الجمهوريـــة الجزائريـــــة الديمقراطيـــة الشعبيـــة

وزارة التعليم العالي والبحث العلمي

جامعــة حمـــة لخضـــر الـــــوادي
كلية العلوم الدقيقة
قسم الإعلام الآلي

مذكرة نهاية التخرج
تدخل ضمن متطلبات الحصول على شهادة

ليســانس أكاديمي

الميدان: رياضيات وإعلام آلي
الشعبة: الإعلام الآلي
التخصص: أنظمة معلوماتية

الموضوع

تصميم وإنجاز تطبيق مكتبي لإدارة المقاولات

من إنجاز الطلبة:

- نعمي سامي
- بن ناجي زياد
- الوردي عبد الفتاح

من اقتراح وتأطير الأستاذ:عباس مسعود

نوقشت يوم 2020/09/27 أمام اللجنة المكونة من الأساتذة:

خلايفة عبد الناصر رئيسا

غربي قدور مقررا

السنة الجامعية: 2020/2019

المحتويات

ملخص.....4

مقدمة عامة.....9

طرح الإشكالية.....10

I . الفصل الأول: مفاهيم عامة.....11

تمهيد.....12

I . 1 مفهوم شركة المقاولات.....12

I . 2 مفهوم المشروع.....12

I . 3 مفهوم ورشة العمل.....13

I . 4 مفهوم العمال.....13

I . 5 مفهوم العميل.....14

I . 6 مفهوم العتاد.....14

I . 7 مفهوم المورد.....15

II . الفصل الثاني: النمذجة والتصميم.....16

تمهيد.....17

II . 1 مواصفات المشروع.....17

II . 2 تصميم باستخدام UML.....17

II . 3 مقدمة عن لغة UML.....18

II . 4 مخططاتUML.....19

II . 5 أداة UMLالمستعملة.....20

II . 5 . 1 مخططات حالة الاستخدام use case.....20

1. مخطط حالة الاستخدام.....21

2.مخطط حالة استخدام المشروع.....22

3. مخطط حالة استخدام للعمال.....22

4. مخطط حالة استخدام للعتاد.....23

5. مخطط حالة استخدام للمخزن.....23

II.5.2 مخطط التتابع.....24
II.5.3 مخطط الصنف.....25
II.6 النموذج العلائقي وقاموس البيانات.....27
II.7 الخاتمة.....34
III . الفصل الثالث: التنفيذ.....35
تمهيد.....36
III.1 أدوات التنفيذ.....36
III.2 إطار العمل 39
III.3 SQL SERVER.....39
III.4 بعض نوافذ التطبيق.....40
III.5 خلاصة.....43
خاتمة عامة.....44
قائمة المراجع.....45

الــمــلــخص

جعلنا مشروع تخرجنا هذا تحت عنوان "تصميم وإنجاز تطبيق مكتبي لإدارة المقاولات" قمنا بإنشاء تطبيق من أجل تحقيق المصداقية والسرعة وسهولة التسيير نظرا للمشاكل واجهة إدارة الشركة والتي نذكر منها:

- العدد الكبير من العمال الذي أدى إلى صعوبة تقييمهم وحالتهم.
- تنوع العتاد مما يصعب معرفة ما إذا كان قيد العمل أم في المستودع أو مؤجر أو معطل وتلبية ما يحتاجه العتاد من صيانة أو وقود ..... إلخ.
- تعدد ورشات العمل الذي قد أقحم إدارة الشركة في صعوبة إسناد العمال والعتاد ومستلزمات كل ورشة.
- صعوبة تسديد الرواتب الشهرية للعمال كل حسب عمله وساعات عمله.
- عملية كراء العتاد والمشاكل المترتبة عنها.

حاولنا البحث عن وسيلة تقلل من هذه المشاكل ولعلها كانت تكمن في إنشاء تطبيق مكتبي يقوم بتخزين ومعالجة البيانات بسرعة وسهولة بحيث يسمح لنا بـ:

- تسيير المشاريع وتعيين المسؤول عنها وتحديد الزبون والورشات التابعة لها.
- إدراج معلومات العمال للتطبيق وعرضها والتعديل عليها.
- إسناد العمال والعتاد لورشات العمل.
- تقييم يومي للعمال (حساب الساعات العادية والساعات الإضافية).
- معرفة حالة العمال (مكان تواجد كل عامل).
- حساب الراتب الشهري للعمال مع إمكانية طباعة كشوف رواتبهم.
- معرفة كل ما هو موجود في ورشات العمل من عتاد وعمال.
- معرفة ما هو موجود في المستودع.

أولا بمرحلة التعرف على مفاهيم الموضوع ثانيا النمذجة بواسطة لغة النمذجة الموحدة وأخيرا أنشأنا تطبيقا يحقق ذلك.

Abstract

Our graduation project is titled “Design and Implementation of a Desktop application for Companies Management” where we created an application in order to achieve credibility, speed and ease of management due to the problems that meet the company's management, which we mention, including:

- The large number of workers, which made it difficult to assess and condition them.
- The diversity of the equipment, which makes it difficult to know whether it is in operation, in the warehouse, rented, or idle, and to meet the needs of the equipment in terms of maintenance or fuel ..... etc.
- The multiplicity of workshops that have forced the company’s management into the difficulty of assigning workers, equipment and requirements for each workshop.
- Difficulty paying workers' monthly salaries, each according to his work and hours of work.
- Equipment rental process and the problems arising from it.

We tried to find a way to reduce these problems and perhaps it was to create a Desktop application that stores and processes data quickly and easily, allowing us to:

- Running projects, appointing the person responsible for them, and identifying the client and its affiliated workshops.
- The inclusion, presentation and modification of worker information for the application.
- Assigning workers and equipment to the workshops.
- Daily evaluation of workers (calculating regular and overtime hours).
- Knowing the status of workers (where each worker is).
- Calculating the monthly salary of workers with the ability to print their payroll.
- Knowing everything that is in the workshops, including equipment and workers

- Knowing what is in the warehouse.
  Firstly, by getting acquainted with the concepts of the topic, second by modeling with the Unified Modeling Language, and finally we created an application that achieves this.

Résumé

Nous avons réalisé ce projet de fin d'études intitulé "Conception et mise en œuvre d'une application bureautique pour la gestion des entreprises". Nous avons créé une application afin d'obtenir fiabilité, rapidité et facilité de gestion en raison des problèmes et de l'interface de gestion de l'entreprise, que nous mentionnons, notamment:

- Le grand nombre de travailleurs, qui a rendu difficile leur évaluation et leur conditionnement.
- La diversité des équipements, qui rend difficile de savoir s'il est en fonctionnement, dans l'entrepôt, loué, ou inactif, et de répondre aux besoins des équipements en termes de maintenance ou de carburant etc.
- La multiplicité des ateliers qui contraignent la direction de l'entreprise, en particulier, lors de l'affectation les travailleurs, les équipements et les besoins de chaque atelier.
- Difficulté de payer les salaires mensuels des travailleurs, chacun en fonction de son travail et de ses heures de travail.
- Processus de location d'équipement et les problèmes qui en découlent.

Nous avons essayé de trouver un moyen de réduire ces problèmes en créant une application bureautique qui stocke et traite les données rapidement et facilement, nous permettant de:

- Conduire les projets, désigner la personne responsable de ceux-ci et identifier le client et ses ateliers affiliés.
- L'inclusion, la présentation et la modification des informations des travailleurs pour l'application.
- Affectation des travailleurs et du matériel aux ateliers.

- Évaluation quotidienne des travailleurs (calcul des heures régulières et supplémentaires).
- Connaître le statut des travailleurs (où se trouve chaque travailleur).
- Calcul du salaire mensuel des travailleurs ayant la possibilité d'imprimer leur paie.
- Connaître tout ce qui se trouve dans les ateliers, y compris l'équipement et les travailleurs.
- Savoir ce qu'il y a dans l'entrepôt.

