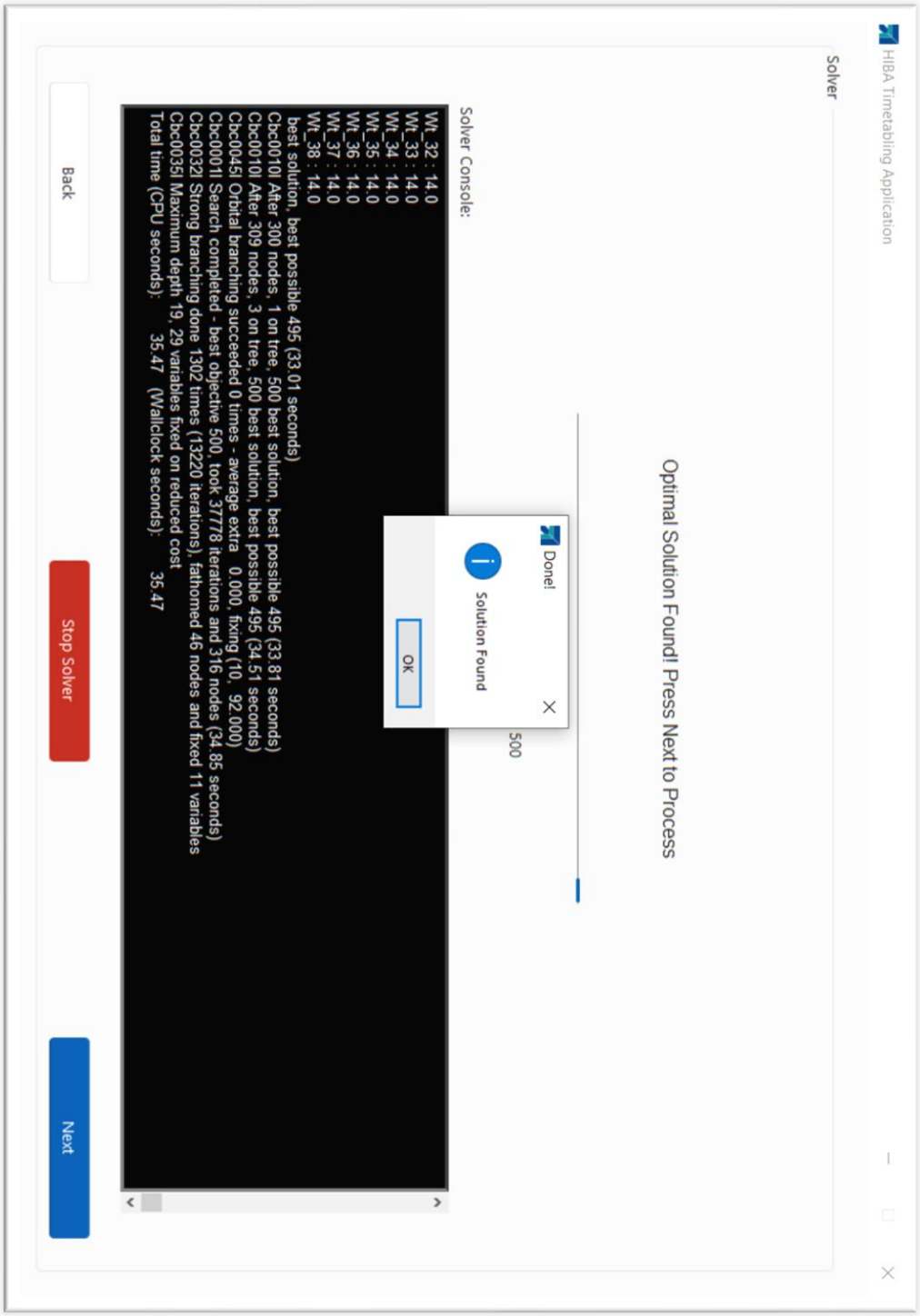
عند الضغط على زر Next تظهر واجهة خاصة بالقيود الاختيارية وهما نوعان. وظيفة النوع الأول من القيود هي تحديد الحد الأقصى لعدد المحاضرات اليومية التي يمكن لمدرس ما إعطاؤها. في المثال الموضح في الشكل أدناه تم تحديد هذا الحد بمحاضرتين فقط لأحد المدرسين، وبالتالي حتى وإن كان هذا المدرس متاح في كل أوقات الأسبوع إلا أن النظام لن يكلفه بأكثر من محاضرتين في كل يوم. أما بالنسبة للنوع الثاني من هذه القيود فإنها تخص القاعات التدريسية، جدير بالذكر أنه في حال كانت المؤسسة لا تواجه مشكلة في عدد القاعات (لا يوجد نقص في القاعات) فلا حاجة لاستخدام هذه القيود، أما إذا كان هناك محدودية في عدد القاعات من نوع ما فمن الأفضل استخدام هذا النوع من القيود. في المثال الموضح في الصورة التالية تم إضافة قيدين من هذا النوع. في القيد الأول تم تحديد عدد القاعات المتاحة من النوع Big room بالعدد 1 وتم تطبيق هذا القيد على جميع فترات يوم الأحد، وبالتالي سيضمن هذا القيد ألا يتم تعيين أكثر من محاضرة واحدة من المحاضرات التي تحتاج قاعات تدريسية كبيرة في كل فترة زمنية يوم الأحد. أما القيد الثاني فتم تحديد عدد صالات الحاسوب المتاحة بقاعة واحدة وذلك فقط في يوم الخميس في الفترة الزمنية (8:30 – 10:00).

يظهر في نفس الواجهة Max waiting time after last solution وهو خيار يتيح للمستخدم تحديد عدد الدقائق التي على محرك الحل التوقف بعدها في حال لم يستطع تحسين الحل. فعلى سبيل المثال إذا تم تحديد 10 دقائق فمحرك الحل سيتوقف بشكل تلقائي في حال وصل إلى حل مقبول ما ولم يجد أفضل من هذا الحل خلال فترة 10 دقائق.



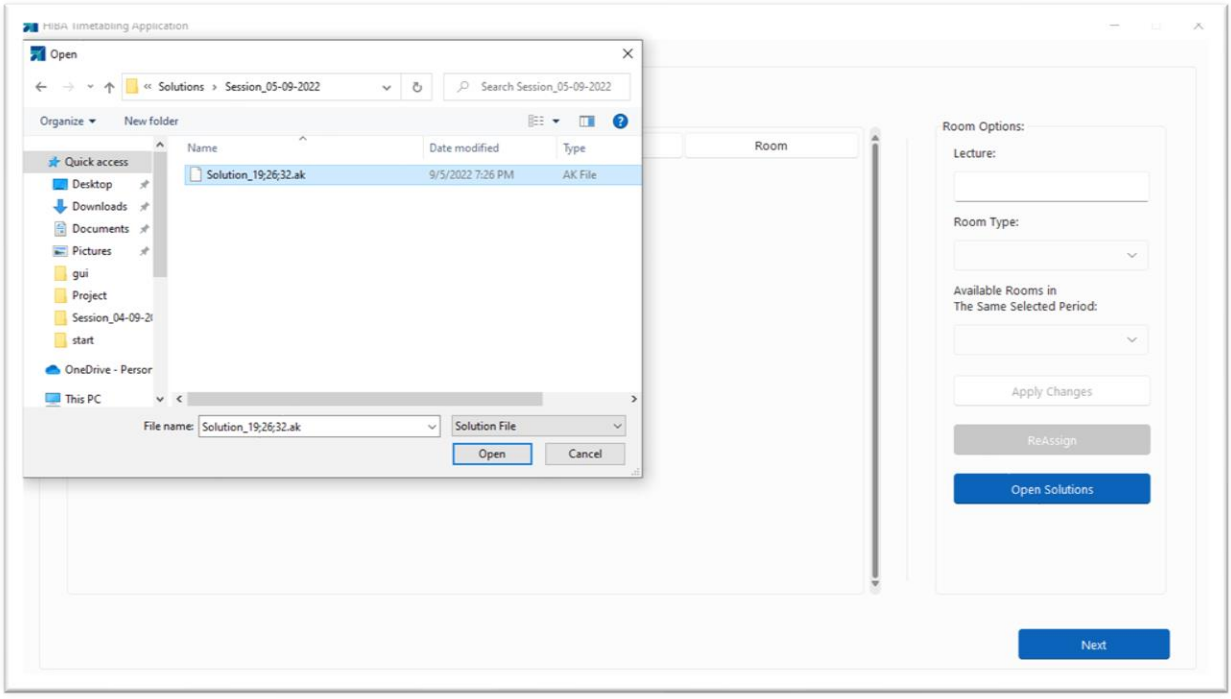
الشكل (43) واجهة إضافة قيود اختيارية 1

عند الضغط على Next يقوم النظام ببناء النموذج الرياضي بناء على المتطلبات التي حددها المستخدم ويستدعي محرك الحل CBC Solver لحل النموذج. بعد الانتهاء من حل النموذج يتم تشغيل خوارزمية تعيين القاعات. عندما ينتهي النظام من حل النموذج وتعيين القاعات يتم حفظ الحل في ملف خارجي لإتاحة إمكانية استخدامه والتعديل عليه في وقت لاحق.