Premièrement, en se familiarisant avec les concepts du sujet, ensuite en modélisant avec le langage de modélisation unifié, et enfin nous avons créé une application qui y parvient.

مقدمة عامة

يعتبر الأعلام الآلي من العلوم الأكثر تأثيرا في حياتنا اليومية حيث ارتقى في شتى العلوم من مستوى الاستغلال التصوري البسيط إلى المستوى التطبيقي الآلي وذلك نظرا لما يقدمه من الخدمات التي تتميز بالدقة وسرعة معالجة المعلومات.

فقد أصبح يفرض على معظم المؤسسات مهما كان نوع نشاطها إمكانية الاستفادة من مزايا خدماته ولمواجهة الحجم الهائل للمعلومات المتداولة فيها أحال من الضروري إدخال أعلام الآلي في مجال إداراته وذلك لغرض زيادة فاعلية العمل وتحقيق أهداف المؤسسة ومن بين الخدمات التي يقدمها للمؤسسة في مجال التسيير والإدارة هي إمكانية تصميم وإنجاز تطبيقات برمجية تعمل على السير الحسن للإدارة.

ومن هذا المنطلق ارتأينا إلى إسقاط هذا التأثير على حقل دراستنا ألا وهي إدارة تصميم وإنجاز تطبيق مكتبي للإدارة المقولات.

صار نشاط البناء والأشغال العمومية جزءا لا يتجزأ من التنمية الصناعية والاقتصادية للبلاد حيث وفر العديد من المرافق الأساسية كالمستشفيات والمدارس والعمارات والطرقات غيرها إضافة لتشييد البنى التحتية الضرورية بما فيها إمدادات المياه والصرف الصحي كما عملت على تعبيد الطرق وهذا كله ساعد على توفير فرص العمل للعديد من العمال.

عالجنا في موضوع تخرجنا طريقة تسيير وتصميم نظام أو أداة من أجل تحسين وتطوير إدارة المشاريع (تطبيق مكتبي للإدارة المقولات) بهدف تحقيق سهولة العمل والمصداقية والسرعة والتنظيم أحسن.

- ❖ في المرحلة الأولى:

  درسنا المفاهيم التي تتعلق بالموضوع وكيفية تسيير المنظومة.

- ❖ في المرحلة الثانية:

  ترجمنا ذلك بتصميم " نمذجة اللغة موحدة "

- ❖ في المرحلة الأخيرة:

  قمنا بتحويل هذا التصميم إلى برنامج بلغة البرمجة C# بواسطة بيئة التطوير Visual Studio

طرح الإشكالية :

في كل مشروع يتوجب على إدارة الشركة بتنظيم ورشات العمل وإسناد العمال لكل ورشة وتمويلها بالعتاد اللازم، حيث لكل من العمال والعتاد تقييم وميزانية خاصة وفي ذلك نسلط الضوء على فئة العمال لأن لهم القسم الأكبر من الإجراءات (التقييم اليومي، الراتب الشهري، التأمين الصحي...) ونظرا لحساسية المرحلة تواجه هذه الشركة العديد من المشاكل منها:

- مشكلة ارتكاب الأخطاء.
- مشكلة تقييم العمال والعتاد.
- مشاكل تسديد رواتب العمال.
- مشاكل إدارة العتاد (صيانة العتاد، فواتير الكراء، ....).

الفصل الأول:
مفاهيم عامة

مفـاهيم عـامـة

تمهيد:

خلال هذا الفصل سنقوم بدراسة أولية للنظام الذي نريد إنشاؤه قصد جمع أكبر حجم ممكن من المعلومات حول النظام، والتي من خلالها نتوصل إلى بحث دقيق للواقع نفسه ونستطيع بواسطتها القيام بحل تنظيمي وإنشاء تصوري مستقبلي للنظام.

1.مفـهـوم شركـة الـمقـاولات:

يعرف نشاط المقاولات بأنه ذلك النشاط الذي من خلاله يتم تنفيذ مشاريع البناء والتشييد ومشاريع البنية التحتية وغيرهم من المشاريع الإنشائية المختلفة. وشركات المقاولات هي المنوطة بتنفيذ هذه المشاريع .إن مصادر حصول شركات المقاولات على المشاريع يتمثل في قطاعين.

- ✓ **القطاع الأهلي**

ويتمثل في المنشأة الأهلية مثل العمارات السكنية والفيلة الخاصة والمنشأة الأخرى التي يملكها أفراد أو حتى شركات خاصة مملوكة لأفراد

- ✓ **القطاع الحكومي بكافة قطاعاته**

وتتمثل هذه المشاريع في المشروعات السكنية والصناعية والجسور والأنفاق والمنشأة الرياضية وإلى آخره من المشروعات الخدمية ومشاريع البنية التحتية ويعتبر التعامل في المشاريع الخاصة من حيث التقيد بالقواعد المتعارف عليها في هذا المجال أيسر بكثير من المتعامل بها في المشاريع الحكومية من حيث التيسير في مواعيد الدفعات او ارتباطها بنسب انجاز معينة أو الضمانات التي يتم تقديمها نظير الدفعات المقدمة أو كفالات حسن الانجاز وغيرها بالرغم أن هناك شركات خاصة يتم تطبيق كل هذه الشروط فيها.

2.مفـهـوم الـمشروع:

هو عمل تقوم به الشركة لتنفيذ فكرة معينة، سوء كانت منتج أو خدمة، وتستخدم فيه موارد معينة وتنفق من أجله أموال للحصول على منافع خلال فترة زمنية محددة ومشاريع هذه الشركة متعددة منها ما هو خدماتي (كراء العتاد) أو مشاريع البناء أو الأشغال العمومية حيث يحوي كل مشروع على ورشة عمل أو عدة ورشات حسب احتياجات ومتطلبات كل مشروع حيث لكل مشروع مسؤول يقوم بتسيير ورشاته وتلبية احتياجات كل ورشة من عتاد وسلع وعمال مع العلم أن لكل مشروع تاريخ بداية وتاريخ نهاية أي مدة تنفيذ المشروع وينقسم إلى مراحل:

1.2مرحلة إعداد المشروع

في هذه المرحلة يتم وضع الإطار العام للمشروع ونأتي هنا إلى ما يسمى بدراسة الجدوى والتي بدورها تنقسم إلى:

- الدراسة الفنية.
- دراسة الهيكل الإداري والتنظيمي.
- الدراسة الربحية أو المالية.

2.2مرحلة تنفيذ المشروع:

وفي هذه المرحلة يتم تطبيق المشروع والإشراف عليه وتسجيل ما تم تنفيذه.

3.2 مرحلة تقييم وتسليم المشروع:

ويكون تقييم المشروع أثناء التنفيذ للمقارنة بين الوضع الحالي ومسار المشروع، ويتم التقييم أيضا بعد التنفيذ للمقارنة ما بين النتائج النهائية ومدى وصول المشروع لهذه النتائج، إذا حقق المشروع النتائج المحددة يتم تسليمه.

3.ورشة العمل:

نستطيع القول عن ورشة العمل أنها مكان عمل المشروع أو جزء من المشروع تحت إشراف مسؤول الورشة الذي يقوم بدوره بإسناد الأدوار للعمال التي تربط بينهم علاقة وطيدة حيث كل عامل هو مكمل لبقية العمال وبطبيعة الحال تحتاج الورشة إلى العديد من المستلزمات كالعتاد (شاحنات ، جرارات حفارات ...) و مواد الخام والسلع (كالإسمنت و الرمل و الحصى ...) ولكل ورشة عمل وموقع محددين كما لها مدة للإنجاز (تاريخ بداية و تاريخ نهاية) و تكون هذه الأخيرة محصورة بين تاريخ بداية المشروع و نهايته ، وتحدد لها ميزانية خاصة بما تتطلبه الحاجة و تختلف الورشات فلكل ورشة ما يميزها عن الأخرى من حيث عدد العمال أو طبيعة عملها وتعرف كل ورشة بالوصف التي أسندت إليه وهذا الوصف معروف عند جميع عمال الورشة.