الشكل (44) واجهة محرك الحل Solver

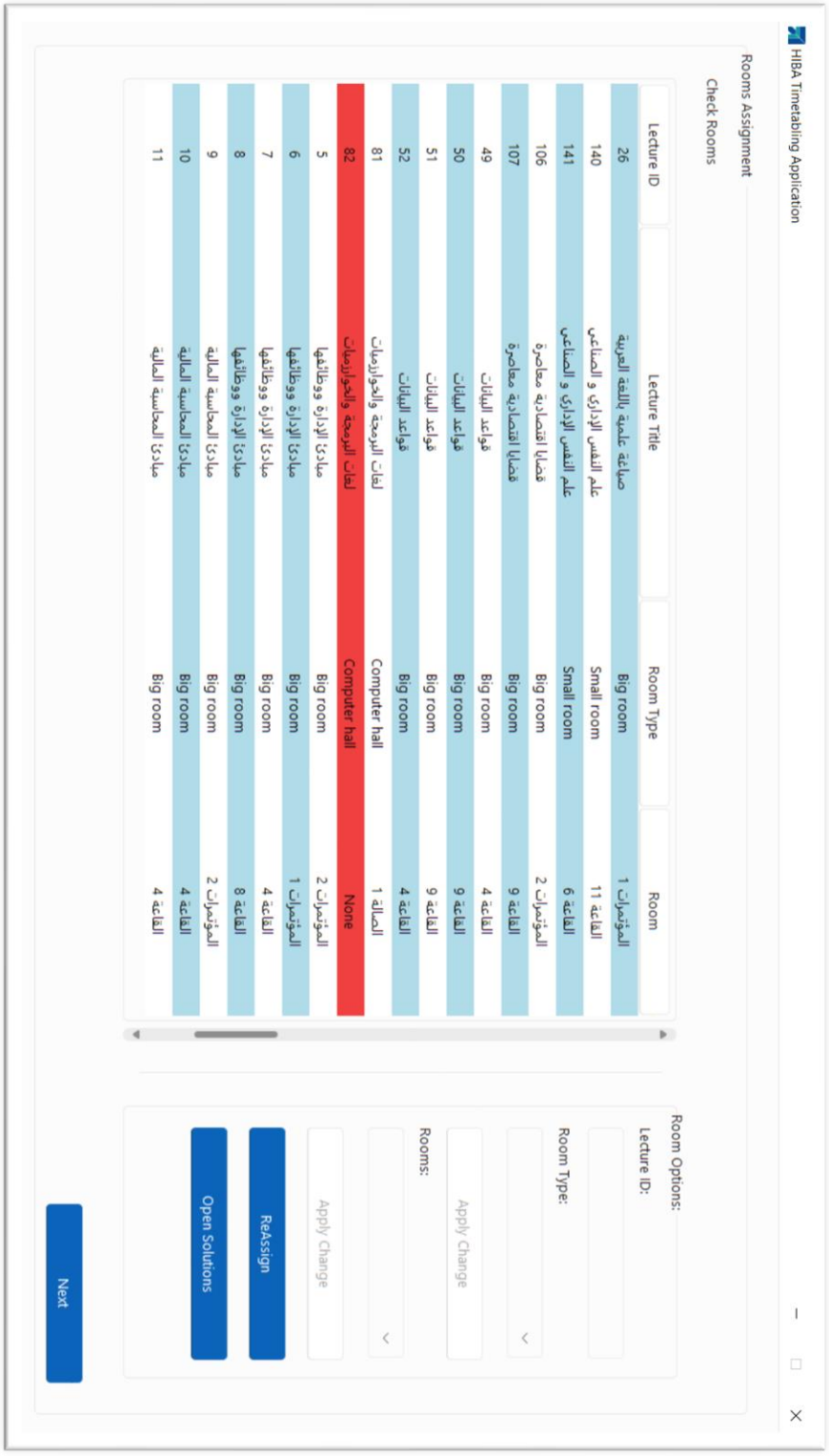
يمكن الانتقال إلى واجهة عرض الجداول إما بالضغط على زر Next أو من خلال الواجهة الرئيسية والضغط على Solutions viewer. يُطلب من المستخدم اختيار الحل الذي يريد عرضه أو التعديل عليه، والذي يقوم النظام بتسميته بناء على تاريخ ووقت إعداد هذا الحل.



الشكل (45) واجهة فتح حل سابق

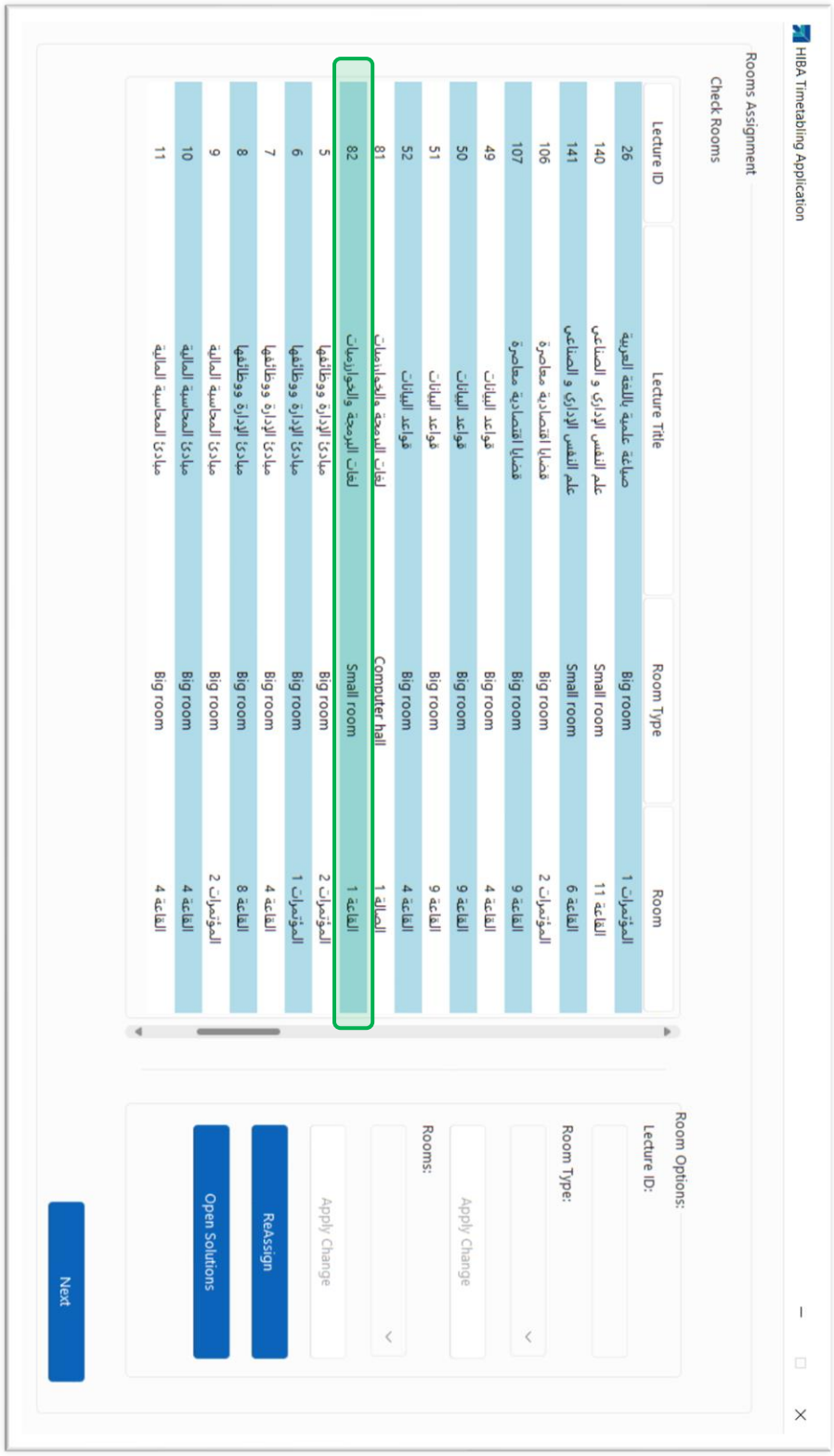
يظهر الحل أولاً على شكل قائمة تحتوي على مجموعة المحاضرات التي تمت جدولتها مع نوع القاعة المطلوب بالنسبة لكل محاضرة بالإضافة للقاعة التدريسية التي تم تعيينها لهذه المحاضرة. في هذه الواجهة يمكن للمستخدم التعديل على نوع القاعة المطلوب بالنسبة لمحاضرة ما أو تعديل القاعة التدريسية بشكل يدوي، وذلك من خلال اختيار النوع المطلوب أو القاعة المطلوبة ومن ثم الضغط على زر Apply change حيث يتم تثبيت التعديل الذي تم إجراؤه. جدير بالذكر أنه من الممكن أن يكون هناك نقص في القاعات التدريسية وذلك في حال لم يتم استخدام القيود الخاصة بتحديد عدد القاعات التدريسية بشكل صحيح، وفي هذه الحالة يُسند النظام قيمة فارغة None لهذه المحاضرة بدلا من تعيين قاعة تدريسية لها، بالإضافة إلى أنه يُظهر رسالة خطأ.

تُظهر الصورة التالي وجود نقص في صالات الحاسوب نتج عن ذلك محاضرة دون قاعة تدريسية، في هذه الحالة يمكن للمستخدم التعديل على نوع القاعة المطلوب بالنسبة لهذه المحاضرة كما هو موضح في الصورة التالية أو إجراء أي تغيير اخر من شأنه أن يحل المشكلة.



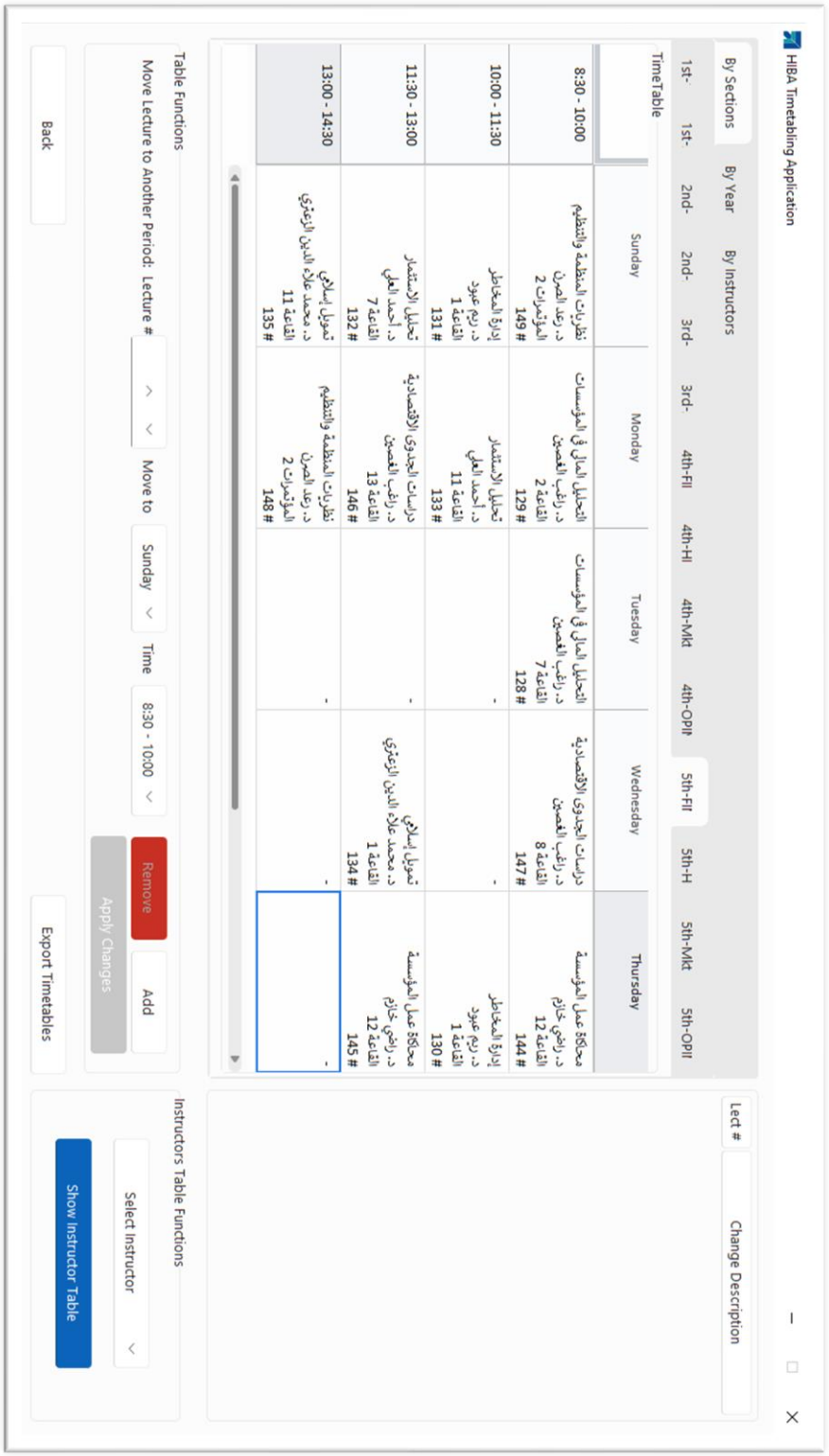
الشكل (46) واجهة قاعات المحاضرات 1

قام المستخدم في هذا المثال بتغيير نوع القاعة المطلوب بالنسبة لهذه المحاضرة والضغط على Apply change لتثبيت التعديل.



الشكل (47) واجهة قاعات المحاضرات 2

بالضغط على زر Next يتم عرض الحل على شكل جداول. كما هو موضح، يتم إعداد جداول مصنفة لكل من فئات الطلاب أو للسنوات الدراسية. يُظهر الجدول التالي جدول المحاضرات الأسبوعية للسنة الخامسة اختصاص الإدارة المالية والمصرفية.



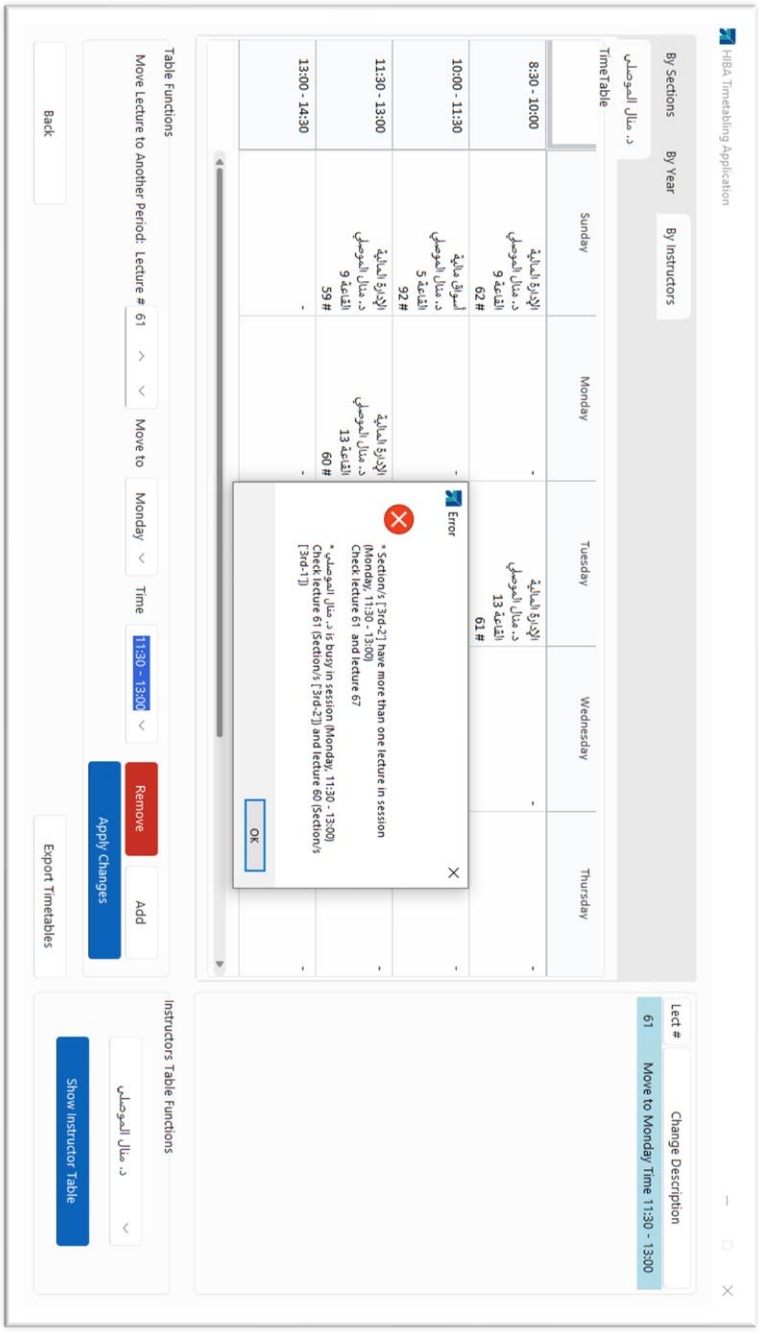
الشكل (48) واجهة الجداول الأسبوعية 1

كما يمكن إعداد الجداول لكل مدرس من خلال تحديد أسم المدرس والضغط على زر Show instructor table.

الشكل (49) واجهة الجداول الأسبوعية 2

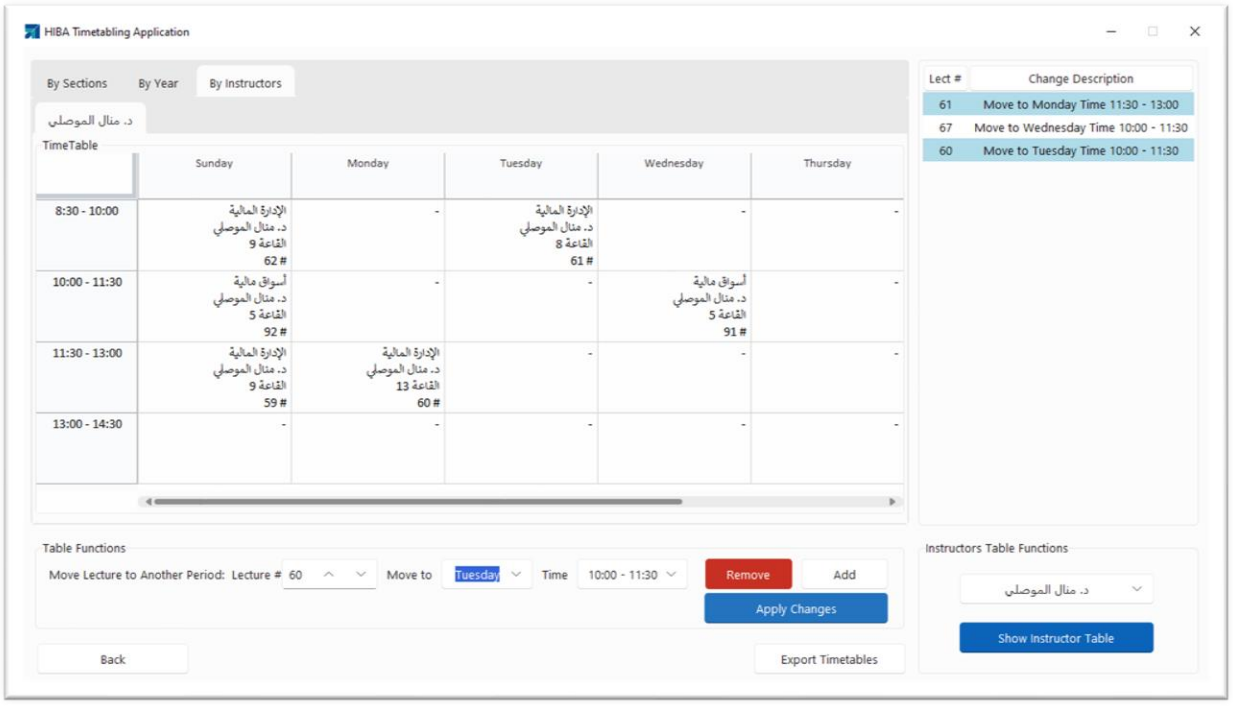
يمكن تغيير وقت أي محاضرة من خلال تحديد رقم هذه المحاضرة واليوم والفترة المراد نقلها إليها، ومن ثم الضغط على Add، بعد إضافة مجموعة التعديلات المراد إجراؤها يقوم المستخدم بالضغط على Apply changes لإجراء التعديلات المطلوبة. يقوم النظام بالتأكد من إمكانية إجراء هذه التعديلات قبل إجرائها ويرجع رسالة خطأ في حال وجود أية تناقض.