4. العمال:

العامل هو النواة الأساسية في الشركة حيث العامل هو ذلك الشخص الذي يؤدي نشاط يدوي و يأخذ عادة أجرا ماديا و معنويا على عمله في المنشأة أو المشروع نظير خدماته المهنية في المشروع حيث أن لكل عامل تاريخ محدد لبداية عمله ونهايته، في كل يوم يعمل العامل 8 ساعات عمل كساعات عادية ويمكن أن يعمل ساعات عمل إضافية جنب ساعات عمله العادية إذا تطلب الأمر، و أيضا لكل عامل

منصب يقمصه حسب المستوى الدراسي أو المعرفي أو قدرته وأيضا للعامل تقييم يومي الذي به تستطيع الشركة دفع الراتب الشهري للعامل .

1.4 *التقييم والراتب الشهري للعامل:*

كما ذكرنا سابقا لكل عامل 8 ساعات عمل عادية في اليوم ويمكنه إضافة ساعات إضافية أخرى إن لزم الأمر كما تمنح للعامل ساعات فراغ بين الفترة الصباحية والفترة المسائية للصلاة والراحة والغذاء ...

الراتب الشهري للعامل يحدد بمجموع الساعات العادية مضروبة في سعر الساعة العادية ومجموع الساعات الإضافية مضروبة في سعر الساعة الإضافية يحدد سعر الساعة على حسب المنصب الذي تقمصه العامل كما لكل عامل إمكانية استخراج وثيقة كشف الراتب حيث من خلالها يستطيع معرفة راتبه عدد ساعات العمل العادية والإضافية وكل تفاصيل التي قام بها خلال ذلك الشهر.

2.4 منصب العمل:

المنصب كلمة ذات أصل عسكري تعني المكان الذي يوضع فيه شخص ما للقيام بمهام معينة وظيفة محددة يتدرج ضمن مجموعة مهام أو وظائف مترابطة ويعرف المنصب أيضا أنه مجموعة من الواجبات التي تؤدى بواسطة فرد ما ولها صفة التكرار وصفة التماثل وتحتاج إلى تجانس في المعارف والمهارات وقد عرف ستيفن منصب العمل بقوله " هو موضع يندرج كهيكل مصمم ومهيأ للإنجاز مجموعة محددة من المهام الموكلة إلى أحد الاعوان"

5. العميل:

وهو الزبون الذي يريد إنجاز المشروع والذي يكلف الشركة التي نحن بصدد دراستها بمقابل مبالغ مالية يستفيد منها أصحاب الشركة وعادة ما يكون العملاء شركة خاصة أو شركة عمومية أو حتى الخواص، بحيث تتواصل إدارة الشركة مع العميل بواسطة بريد الكتروني ورقم الهاتف أو حتى في أرض الواقع حيث لكل عميل اسم ولقب وعنوان ورقم خاص به، كما تقوم الشركة بتقييم المشروع قبل البدا فيه للتفاهم مع العميل في شروط العمل والوصول إلى مبلغ يرضي الطرفين.

6. العتاد:

العتاد هو مجموعة لألات كالجرارات والحفارات وشاحنات التي تستخدم من قبل العمال في ورشات العمل بحيث يتعدد أنواع العتاد حسب كل ورشة عمل ومتطلباتها وطبيعة العمل المسؤولة عليه في ذلك المشروع، كما لكل آلة اسم ونوع ورقم تسجيل وتاريخ ضمان ونهايته وتاريخ بداية الفحص ونهايته وبطاقة

رمادية، تتعدد وظيفة الألة ومكان العمل أي لكل آلة ورشة محددة وقد تكون الألة ملك للشركة أو مستأجرة من طرفها أو من الخاص وكما يمكن أيضا للشركة أن تأجر.

1.6 *تقييم العتاد المستأجر عند الشركات أخرى:*

للعتاد المأجور 8 ساعات عمل عادية في مع يمكنه إضافة ساعات إضافية أخرى إن لزم الأمر ، بحيث يكون سائق العتاد المأجور عامل لدى هذه الشركة يعني تأجر الألة معا السائق ، راتب العتاد يحدد حسب فترة العمل حيث يحسب بمجموع الساعات العادية مضروبة في سعر الساعة العادية ومجموع الساعات الإضافية مضروبة في سعر الساعة الإضافية يحدد سعر الساعة على حسب كل آلة حيث لكل الألة سعر خاص بها كما استخراج فاتورة كراء العتاد حيث من خلالها يستطيع معرفة القيمة و عدد ساعات العمل العادية والإضافية و كل تفاصيل التي قامت بها الألة خلال تلك الفترة .

7. المورد :

المورد هو أي شخص أو الشركة التي تزود شركة أو مجتمع آخر بشيء ما تقوم بتوفير احتياجات المنشأة من خامات ووقود أو منتجات وسيطة أو منتجات تامة الصنع سواء أو قطع غيار لجميع لألات الموجودة لدى الشركة، أي يوفر احتياجات الشركة من سلع ومواد التي تحتاجها الشركة في إنشاء مشروعها.

الفصل الثاني:
نمذجة والتصميم

تـمـهـيـد :

الهدف من هذا الفصل هو تصميم ونمذجة برامج إدارة مشاريع مراقبة الموظفين والموارد والموقع. في هذا الفصل سنقدم المشروع باستخدام نماذج UML والتي من خلالها سنتعرف على أهم جزئيات التصميم، سنستخدم أداة Start UML، ونبدأ مع متطلبات التقديم ومواصفات المشروع، بعدها نقدم مناقشة المخططات الرئيسية لنموذجنا. وفي الأخير ننهي عملنا بتقديم قاموس المعطيات.

1.مـواصفـات الـمشروع:

هدفنا هو إنجاز تطبيق يتناسب مع خطوات إدارة المشاريع "إدارة المشاريع مراقبة الموظفين والموارد والموقع " لإدراج جميع المعلومات المتعلقة بالشركة، العمال، العتاد، ورشات العمل، الموزعون، المخازن، العملاء.

في قاعدة البيانات الخاصة بنا فإننا سنحتاج إلى المعلومات التالية:

- ❖ معلومات الشركة (اسم الشركة، عنوان الشركة، رقم وبريد الإلكتروني للشركة).
- ❖ معلومات لكل عامل (اسم، لقب، تاريخ الميلاد، المنصب، الهاتف).
- ❖ معلومات لكل عتاد (رقم التسلسلي، اسم، الوزن، الوثائق).
- ❖ معلومات لكل مشروع (اسم المشروع، تاريخ بداية ونهاية المشروع، العميل).
- ❖ لكل مشروع ميزانية وعدد ورشات معين حسب طاقته.
- ❖ بالنسبة لكل عامل منصب معين يتم تعينه حسب شهادته وكفاءته.
- ❖ بالنسبة لكل عتاد لديه عمل مخصص له حسب نوعية العتاد وإمكانيته.
- ❖ كل عامل وعتاد ينتمي إلى ورشة عمل، وورشة العمل بدورها تنتمي إلى مشروع معين.
- ❖ لجميع العمال والعتاد تقييمات خاصة بها كل حسب وظيفته وإمكانيته.
- ❖ للمشروع عامل يترأسه ويشرف على مهامه.