لنفترض أن المستخدم في المثال السابق قام بنقل المحاضرة 61 إلى يوم الاثنين الساعة 11:30. بعد الضغط على Add ومن ثم Apply Changes ظهر للمستخدم رسالة خطأ تفيد بأن هناك تعارض بين المحاضرتين 61 و67 لانهما يشتركا بنفس الفئة وتعارض بين 61 و60 لأنهما يشتركا بنفس المدرس.



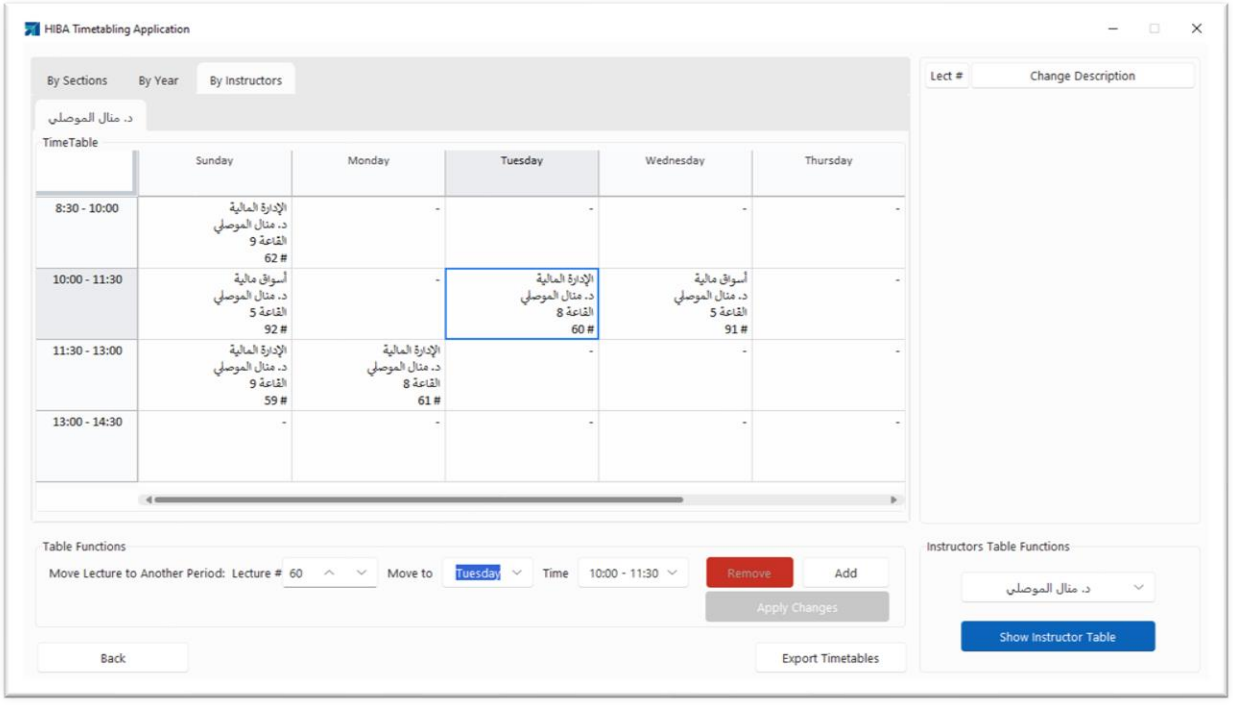
الشكل (50) واجهة الجداول الأسبوعية 3

من أجل حل هذه المشكلة قام المستخدم بإضافة مجموعة من التعديلات حيث نقل المحاضرة 67 إلى يوم الأربعاء الساعة 10:00 ونقل المحاضرة 60 إلى يوم الثلاثاء الساعة 10:00.



الشكل (51) واجهة الجداول الأسبوعية 4

بعد الضغط علة زر Apply changes تم نقل المحاضرات بنجاح.



الشكل (52) واجهة الجداول الأسبوعية 5

2.2.4.4 الاختبارات

يتناول هذا القسم مجموعة من الاختبارات بتنويعات مختلفة، تختلف بحجم المدخلات والقيود المرغوب بتحققها، بهدف قياس جودة النظام وكفاءته في الواقع العملي. حيث تم تصنيف الاختبارات في ثلاثة أقسام رئيسية حسب عدد المحاضرات المراد جدولتها، وضمن كل قسم تم إجراء مجموعة من الاختبارات تختلف بنوعية القيود وبالتالي تختلف من حيث عدد القيود والمتغيرات في النموذج الرياضي.

1.2.2.4.4 تصنيف الاختبارات

يبين الجدول التالي الأصناف الثلاثة للاختبارات، حيث يبدأ بحجم البيانات الأكبر في الصنف A ثم يبدأ بالتناقص من حيث عدد المحاضرات والمدرسين والفئات. تم تثبيت عدد أيام الدوام بـ 5 وكذلك عدد الفترات لأن هذه هي الحالة الطبيعية في المعهد العالي لإدارة الأعمال.

تصنيفات الاختبارات	عدد المقررات	عدد المحاضرات	عدد المدرسين	عدد الفئات	عدد الأيام	عدد الفترات اليومية	عدد الفترات الأسبوعية
A	53	145	39	14	5	4	20
B	19	74	18	6	5	4	20
C	7	26	7	2	5	4	20

الجدول (11) تصنيف الاختبارات

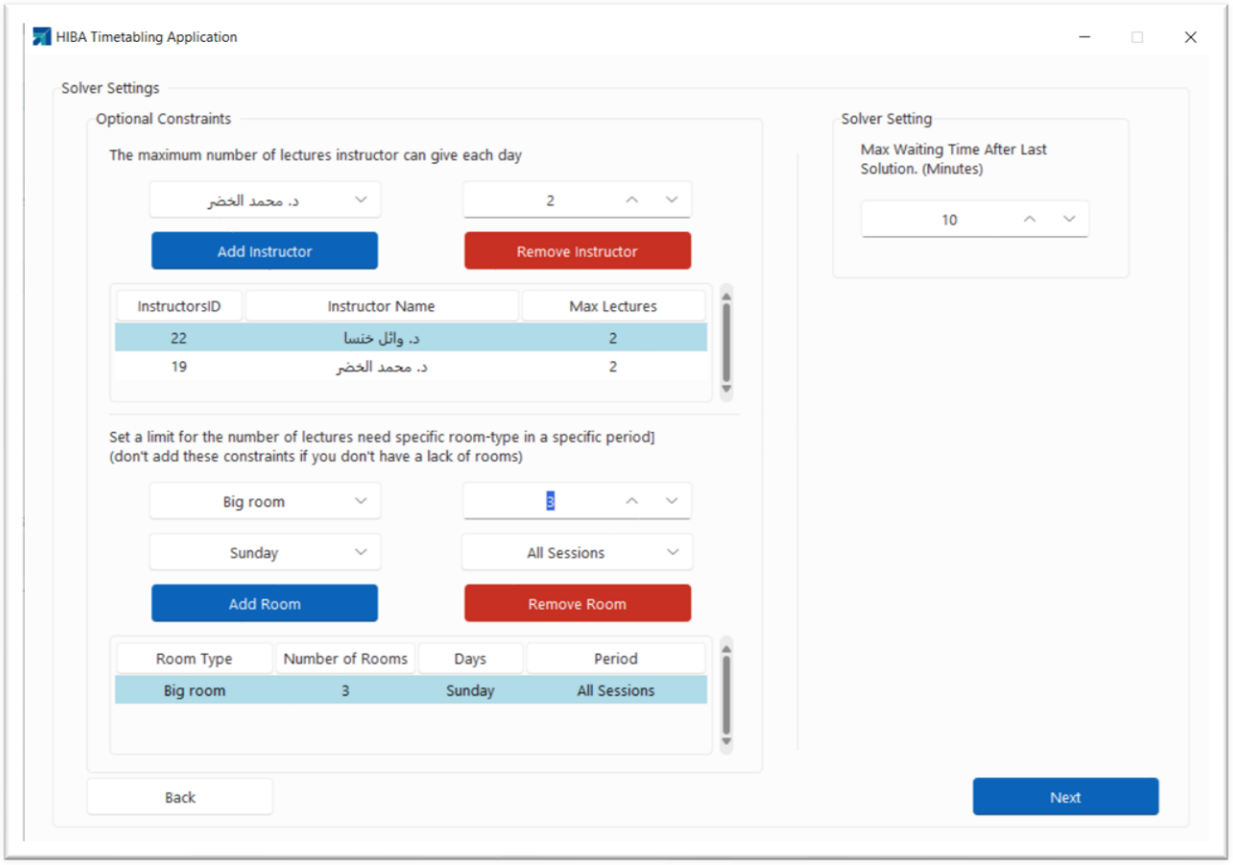
يتضمن الجدول التالي مدخلات الاختبارات التي تنتمي لمجموعة الاختبارات A. تختلف هذه الاختبارات عن بعضها البعض بعدد بالقيود المفعلة. تم تحديد التكاليف على النحو التالي، تكلفة يوم عمل المدرس تساوي 100، تكلفة ففترة تعطل المدرس تساوي 10، تكلفة يوم دوام الفئة تساوي 10، تكلفة فترة تعطل الفئة تساوي 1.

TSwift Test		القيود المرنة				القيود الاختيارية	
رقم التجربة	عدد المحاضرات المجدولة	قيد تخفيض أيام عمل المحاضرون	قيد تخفيض أوقات تعطل المحاضرون	قيد تخفيض أيام حضور الطلاب	قيد تخفيض أوقات تعطل الطلاب	قيد عدد المحاضرات الأعظمي للمحاضر	قيد القاعات
A1	145	0	0	0	0	0	0
A2	145	39	39	0	0	0	0
A3	145	39	18	0	0	0	0
A4	145	18	39	0	0	0	0
A5	145	18	18	0	0	0	0
A6	145	18	18	مفعل	0	0	0
A7	145	18	18	مفعل	0	2	1
A8	145	18	18	مفعل	مفعل	0	0
A9	145	39	39	مفعل	0	0	0
A10	145	39	39	مفعل	مفعل	0	0

الجدول (12) مدخلات مجموعة اختبارات A

في هذه المجموعة من الاختبارات تم البدء بالحالة الأكثر سهولة A1 وهي التي لم يتم تفعيل أية قيود مرنة فيها، ففي هذه الحالة لم نطلب من النظام تفعيل قيود ضغط الجدول الزمني للمدرسين أو الفئات ولم يتم استخدام أي من القيود الاختيارية. في الاختبار A2 تم تفعيل قيود ضغط الجدول الزمني لجميع المدرسين دون تفعيل قيود الضغط للطلاب. في الاختبار A3 تم تفعيل قيود تخفيض أيام العمل لكل المدرسين بينما لم يتم تفعيل قيود تخفيض أوقات التعطل إلا للمدرسين الخارجيين. أما الاختبار A4 فعلى عكس الاختبار A3 تم تفعيل قيود تخفيض أوقات التعطل لكل المدرسين بينما لم يتم تفعيل قيود تخفيض أيام العمل إلا للمدرسين الخارجيين. في الاختبار A5 تم تفعيل قيود ضغط الجدول الزمني للمدرسين الخارجيين فقط.

يعتبر الاختبار A6 هو الأكثر مطابقةً للحالة الطبيعية، حيث تهتم إدارة المعهد بضغط الجدول الزمني للمدرسين الخارجيين فقط، أما المدرسين الداخليين فلا يطالبون بضغط جدولهم وذلك لوجودهم الدائم في حرم المعهد. أما بالنسبة للطلاب فهم يفضلون تقليل أيام دوامهم وذلك بسبب صعوبة الوصول إلى المعهد في هذه الأيام. ونظراً إلى أن الاختبار هو الأكثر مطابقة للحالة الطبيعية تم إجراء اختبار آخر بنفس القيود مضافاً إليها مجموعة من القيود الاختيارية وذلك في الاختبار A7، توضح الواجهة التالية تفاصيل القيود الاختيارية.



الشكل (53) واجهة إضافة قيود اختيارية 2

إن الاختبار A8 مشابه للاختبار A6 ولكن تم تفعيل قيود تخفيض أوقات تعطل الطلاب. الاختبار A9 أيضاً مشابه لـ A6 ولكن بدلاً من تفعيل قيود ضغط الجدول الزمني للمدرسين الخارجيين تم تفعيل هذه القيود لجميع المدرسين. الاختبار A10 مشابه لـ A8 ولكن بدلاً من تفعيل قيود ضغط الجدول الزمني للمدرسين الخارجيين تم تفعيل هذه القيود لجميع المدرسين.

يتضمن الجدول التالي مجموعة الاختبارات B وC.

TSwift Test		القيود المرنة				القيود الاختيارية	
رقم التجربة	عدد المحاضرات المجدولة	قيد تخفيض أيام عمل المحاضرون	قيد تخفيض أوقات تعطل المحاضرون	قيد تخفيض أيام حضور الطلاب	قيد تخفيض أوقات تعطل الطلاب	قيد عدد المحاضرات الأعظمي للمحاضر	قيد القاعات
B1	74	0	0	0	0	0	0
B2	74	18	18	0	0	0	0
B3	74	18	7	0	0	0	0
B4	74	7	18	0	0	0	0
B5	74	7	7	0	0	0	0
B6	74	7	7	مفعل	0	0	0
B7	74	7	7	مفعل	0	2	1
B8	74	7	7	مفعل	مفعل	0	0
B9	74	18	18	مفعل	0	0	0
B10	74	18	18	مفعل	مفعل	0	0
C1	26	0	0	0	0	0	0
C2	26	7	7	0	0	0	0
C3	26	3	7	0	0	0	0
C4	26	7	3	0	0	0	0
C5	26	3	3	0	0	0	0
C6	26	3	3	مفعل	0	0	0
C7	26	3	3	مفعل	0	2	1
C8	26	3	3	مفعل	مفعل	0	0
C9	26	7	7	مفعل	0	0	0
C10	26	7	7	مفعل	مفعل	0	0

الجدول (13) مدخلات مجموعة اختبارات B وC

2.2.2.4.4 نتائج الاختبارات

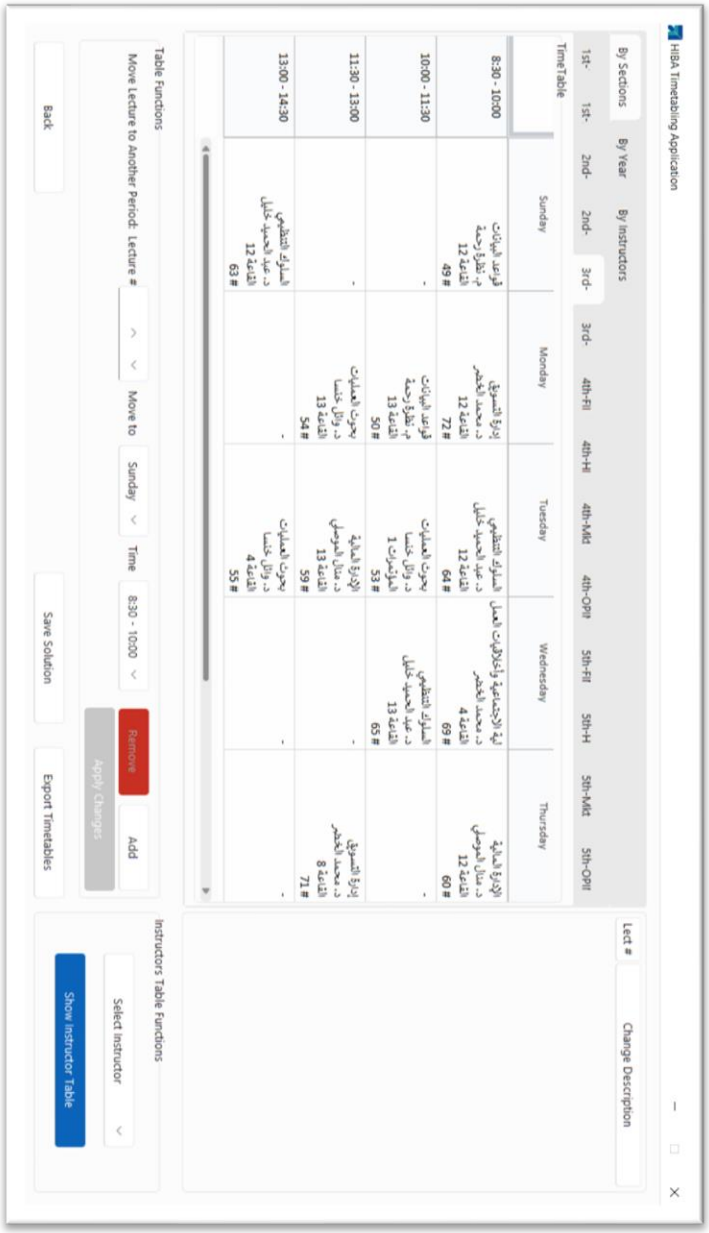
يتناول هذا القسم نتائج الاختبارات التي تم توضيحها في القسم السابق. حيث تم إجراء هذه الاختبارات وتقييمها من خلال مجموعة من المعايير، وهي عدد المتغيرات والقيود وعدد العقد التي تم المرور عليها من قبل محرك الحل (Solver). بالإضافة لقيمة دالة الهدف الخاصة بالحل وأفضل حل ممكن تم تحديده من قبل محرك الحل والفرق بينهما (Gap)، وأخيراً الوقت المستغرق لإيجاد الحل.

TSwift Tests		تفاصيل الحل						
رقم التجربة	عدد المحاضرات المجدولة	عدد المتغيرات	عدد القيود	عدد العقد	قيمة دالة الهدف	أفضل حل ممكن	Gap	الوقت المستغرق لإيجاد الحل (Seconds)
A1	145	4060	294717	0	0	0	0	8.6
A2	145	5464	298032	15	4700	4700	0	24.27
A3	145	5464	298032	7	4700	4700	0	22.46
A4	145	5464	298032	2	2000	2000	0	17.17
A5	145	5464	298032	0	2000	2000	0	12.97
A6	145	5562	298424	30028	2500	2490	10	960.62
A7	145	5562	298442	26423	2600	2562.08	37.92	927.69
A8	145	5968	299222	17471	2500	2460	40	939.88
A9	145	5562	298424	11502	5220	5062.5	157.5	1108.05
A10	145	5968	299222	12087	5230	5140	90	2212.4
B1	74	2072	76798	0	0	0	0	2.32
B2	74	2720	78328	0	2100	2100	0	3.72
B3	74	2720	78328	0	2100	2100	0	3.88
B4	74	2720	78328	0	800	800	0	3.34
B5	74	2720	78328	0	800	800	0	3.58
B6	74	2818	78720	0	1020	1020	0	4.38
B7	74	2818	78738	1	1120	1120	0	8.13
B8	74	3224	79518	6	1020	1020	0	6.4
B9	74	2818	78720	52161	2320	2297.5	22.5	905.12
B10	74	3224	79518	43809	2320	2297.5	22.5	907.82
C1	26	728	9530	0	0	0	0	1.07
C2	26	980	10125	0	700	700	0	1.5
C3	26	980	10125	0	700	700	0	1.54
C4	26	980	10125	0	300	300	0	1.31
C5	26	980	10125	0	300	300	0	1.27
C6	26	1078	10517	0	380	380	0	1.83
C7	26	1078	10559	0	480	480	0	1.82
C8	26	1078	10517	0	380	380	0	1.78
C9	26	1078	10517	0	780	780	0	2.08
C10	26	1484	11315	241651	780	771	9	901.81