2. تـصـمـيـم بـاسـتـخـدام UML:

النمذجة هو خلق تمثيل مبسط لمشكلة (نموذج). من الممكن أن يكون مجرد مشكلة، وهو مفهوم ومحاكاة. هذا النموذج هو تمثيل ممكن للنظام للحصول على غرض معين. في هذه الخطوة سنستخدم لغة النمذجة UML

3. مقدمة عن لغة UML :

UML هي لغة النمذجة الموحدة (Unified Modelling language): وهي لغة نمذجة رسومية تقدم لنا صيغة لوصف العناصر الرئيسية للنظم البرمجية. تتجه UMLبطبيعتها نحو بناء البرمجيات كائنية المنحى " Oriented Oject ".

1.3 ماهي UML:

لغة النمذجة الموحدة (Unified Modelling language) هي لغة نمذجة رسومية تقدم لنا صيغة لوصف العناصر الرئيسية للنظم البرمجية (هي عناصر تسمى artifacts مشغولات في UML) وهي لغة قياسية للتحديد، تصور، بناء وتوثيق الأعمال للبرمجيات الأنظمة، فضلا عن نماذج أعمال التجارية وغيرها من النظم المختلفة عن البرمجيات. تمثل مجموعة من أفضل التطبيقات الهندسية التي ثبت نجاحها في نمذجة النظم الضخمة والمعقدة.

حيث تعتبر جزء هام لتطوير البرمجة الكائنية التوجه " Object Oriented Softwore" وعملية تطوير البرمجيات "Softwore Développement". وتستخدم الرموز الرسومية في الغالب للتعبير عن طريقة تصميم مشاريع البرامج. يساعد فريق العمل في المشروع الذي يستخدم لغة النمذجة الموحدة في التواصل، استكشاف إمكانات التصاميم، والتحقق من صحة التصميم الهندسي لبرنامج.

2.3 أهمية استخدام لغة النمذجة الموحدة UML:

- تصميم البرمجيات بشكل احترافي.
- توثيق التصميم قبل البدء بالبرمجة.
- إعادة الاستخدام Reusability (تخفيض الكلفة).
- سهولة التعديل والصيانة وبكلفة منخفضة.
- مخططات UML تساعد المطورين على فهم النظام بسهولة وسرعة.
- لغة التواصل بين المطورين والمصممين.
- تقدم لنا اللغة رموزا ننتج بها نماذج.

3.3 إيجابيات لغة الموحدة (UML):

- ✓ تزويد المستخدمين بلغة نمذجة بصرية تعبيرية جاهزة للاستعمال بحيث يتمكنون من تطوير وتبادل النماذج التعبيرية.
- ✓ توفير قابلية التمدد وآليات التخصيص، لتوسيع المفاهيم الأساسية للمشروع.

- ✓ تكون مستقل عن لغات البرمجة الخاصة وعمليات توفير مناهج أو قواعد أساسية لفهم لغة النمذجة.
- ✓ تشجيع نمو كائنية التوجه وأدوات السوق.
- ✓ دمج أفضل الممارسات ودعم أعلى مستوى تطوير المفاهيم.

4. مخططات ال UML:

يتكون UML من أربعة عشر مخطط التي تستخدم دلالات دقيقة. واستخدمها الباحثون لتصميم نماذج يمكن الاعتماد عليها وللحد من الأخطاء البرمجية الكبيرة (أخطاء التصميم).

تمكننا لغة البرمجة الموحدة من تحويل مشروع في الواقع إلى تصميم أقرب للفهم وترجمته بإحدى لغات البرمجة (C# .... Java).

المخطط يعطي للمستخدم وسيلة لعرض والتعامل مع العناصر النمذجة. يوجد ثلاثة عشر نوع من المخططات وهي كافية لوضع نموذج لغالبية الأنظمة.

وتنقسم هذه المخططات إلى مجموعتين:

أ- مخططات ثابتة:

- مخطط فئة.
- مخطط الكائن.
- مخطط مكون.
- مخطط نشر.
- مخطط لحزم.
- مخطط هيكل مركب.

ب- المخططات السلوكية:

- مخطط حالة استخدام.
- مخطط حالة الانتقال.
- مخطط الحركة.
- مخطط تتابع.
- مخطط الاتصالات.
- مخطط التفاعل العام.
- مخطط التوقيت.

5. أداة UML المستعملة:

برنامج رسم المخططات Start UML هو برنامج يساعد على رسم المخططات المستخدمة في عمليات البرمجة وتصميم البرامج ويقوم برسم الجداول والأشكال بكل سهولة لواجهة البرنامج. يعمل تحليلات دائمة على شكل وثائق Word or PDF، ليسهل التعامل معها و فهمها.

لقد استخدمنا في مشروعنا هذا ثلاث مخططات وهي كالاتي:

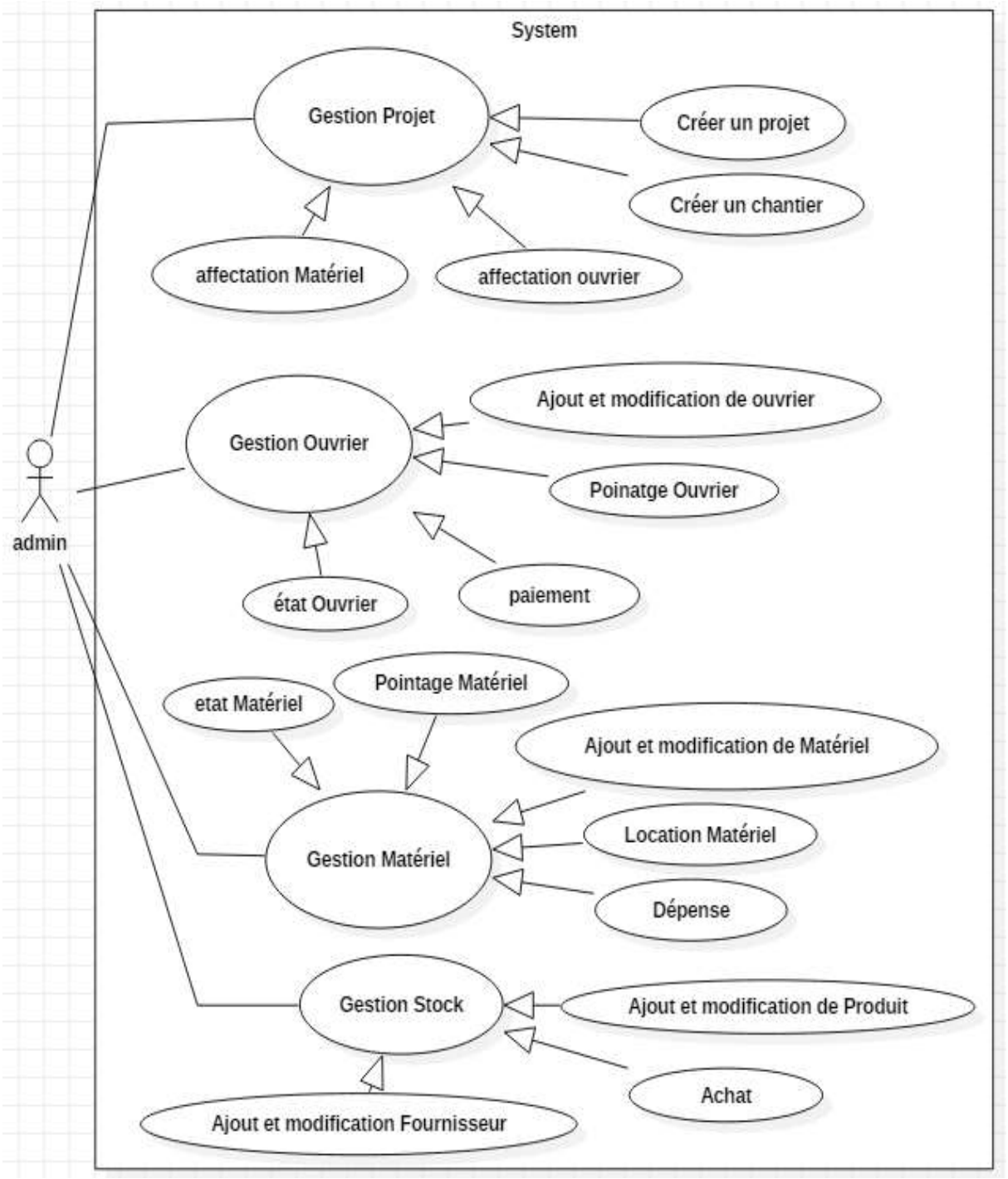
أ- مخطط حالة الاستخدام (Use Case):

هي وصف لخطوات أو إجراءات بين المستخدم، والنظام البرمجيات الذي يقود المستخدم إلى شيء مفيد، يمكن أن يكون المستخدم أو الممثل شخصا أو شيء أكثر تجريدا.