الجدول (14) نتائج الاختبارات

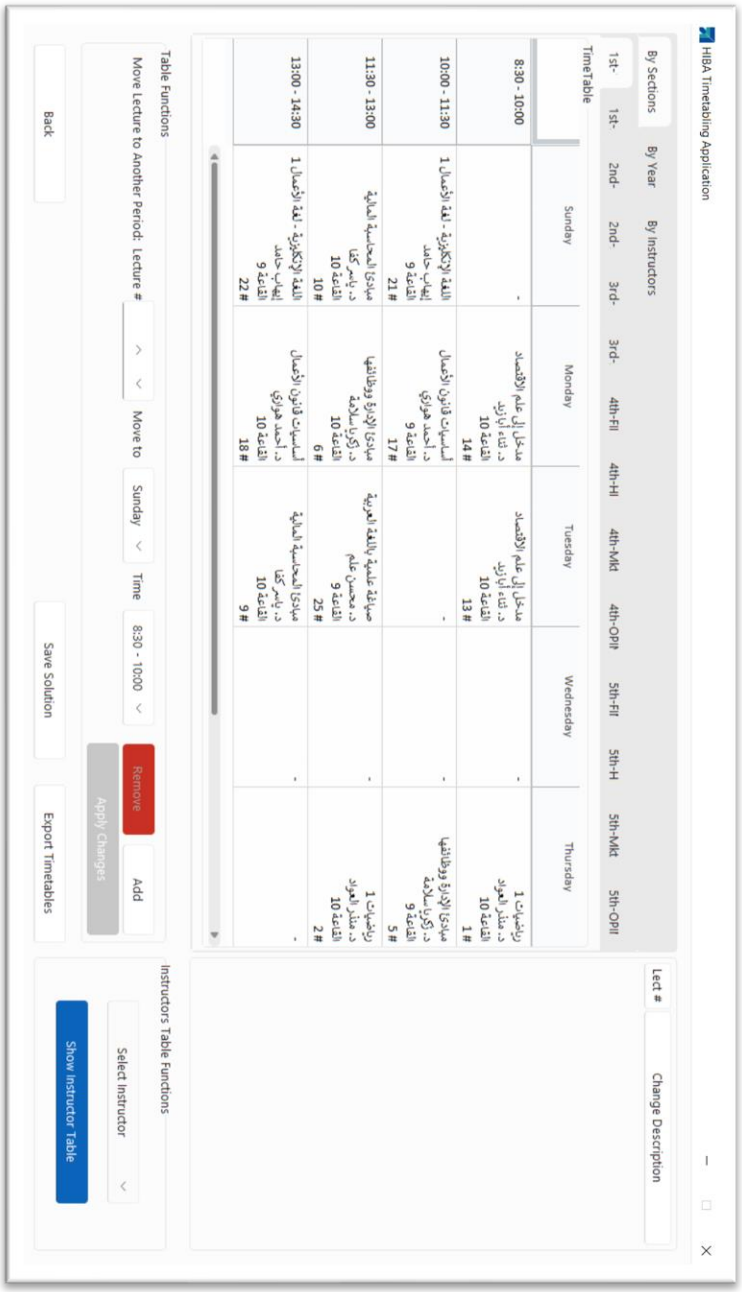
يجب الإشارة إلى أنه في بعض الاختبارات المذكورة، تم الوصول إلى الحل الأمثلي بوقت مبكر إلا أن محرك الحل لم يتأكد من أمثلية هذا الحل – أي لم تتطابق قيمة دالة الهدف مع أفضل حل ممكن و هي مشكلة شائعة عند حل مسائل البرمجة الطبيعية – كما هو الحال في التجارب A6,A7,A8, B9, B10, C10 حيث تبين بعد عرض هذه الحلول انه لم يكن هنالك أية اختراق للقيود. فمثلا في الاختبار A6 كانت أيام عمل المدرسين الخارجين بحدها الأدنى وكذلك الحال بالنسبة للطلاب، ولم يكن هنالك أية وقت تعطل للمدرسين الخارجيين.

لتوضيح أثر تفعيل قيود ضغط الجدول الزمني للطلاب سنعرض جدول المحاضرات الأسبوعية للسنة الثالثة الفئة الأولى والذي نتج عن التجربة A1، حيث نلاحظ أنه لم يتم تخفيض عدد أيام دوام الطلاب ولا أوقات تعطلهم ونفس الجدول ولك في الاختبار A6 بعد أن تم تفعيل قيود تخفيض عدد أيام الدوام.



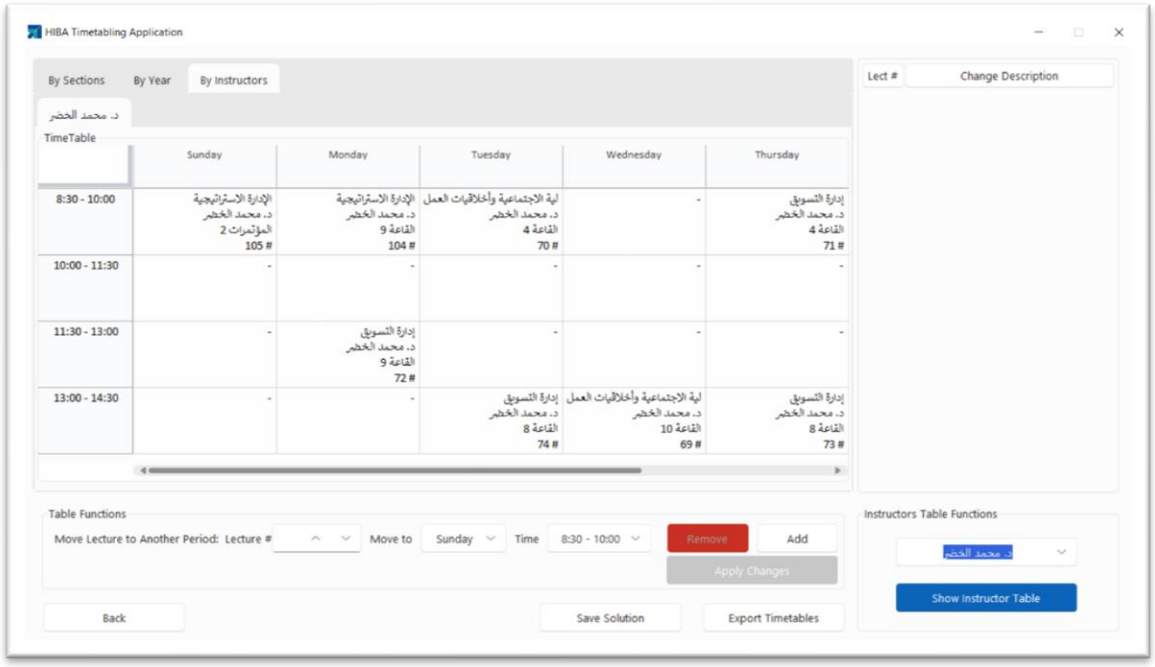
الشكل (54) واجهة الجداول الأسبوعية 6

بعد تفعيل القيود الخاصة بتخفيض أيام دوام الطلاب في الاختبار A6 تم تخفيض عدد أيام طلاب السنة الثالثة الفئة الأولى من 5 أيام إلى 4 أيام كما هو موضح في الواجهة التالية.

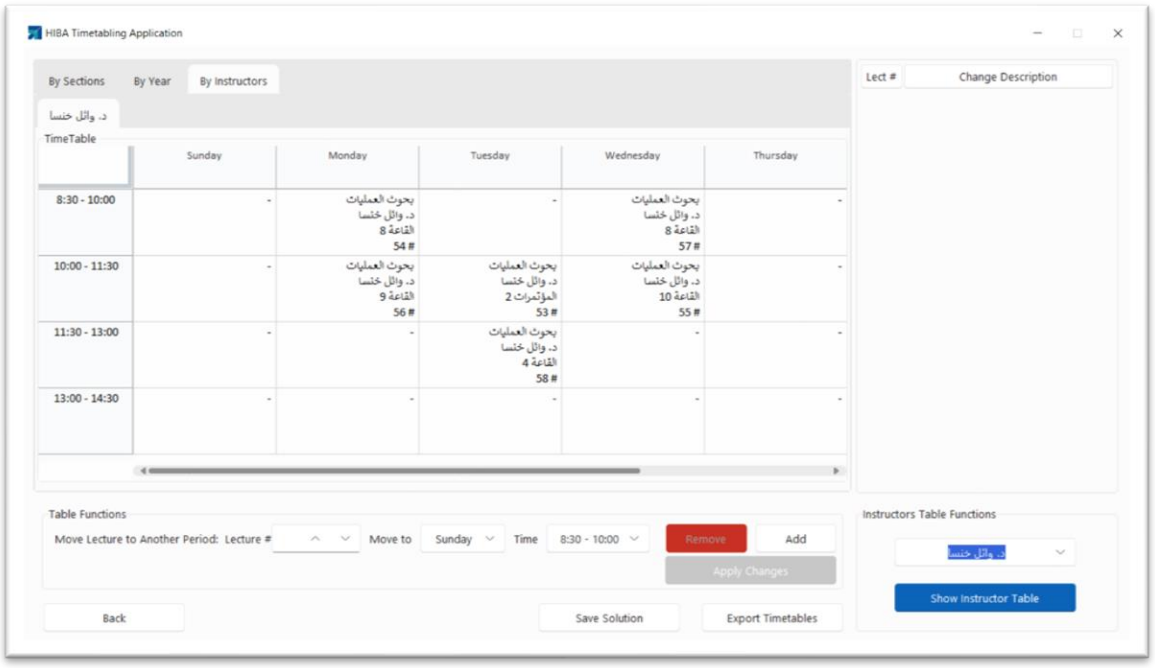


الشكل (55) واجهة الجداول الأسبوعية 7

سنقوم بمناقشة التجربة A7 كوننا قد فعلنا فيها القيود المرنة وكذلك القيود الاختيارية. الشكل التالي يوضح جداول المدرسين الذين تم تحديد الحد الأقصى لعدد محاضراتهم اليومي.