حالات الاستخدام هي تقنية لغة النمذجة التي تساعد مطوري البرمجيات على تحديد ملامح، لتنفيذ وحل الأخطاء بأمان .

سنقدم مخطط العام حالة الاستخدام لمشروعنا:

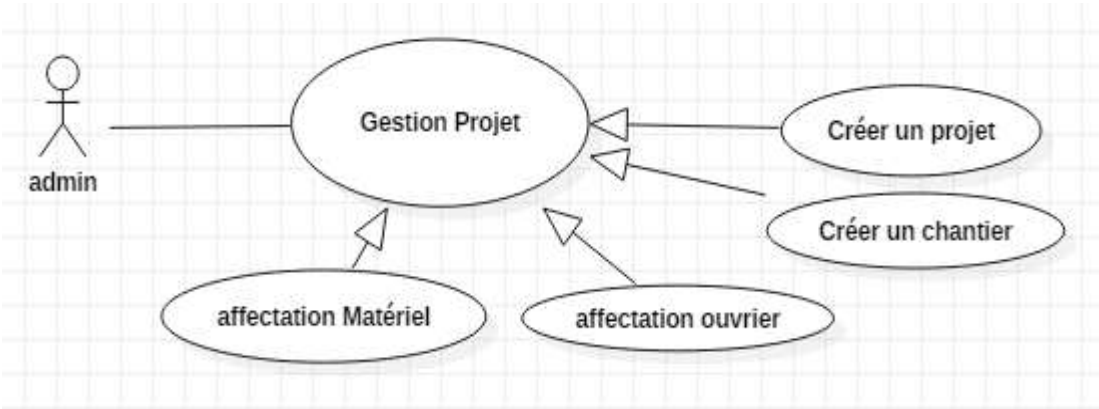
يوضح حالات استخدام كل من إدارة شؤون العمال وإدارة العتاد وإدارة المخزون.



الشكل 1: مخطط العام لحالة الاستخدام

- مخطط حالة الاستخدام " المشروع":

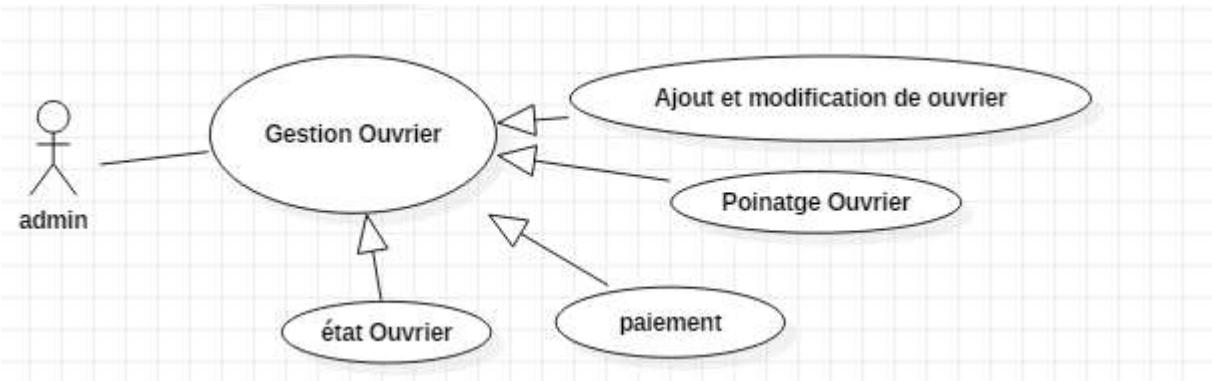
يستطيع عامل الإدارة بإنشاء مشروع وتحديد معلوماته الأساسية مثل (الزبون ومكان المشروع والمسؤول ....)، وإسناد إليه ورشات العمل التي بدورها تسند إليها عمال وعتاد.



الشكل 2: مخطط حالة استخدام للمشروع

- مخطط حالة الاستخدام " العمال":

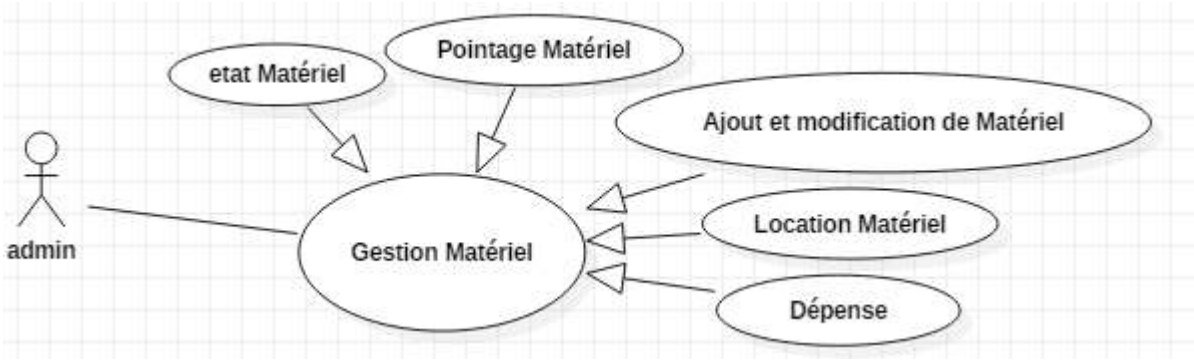
يستطيع عامل الإدارة متابعة العمال وتقييمهم واسنادهم للورشات وكل ما يخص رواتبهم كما يمكنه إيقافه عن العمل.



الشكل 3: مخطط حالة استخدام للعمال

- مخطط حالة الاستخدام " العتاد":

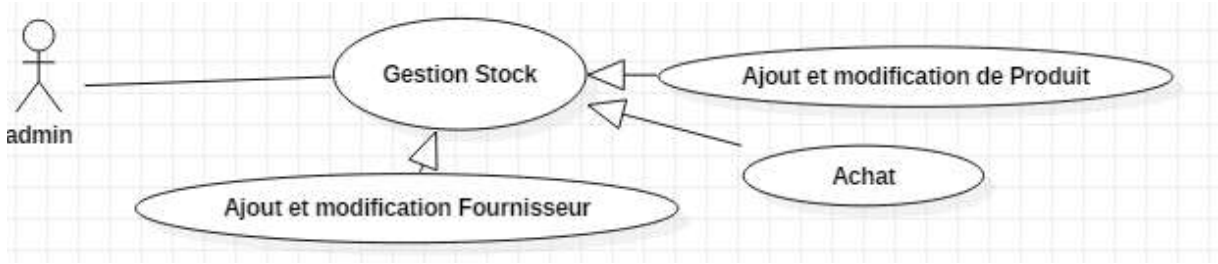
يستطيع عامل الإدارة متابعة العتاد وتقييمهم واسنادهم للورشات وكل ما يخص كرائهم.



الشكل 4: مخطط حالة استخدام للعتاد

- مخطط حالة الاستخدام " المخزن"

يستطيع عامل الإدارة من إضافة مواد وتعديل عليهم وكذلك المزودين.

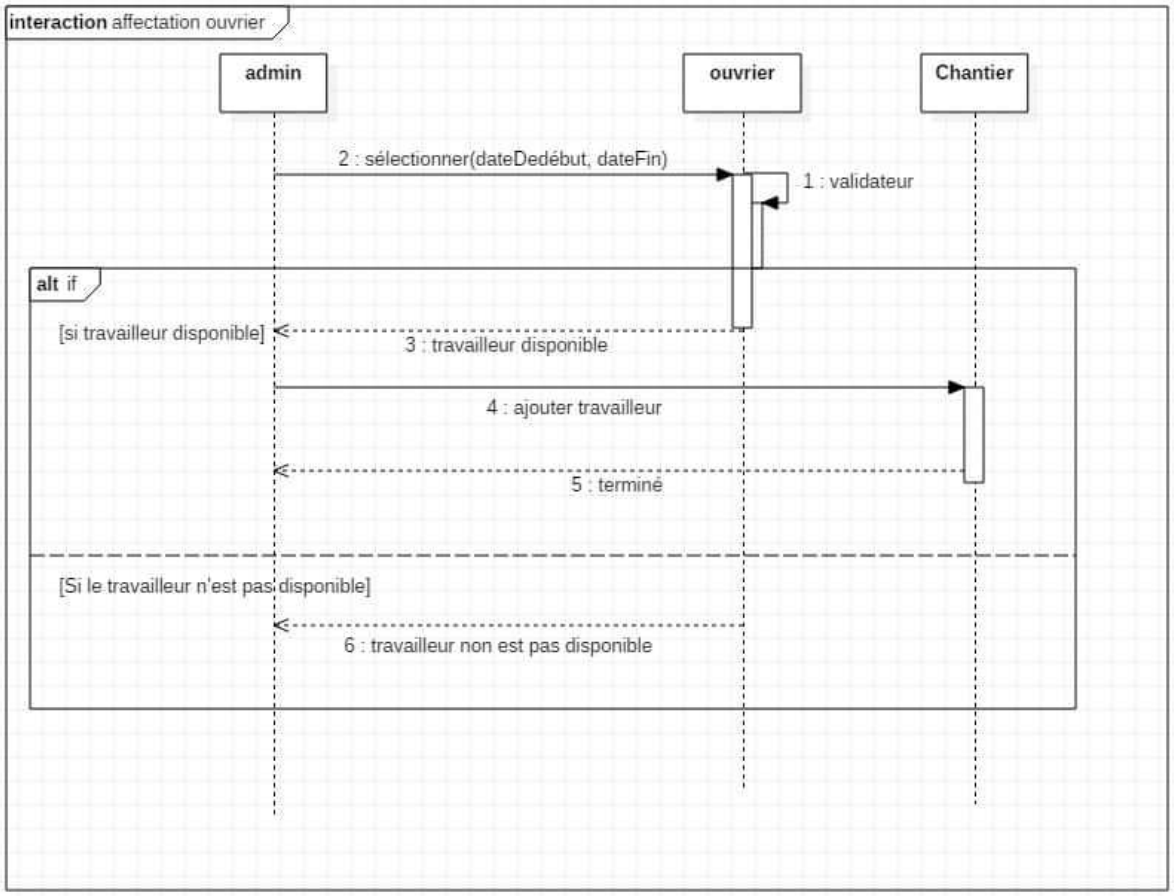


الشكل 5: مخطط حالة استخدام للمخزن

ب- مخطط التتابع (Diagramme de séquence):

هو مخطط يهدف إلى الاتصالات بين Object عبر الزمن أي يتم إدخال بعد الزمن، وبالتالي يتم توضيح كل التفاعلات والاتصالات بين Object وفق التسلسل الزمني.

ونوضح في هذا الشكل مخطط التتابع العام كالاتي:



يقوم المسؤول بتحديد فترة الإسناد والعامل ثم يتحقق النظام من إمكانية العملية إذا أمكن يقوم بإسناده إلى ورشة العمل.

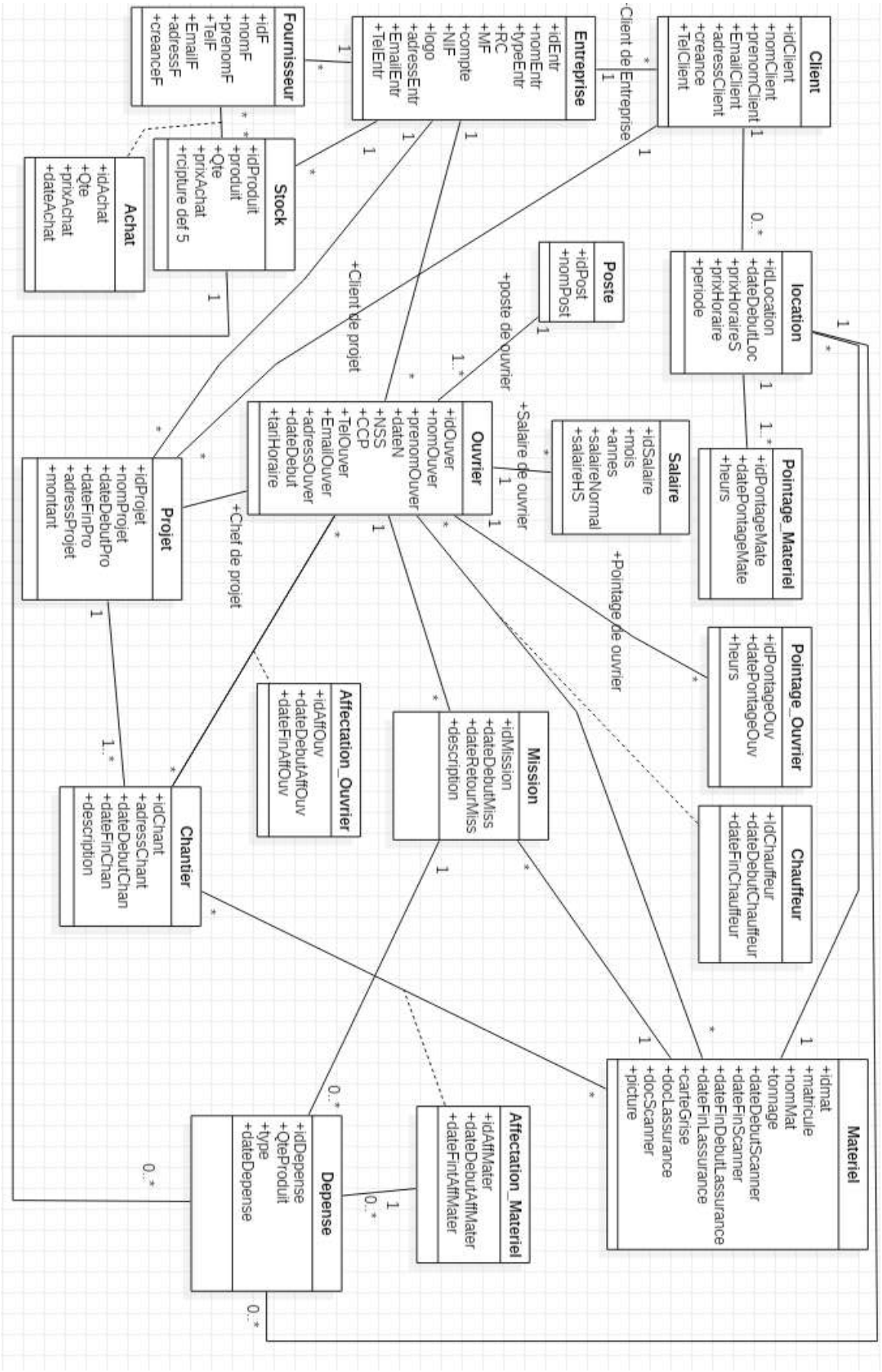
ت- مخطط الصنف (Diagramme de classes):

هو من أحد أنواع مخططات الفئة في لغة النمذجة الموحدة (UML) سهل الرسم نسبيا، هو نوع من ثابت هيكل الرسم التي تصف بنية النظام من خلال إظهار طبقات النظام، وخصائصها، والعمليات والعلاقات بين الكائنات.