الشكل (56) واجهة الجداول الأسبوعية 8



الشكل (57) واجهة الجداول الأسبوعية 9

رأينا في واجهة القيود الاختيارية التي تم عرضها المدرسين الذين قد حددنا عدد المحاضرات اليومية الأعظمي بالنسبة لهم. وكان أحدهم مدرساً خارجياً (قد تم تفعيل قيود ضغط الجدول الزمني له) وهو الدكتور وائل خنسا، والآخر هو الدكتور محمد الخضر. وكانت النتيجة كما هو موضح في الجداول أعلاه، حيث تم ضغط الجدول الزمني للأول ولم يتم ذلك للثاني، ولم يتم إسناد أكثر من محاضرتين في كل يوم لكل منهما.

لتوضيح الفروقات بين الاختبارات من النوع A بشكل أكثر شمولاً تم تلخيص النتائج في الجدولين التاليين حيث تمت المقارنة بين نتائج ثلاثة تجارب هي A1 وA7 وA10.

المدرسين			A1		A7		A10	
أسم المدرس	داخلي خارجي	عدد المحاضرات الأسبوعية	WDs	ITs	WDs3	ITs4	WDs5	ITs6
د. إيهاب حامد	خارجي	4	3	1	1	0	1	0
د. منذر العواد	خارجي	4	4	0	1	0	1	0
د. محسن علم	خارجي	2	2	0	1	0	1	0
أ. خالد سنيور	خارجي	2	1	0	1	0	1	0
أ. هلا بريدي	خارجي	4	3	2	1	0	1	0
د. تيسير زاهر	خارجي	6	3	1	2	0	2	0
د. فداء ناصر	خارجي	2	1	0	1	0	1	0
د. وائل خنسا	خارجي	6	3	3	3	0	2	0
د. هيثم عيسى	خارجي	2	2	0	1	0	1	0
د. راضي خازم	خارجي	4	3	1	1	0	1	0
د. صلاح الدوه جي	خارجي	2	2	0	1	0	1	0
د. ريم عبود	خارجي	2	2	0	1	0	1	0
د. كادان جمعة	خارجي	2	2	0	1	0	1	0
د. أحمد العلي	خارجي	2	2	0	1	0	1	0
د. غسان الخلف	خارجي	2	1	1	1	0	1	0
د. محمد علاء الدين الزعتري	خارجي	2	2	0	1	0	1	0
د. بسام التزه	خارجي	2	1	1	1	0	1	0
د. سمير كرمان	خارجي	2	1	1	1	0	1	0
د. أحمد هواري	داخلي	4	3	1	3	0	1	0
د. زكريا سلامة	داخلي	4	2	0	3	1	1	0
د. ثناء أبا زيد	داخلي	4	3	0	3	0	1	0
د. راتب البلخي	داخلي	4	2	1	3	2	1	0
د. فاطمة صندوق	داخلي	4	4	0	3	2	1	0
د. طلال عبود	داخلي	6	4	2	4	0	2	0
د. ياسر كفا	داخلي	6	3	1	3	1	2	0
د. منال الموصلي	داخلي	6	3	3	4	1	2	0
د. محمد الخضر	داخلي	8	4	0	5	5	2	0
د. عبد الحميد خليل	داخلي	8	5	1	5	3	2	0
م. نظرة رحمة	داخلي	6	3	0	5	0	2	0
د. رعد الصرن	داخلي	3	2	0	3	0	1	0
د. نريمان عمار	داخلي	4	2	3	2	2	1	0
د. مالك نجار	داخلي	4	2	2	3	0	1	0
د. حسان اسماعيل	داخلي	4	2	0	4	0	1	0
د. رانية المجني	داخلي	2	2	0	2	0	1	0
د. آلاء بركة	داخلي	4	4	0	3	1	1	0
د. حيان ديب	داخلي	2	2	0	2	0	1	0
د. مؤيد الحاج صالح	داخلي	2	2	0	2	0	1	0
د. راغب الغصين	داخلي	4	4	0	4	0	1	0
أ. محمد رجب	داخلي	4	3	2	3	1	1	0

الجدول (15) تفاصيل نتائج مجموعة اختبارات A بالنسبة للمدرسين

في الجدول السابق تم بيان عدد أيام دوام كل مدرس Working Days (WDs) وأوقات تعطل كل مدرس Idle Times (ITs) وذلك في كل تجربة من التجارب الثلاث. بالمقارنة بين A1 وA7 نلاحظ تخفيضاً واضحاً بعدد أيام الدوام وعدد أوقات التعطل للمدرسين الخارجيين فقط. بالمقارنة بين A1 وA10 نلاحظ تخفيض عدد أيام الدوام وأوقات التعطل لكل من المدرسين الداخليين والخارجيين.

بهدف دراسة أثر تطبيق قيود ضغط الجدول الزمني للمدرسين على عدد أيام عملهم وعدد أوقات تعطلهم تم تطبيق اختبار Paired sample t-test على العينتين المتعلقتين بعدد أيام عمل كل مدرس في العينة A1 والعينة A10 قبل وبعد تفعيل قيود ضغط الجدول الزمني. وكذلك الأمر على العينتين المتعلقتين بعدد أوقات تعطل كل مدرس في العينة A1 والعينة A10. وكانت النتائج كما هو موضح في الجدولين التاليين:

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	A1_Instructors_WDs	2.54	39	.996	.159
	A10_Instructors_WDs	1.21	39	.409	.066
Pair 2	A1_Instructors_ITs	.69	39	.950	.152
	A10_Instructors_ITs	.00	39	.000	.000

الجدول (16) Paired Samples Statistics للمدرسين

Paired Samples Test

		Paired Differences					t	df	Sig. 2-tailed
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	A1_Instructors_WDs A10_Instructors_WDs	1.333	.869	.139	1.052	1.615	9.587	38	.000
Pair 2	A1_Instructors_ITs A10_Instructors_ITs	.692	.950	.152	.384	1.000	4.550	38	.000

الجدول (17) Paired Sample Test للمدرسين

بالنسبة للاختبار الأول pair 1 والذي يقارن بين متوسط عدد أيام عمل المدرسين قبل وبعد تفعيل القيود المرنة، بما أن sig المحسوبة أصغر من 0.05 يعني ذلك أن هناك فارق جوهري بين المتوسطين قبل وبعد تفعيل القيود، أي أنه يوجد أثر لتفعيل القيود على عدد أيام عمل المدرسين.

أما بالنسبة للاختبار الثاني pair 2 والذي يقارن بين متوسط عدد أوقات تعطل المدرسين قبل وبعد تفعيل القيود المرنة، بما أن sig المحسوبة أصغر من 0.05 يعني ذلك أن هناك فارق جوهري بين المتوسطين قبل وبعد تفعيل القيود، أي أنه يوجد أثر لتفعيل القيود على عدد أوقات تعطل المدرسين.

في الجدول التالي تمت المقارنة بين الاختبارات الثلاثة نفسها ولكن من حيث عدد أيام دوام الفئات (الطلاب) وأوقات تعطلهم. أيضا تم بيان عدد أيام دوام كل فئة ADs وعدد أوقات تعطل كل فئة ITs.

A10		A7		A1		الفئات	
ITs	ADs	ITs	ADs	ITs	ADs	عدد المحاضرات الأسبوعية	اسم الفئة
0	4	1	4	2	5	13	1st-1
0	4	0	4	2	5	13	1st-2
0	3	1	3	2	5	11	2nd-1
0	3	1	3	3	5	11	2nd-2
0	4	2	4	3	5	13	3rd-1
0	4	2	4	1	5	13	3rd-2
0	4	0	3	1	5	12	4th-Mktg
0	4	0	3	0	5	12	4th-OPIM
0	3	0	3	0	5	12	4th-FIN
0	4	0	3	1	5	12	4th-HR
0	4	1	4	2	4	14	5th-Mktg
0	4	2	4	2	5	14	5th-OPIM
0	4	1	4	3	5	14	5th-FIN
0	4	0	4	3	5	14	5th-HR

الجدول (18) تفاصيل نتائج مجموعة اختبارات A بالنسبة للفئات

بهدف دراسة أثر تطبيق قيود ضغط الجدول الزمني للفئات تم تطبيق اختبار Paired sample t-test وكانت النتائج كما هو موضح في الجدولين التاليين:

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	A1_Sections_WDs	4.93	14	.267	.071
	A10_Sections_WDs	3.79	14	.426	.114
Pair 2	A1_Sections_ITs	1.79	14	1.051	.281
	A10_Sections_ITs	.00	14	.000	.000

الجدول (19) Paired Sample Statistics للفئات

Paired Samples Test									
		Paired Differences					t	df	Sig. 2-tailed
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	A1_Sections_WDs A10_Sections_WDs	1.143	.535	.143	.834	1.451	8.000	13	.000
Pair 2	A1_Sections_ITs A10_Sections_ITs	1.786	1.051	.281	1.179	2.392	6.358	13	.000

الجدول (20) Paired Sample Test للفئات

بالنسبة للاختبار الأول pair 1 والذي يقارن بين متوسط عدد أيام دوام الفئات قبل وبعد تفعيل القيود المرنة، بما أن sig المحسوبة أصغر من 0.05 يعني ذلك أن هناك فارق جوهري بين المتوسطين قبل وبعد تفعيل القيود، أي انه يوجد أثر لتفعيل القيود على عدد أيام دوام الفئات.

أما بالنسبة للاختبار الثاني pair 2 والذي يقارن بين متوسط عدد أوقات تعطل الفئات قبل وبعد تفعيل القيود المرنة، بما أن sig المحسوبة أصغر من 0.05 يعني ذلك أن هناك فارق جوهري بين المتوسطين قبل وبعد تفعيل القيود، أي انه يوجد أثر لتفعيل القيود على عدد أوقات تعطل الفئات.

5

الفصل الخامس

النتائج والتوصيات

- تمهيد
- خلاصة البحث
- الآفاق المستقبلية

SWIFT

Timetable Management System

1.5 تمهيد

في هذه الفصل تم تلخيص النتائج التي توصل إليها البحث وذلك على الصعيدين النظري والعملي. بالإضافة إلى مجموعة من الافاق المستقبلية التي قد يكون للتطوير عليها قيمة مضافة كبيرة. سواء من الناحية التقنية، أي كل ما من شأنه تحسين جودة وكفاءة الحل، أو من الناحية الفنية مثل إضافة ميزات جديدة للنظام.