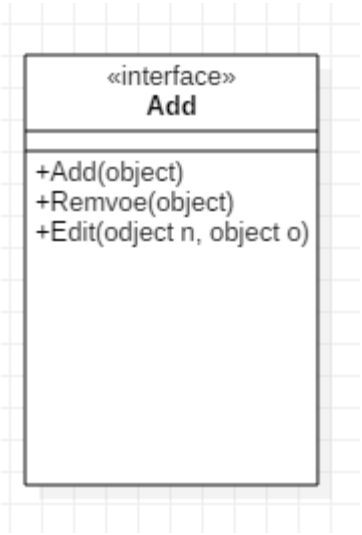
- **مخطط الصنف للمشروع:**

في هذا المخطط قدمنا صورة كاملة عن جميع الفئات المؤثرة، بحيث وضعنا علاقات بين مختلف هذه الفئات ونظرا إلى عدم إمكانية تبيان كل شيء في ورقة البحث توجد واجهة تحتوي على تعاريف دوال الإدخال والحذف وتعديل وتطبق على جميع الفصول وأيضا كل فصل يحتوي على دوال مميزة خاصة به مثل فئة العامل لديها إجراءات دفع.

ونوضح في هذا الشكل مخطط الصنف العام كالاتي:



الشكل 6: مخطط الصنف



تحتوي هذه الواجهة على ثلاث عمليات أساسية، حيث كل الفئات في الشكل السابق ترث نفس هذه العمليات.

- **النموذج العلائقي وقاموس البيانات:**
  - **النموذج العلائقي:**

- Entreprise (idEntr, nomEntr, typeEntr, RC, MF, compte, NIF, logo, adressEntr, EmailEntr, TelEntr);
- Client (idClient, nomClient, prenomClient, EmailClient, adressClient, creance, TelClient, #idEntr) ;
- Fournisseur (idF, nomF, prenomF, TelF, EmailF, adressF, creanceF, #idEntr) ;
- Stock (idProduit, produit, Qte, prixAchat, rcipture, #idEntr) ;
- Achat (idAchat, #idF, #idproduit, Qte, prixAchat, dateAchat) ;
- Projet (idProjet, nomProjet, dateDebutPro, dateFinPro, #client, adressProjet, #chefProjet, montant) ;
- Materiel (idmat, matricule, nomMat, tonnage, dateDebutScanner, dateFinScanner, dateFinDebutLassurance, dateFinLassurance, carteGrise, docLassurance, docScanner, picture) ;
- Poste (idPost, nomPost) ;
- Ouvrier (idOuver, nomOuver, prenomOuver, dateN, NSS, CCP, TelOuver, EmailOuver, adressOuver, #poste, dateDebut, tariHoraire, #idEntr) ;

- Pointage_Ouvrier (idPontageOuv, datePontageOuv, heurs, #idOuver) ;
- Salaire (idSalaire, mois, annes, salaireNormal, salaireHS, #idOuver) ;
- Chantier (idChant, adressChant, dateDebutChan, dateFinChan, description, #idProjet) ;
- Affectation_Ouvrier (idAffOuv, dateDebutAffOuv, dateFinAffOuv, #idChant, #idOuvr) ;
- Chauffeur (IdChauffeur, dateDebutChauffeur, dateFinChauffeur, #idmat, #idOuver) ;
- Mission (idMission, dateDebutMiss, dateRetourMiss, description, #idOuver, #idmat) ;
- Affectation_Materiel(idAffMater, dateDebutAffMater, dateFintAffMater, #idMat, # idChant) ;
- Pointage_Materiel (idPontageMate, datePontageMate, heurs, #idLocation) ;
- Depense (idDepense, QteProduit, dateDepense, type, #idLocation, #idMission, #idAff, #idProduit) ;

- **قاموس البيانات:**

هو عبارة عن مجموعة من المعلومات التي يديرها نظام قواعد البيانات (DB)، فاهو يساعد على معرفة ماهية البيانات (سلسلة رقمية، تاريخ، نص...) و أسماء محددة تابعة لها.

Les Classes	L'attribut	Signification	Type
Entreprise	idEntr	Numéro d'inscription de Entreprise	integer
	nomEntr	Nom d'entreprise	string
	typeEntr	Classe d'entreprise	string
	RC	Registre du commerce	string

	MF	Numéro de sécurité sociale	string
	compte	Le compte courant de l'entreprise	string
	NIF	Numéro d'identification fiscale	string
	logo	Logo d'entreprise	image
	EmailEntr	Email de l'entreprise	string
	TelEntr	Numéro de téléphone de l'entreprise	string
Client	idClient	Numéro d'inscription de client	integer
	nomClient	Nom de client	string
	prenomClient	Prénom de client	string
	EmailClient	Email de client	string
	adressClient	Address de client	string
	creance	Créance de client	decimal
	TelClient	Numéro de téléphone de client	string
Fournisseur	idF	Numéro d'inscription de Fournisseur	integer
	nomF	Nom de fournisseur	string
	prenomF	Prénom de fournisseur	string
	TelF	Numéro de téléphone de fournisseur	string
	EmailF	Email de fournisseur	string
	adressF	Address de fournisseur	string

	creanceF	Créance de fournisseur	decimal
Stock	idProduit	Numéro d'inscription de produit	integer
	produit	Nom de produit	string
	Qte	Quantité de produit	integer
	prixAchat	Prix d'achat de produit	decimal
	rcipture	pour notification	integer
Achat	idAchat	Numéro d'inscription de Achat	integer
	Qte	Quantité de produit	integer
	prixAchat	Prix d'achat de produit	decimal
	dateAchat	Date d'achat	date
Projet	idProjet	Numéro d'inscription de Projet	integer
	nomProjet	Nom de projet	string
	dateDebutPro	Date de début de projet	date
	dateFinPro	Date de Fin de projet	date
	adressProjet	Address de projet	string
	montant	Montant de projet	decimal
Materiel	idmat	Numéro d'inscription de Matériel	integer
	matricule	Matricule de matériel	string
	nomMat	Nom de matériel	string
	tonnage	Capacité de charge de matériel	integer

	dateDebutScanner	Date de début de scanner	date
	dateFinScanner	Date de Fin de scanner	date
	dateDebutLassurance	Date de Début l'assurance	date
	dateFinLassurance	Date de Fin l’assurance	date
	carteGrise	Carte de Grise	image
	docLassurance	Carte l'assurance	image
	docScanner	Carte scanner	image
	picture	Image de matériel	image
Ouvrier	idOuver	Numéro d'inscription de Ouvrier	integer
	nomOuver	Nom de ouvrier	string
	prenomOuver	Prénom de ouvrier	string
	dateN	Date de naissance	date
	NSS	Numéro de sécurité sociale	string
	CCP	Le compte courant de ouvrier	string
	TelOuver	Numéro de téléphone de ouvrier	string
	EmailOuver	Email de ouvrier	string
	adressOuver	Address de ouvrier	string
	dateDebut	Date de début	date
	tariHoraire	Tarif Horaire	decimal
	tariHoraireSup	Tarif Heures supplémentaires	decimal