2.5 خلاصة البحث

من خلال هذا البحث تم التعريف بمسألة الجدولة الزمنية ومناقشة أنواعها وخصائصها المختلفة ومتطلبات المؤسسات المتباينة وبيان أوجه الاختلاف فيما بينها. ونتيجة للكم الكبير من المتطلبات التي ترغب الكثير من المؤسسات التعليمية في تحققها تم بيان مدى تعقيد المشكلة وعرض مجموعة من المنهجيات التي عمل الباحثون على تطويرها في العقود الأخيرة الماضية بغرض حل هذه المشكلة بأفضل كفاءة ممكنة. كما تم عرض مجموعة من الأنظمة – منها المجاني ومنها المدفوع – والتي تختلف بشكل كبير من حيث مجموعة المواصفات التي تقدمها.

في الجانب النظري للبحث تم التعريف بالعديد من المفاهيم المحورية التي بُني البحث على أساسها. حيث تم البدء بالحديث عن بحوث العمليات بشكل عام ومن ثم تم تخصيص قسم للحديث عن البرمجة الخطية والبرمجة الطبيعية. كما تم تخصيص قسم مستقل للحديث عن دورة حياة تطوير النظم ومراحلها، بالإضافة إلى عرض مجموعة المنهجيات الأكثر شيوعاً في تحليل وتصميم النظم.

في الجانب العملي للبحث تم تحديد متطلبات النظام من خلال:

1. التعرف على النظام التعليمي في المعهد العالي لإدارة الأعمال.
2. تحليل نظام الجدولة الحالي في المعهد العالي لإدارة الأعمال.
3. إجراء بعض المقابلات مع الإداريين في المعهد.
4. صياغة مسألة الجدولة الزمنية للمقررات الدراسية بشكل واضح.

بناء على المتطلبات تم تحديد حالات الاستخدام وتوصيفها. وبعد ذلك تم تصميم قاعدة المعطيات من خلال تحديد الكيانات الرئيسية في النظام ودراسة العلاقات بين هذه الكيانات.

في مرحلة التصميم تم عرض فكرة الحل الأساسية والتي سيقوم النظام بالاعتماد عليها في إيجاد حلول للمسألة. تكونت فكرة الحل من مرحلتين رئيسيتين يعملان بشكل متكامل، تم التفصيل في مناقشتهما في الجانب العملي للبحث، وهما نموذج البرمجة الطبيعية المختلطة الذي تم بناؤه ليقوم بوظيفة الجدولة الزمنية مع تحقيق متطلبات ضغط الجدول الزمني، وخوارزمية تعيين القاعات الدراسية.

القسم الخاص بمرحلة التنجيز هو الخاص بمرحلة ما بعد التصميم، حيث تم التعريف بالأدوات المستخدمة في عملية التطوير من لغات برمجة ومكتبات ونظم إدارة قواعد البيانات. وعرض وشرح واجهات الاستخدام التي تم بناؤها لإتاحة التعامل بشكل مبسط مع النظام. وأخيرا مجموعة من الاختبارات أو التجارب التي تختلف عن بعضها البعض في حجم البيانات المدخلة وتنويعة القيود التي تم تفعيلها، حيث هدفت هذه

الاختبارات إلى بيان مدى كفاءة النظام من حيث المدة الزمنية التي يستغرقها لإيجاد الحلول، وجودة هذه الحلول ومدى تحقيقها للمتطلبات.

النتيجة النهائية للبحث هي نظام إدارة الجدولة الزمنية TSwift والذي تم تطوير من الصفر إلى أن أصبح جاهزاً تماماً للاستخدام.

3.5 الآفاق المستقبلية

يمكن تلخيص الافاق المستقبلية بالنقاط التالية:

1. إضافة ميزة للنظام تسمح للمدرسين بتحديد فتراتهم المتاحة والغيرة متاحة عن طريق الويب.
2. إضافة خوارزميات أو نماذج أخرى للحل، وبالتالي تتاح للمستخدم إمكانية اختيار طريقة الحل.
3. تطوير نموذج البرمجة الرياضية ليأخذ بعين الاعتبار تفضيلات المدرسين وليس فقط فتراتهم المتاحة أو الغير متاحة.
4. دراسة إمكانية تجزيء عملية حل المسألة إلى مراحل متعددة دون المساس بجدودة الحل النهائي.
5. تطوير خوارزمية تعيين القاعات لتأخذ بعين الاعتبار مجموعة القاعات المفضلة لكل مدرس.
6. الربط بين النظام وبين قاعدة المعطيات الرئيسية للمعهد العالي لإدارة الأعمال.

6

الفصل السادس
المراجع

SWIFT
Timetable Management System

1.6 المراجع الأجنبية

[1] A. Schaerf. A Survey of Automated Timetabling. *Artificial Intelligence Review, Pages 87–127* (1996)

[2] Alvarez-Valdes, Ramon, Enric Crespo, & Jose M. Tamarit. Design and implementation of a course scheduling system using Tabu Search. *European Journal of Operational Research, Pages 512-523* (2002)

[3] Árton P. Dorneles, Olinto C. B. de Araújo & Luciana S. Buriol. The Impact of compactness requirements on the resolution of high school timetabling problem (2012)

[4] Badri, Masood A. A two-stage multiobjective scheduling model for [faculty-course-time] assignments. *European Journal of Operational Research, Pages 16-28* (1996)

[5] Badri, Masood A., Donald L. Lewis, Donna F. Davis, & John Hollingsworth. A multi-objective course scheduling model: Combining faculty preferences for courses and times. *Computers & Operations Research, Pages 303-316.* (1998)

[6] Burke, E.K., & Sanja Petrovic. "Recent Research Directions in Automated Timetabling. *European Journal of Operational Research, Pages 266-280.* (2002)

[7] Carrasco, M.P., and M.V. Pato. A Comparison of Discrete and Continuous Neural Network Approaches to Solve the Class/Teacher Timetabling Problem. *European Journal of Operational Research, Pages 65-79.* (2004)

[8] Costa, Daniel. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research, Pages 98-110.* (1994)

[9] D. J. A. Welsh, M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal, Pages 85–86.* (1967)

[10] E. A. Akkoyunlu. A linear algorithm for computing the optimum university timetable. *The Computer Journal, Pages 347-350.* (1973)

[11] E.K. Burke, G. Kendall & E. Soubeiga. A Tabu-Search Hyperheuristic for Timetabling and Rostering. *Journal of Heuristics, Pages 451–470.* (2003)

[12] Michael W. Carter & Gilbert Laporte. Recent developments in practical course timetabling. (1998)

[13] N. L. Lawrie. An integer linear programming model of a school timetabling problem. *The Computer Journal, Pages 307–316.* (1969)

[14] Nevena MITROVIC. The University Timetabling Problem – Complexity and an Integer Linear Programming Formulation: a Case Study of UP FAMNIT. (2017)

[15] Rasmus Ørnstrup Mikkelsen. Application of Mixed Integer Programming Methods for Practical Educational Timetabling. (2021)

[16] S. Srinivasan. Modeling the Homeschool timetabling problem using Integer programming. (2011)

[17] Schniederjans, Marc J., & Gyu Chan Kim. A goal programming model to optimize departmental preference in course assignments. *Computers & Operations Research, Pages 87-96.* (1987)

[18] T Birbas, S Daskalaki & E Housos. Timetabling for Greek high schools. Journal of the Operational Research Society, *Pages 1191–1200*. (1997)

[19] Victor A. Bardadym. Computer-aided school and university timetabling: The new wave. (1996)

[20] Yu, Enzhe, and Ki-Seok Sung. A Genetic Algorithm for a University Weekly Courses Timetabling Problem. *International Transactions in Operational Research, Pages 703 – 717.* (2002)

[21] UniTime Website

[22] Prime Timetable Website

[23] Bullet Solutions Website

[24] FET Timetabling

[25] Alan Dennis, Barb Wixom, Robby Roth. System Analysis and Design, Fifth Edition. *Wiley, Pages 5-45.* (2012)

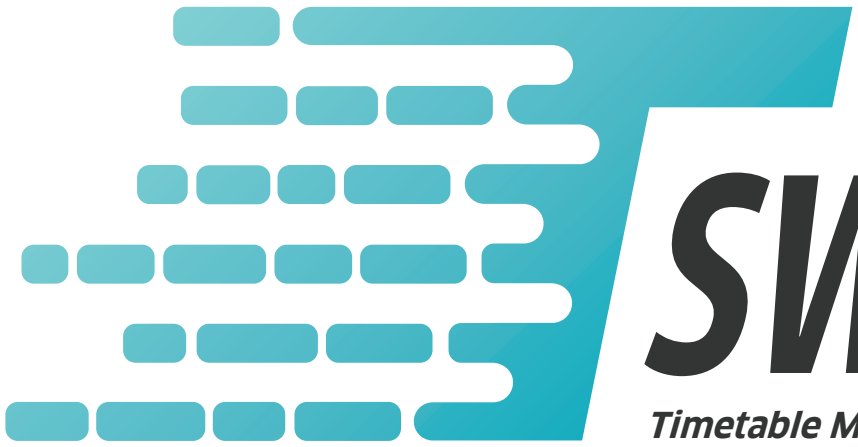
2.6 المراجع العربية

[26] م. محمد راتب حباب .توليد الجداول الدراسية اعتماداً على الخوارزميات الجينية وتعلم الالة تطبيق على المعهد العالي. (2021)

[27] مـــحـــمـــد الريّـــــس. تحليل وتصميم تطبيق خاص بمتجر الكتروني مع نظام دفع الكتروني ونظام إدارة عمليات توصيل الطلبات E-Mercato. (2019)

[28] الموقع الالكتروني للمعهد العالي لإدارة الأعمال.

نهاية البحث



SWIFT

Timetable Management System