Poste	idPost	Numéro d'inscription de Poste	integer
	nomPost	Nom de poste	string
location	idLocation	Numéro d'inscription de location	integer
	dateDebutLoc	Date de début de location	date
	prixHoraireS	Prix de Horaire supplémentaires	decimal
	prixHoraire	Prix de Horaire	decimal
	periode	Période par jour	integer
Pointage_Materiel	idPontageMate	Numéro d'inscription de Pointage Matériel	integer
	datePontageMate	Date de pointage matériel	date
	heurs	Nombre d'heures de travail	integer
Pointage_Ouvrier	idPontageOuv	Numéro d'inscription de Pointage Ouvrier	integer
	datePontageOuv	Date de pointage ouvrier	integer
	heurs	Nombre d'heures de travail	integer
Salaire	idSalaire	Numéro d'inscription de Salaire	integer
	mois	Mois de paie	integer
	annes	Année de paie	integer
	salaireNormal	Salaire de heures normal	decimal
	salaireHS	Salaire de heures supplémentaires	decimal
Chauffeur	IdChauffeur	Numéro d'inscription de Chauffeur	integer

	dateDebutChauffeur	Date de début chauffeur	date
	dateFinChauffeur	Date de fin chauffer	date
Chantier	idChant	Numéro d'inscription de Chantier	integer
	adressChant	Address de chantier	String
	dateDebutChan	Date de début chantier	date
	dateFinChan	Date de fin chantier	date
	description	Description de chantier	string
Affectation_Ouvrier	idAffOuv	Numéro d'inscription d'affectation de ouvrier	integer
	dateDebutAffOuv	Date de début d'affectation de ouvrier	date
	dateFinAffOuv	Date de fin d'affectation de ouvrier	date
Affectation_Materiel	idAffMater	Numéro d'inscription d'affectation de Matériel	integer
	dateDebutAffMater	Date de début d'affectation de Matériel	date
	dateFintAffMater	Date de fin d'affectation de Matériel	date
Mission	idMission	Numéro d'inscription de mission	integer
	dateDebutMiss	Date de début de mission	date
	dateRetourMiss	Date de retour de mission	date
	description	Description de mission	date
Depense	idDepense	Numéro d'inscription de Dépense	integer

	QteProduit	Quantité de produit utilisé	integer
	type	Type de dépense	integer
	dateDepense	Date de dépense	date

جدول قاموس البيانات

الخاتمة:

حاولنا في هذا الفصل تقديم موجز كافي لشرح طريقة عمل النظام، وهذا بعد انجاز المخططات اللازمة (مخطط حالات الاستخدام، مخطط التتابع) النموذج التصوري للمعطيات (مخطط الفئات) والذي من خلالها أصبح لدينا تصور كافي عن التطبيق الجاري إنجازه. بوصولنا لهذه المرحلة لم يبقى لنا سوى الشطر الأخير الا وهو الدراسة التقنية والتي من خلالها نصل بحول الله الى تحقيق الهدف وانهاء دراستنا ان شاء لله.

الفصل الثالث:
التنفيذ

تـمـهـيـد :

في هذا الفصل ترجمنا مشروعنا إلى برنامج سطح مكتب، يحاكي عملنا النظري، باستعمال لغة C# مع إطار عمل windows forms dotnet، وبمساعدة بيئة العمل Visual Studio 2019، وأنشأن قاعدة بيانات وادرناه باستعمال SQL server 2017

III.1 الأدوات الـتـنـفـيـذ:

III.1.1 لـغـة C#:

هي لغة برمجة متعددة الأنماط تتمتع بكونها سكونية التنميط وأمرية وتعريفية ووظيفية وإجرائية وعمومية وشيئية المنحى (غرضية التوجه) (باستخدام الصفوف) كما تخضع لمبادئ البرمجة التركيبية المنحى.

قامت مايكروسوفت بتطوير هذه اللغة في إطار عملها على تطوير دوت نِت وتمت الموافقة على تعييرها من منظمة Ecma (المعيار Ecma-334) والمنظمة الدولية للمعايير (المعيار ISO/IEC 23270:2006). إن سي شارب إحدى لغات البرمجة المصممة للعمل على البنية التحتية المشتركة للغات البرمجة (CLI). صُممت لغة سي شارب لتكون لغة بسيطة وحديثة وعامة الأغراض وشيئية المنحى. وقد قاد أندرس هيلسبرغ فريق تطويرها.

III.1.2 تـاريـخ لـغـة C#:

بدأ تطوير منصة دوت نت بكتابة مجموعة من مكتبات الصفوف، وقد استخدم نظام تصريف مدار اسمه Simple Managed C أو اختصاراً SMC للقيام بذلك. لاحقاً وبالتحديد في كانون الثاني 1999 شكل أندرس هيلسبرغ فريقاً من المطورين بهدف بناء لغة جديدة اسمها كول) بالإنجليزية(Cool :، يشكل الاسم اختصاراً لعبارة "لغة غرضية التوجه شبيهة بلغة) "C بالإنجليزية C-like Object Oriented : (Language. قررت مايكروسوفت الإبقاء على هذا الاسم إلا أنها تخلت عن ذلك لاحقاً لأسباب قانونية لها علاقة بحقوق العلامات المسجلة. على التوازي مع ذلك أُعلن مشروع دوت نت رسمياً في مؤتمر للمطورين المحترفين (PDC) في تموز عام 2000 وأعيد تسمية اللغة إلى سي# كما تم تصدير وقت التنفيذ الخاص بلغة إيه إس بي دوت نت بالإضافة إلى مكتبات الصفوف إلى هذه اللغة.

اعتبر مصمم لغة جافا جيمس غوسلينغ وبيل جوي وهو أحد مؤسسي شركة صن مايكروسيستمز التي أتت بلغة جافا أن لغة سي# ليست سوى "تقليداً" للغة جافا؛ وقد

قال غوسلينغ معقباً "إنها [المقصود سي#] كجافا نوعاً ما ولكن بعد التخلي عن الاعتمادية والإنتاجية والأمان". كتب كل من كلاوس كرفت وأنجلكا لانجر في مقال لهما في مدونة "إن جافا وسي# لغتا برمجة متطابقتان تقريباً. وهذا تكرار مضجر يفتقر الإبداع"، "من الصعب جداً الادعاء أن جافا أو سي# لغة برمجة ثورية غيرت الأسلوب الذي نكتب فيه البرامج"، "لقد استعارت سي# الكثير من جافا -والعكس صحيح. حيث تدعم سي# ميزة التعليب وفك التعليب الآن وقريباً سوف نجد ميزةً شبيهة في جافا". قال أندرس هيلسبرغ في تموز عام 2000 أن سي# ليست "نسخة من جافا" بل أنها "أكثر قرباً إلى لغة سي++" من ناحية التصميم.

في تشرين الثاني 2005 أعلن عن الإصدار 2.0 من سي# ومن هنا بدأت سي# وجافا بالتطور في اتجاهات متزايدة الاختلاف. إن أول وأهم هذه الاختلافات كان في إضافة الأنماط العمومية) بالإنجليزية(Generics : إلى كلتا اللغتين حيث كان تحقيقهما لهذه الأنماط شديد الاختلاف حيث تقوم سي# بالتعامل مع الأنماط العمومية كصفوف حقيقية وتولد الكود الخاص بها وقت التنفيذ بينما تتعامل جافا مع هذه الأنماط كميزة مضافة إلى نحو اللغة تمكن المطور من كتابة كود عمومي كما تمكن المترجم من التأكد من صحة الأنماط فقط بينما لا تحول هذه الأنماط إلى أنماط حقيقية وقت التنفيذ ولا يتم توليد كود خاص بها على غرار سي.#

إضافة إلى ذلك فقد أضيفت إلى سي# مجموعة من الميزات الهامة بهدف تمكين استخدام البرمجة الوظيفية فيها كُللت بإضافة لينك في الإصدار 3.0 والإطار البرمجي الداعم لتعابير لامبدا والطرق الملحقة والأنماط غير المسماة. تمكن هذه الميزات المطور من استخدام تقنيات البرمجة الوظيفية عندما يكون من المستحسن القيام بذلك. إن إضافات لينك وغيرها من الميزات الوظيفية تساعد المطور على كتابة أسطر أقل عند القيام بمهام روتينية كالاستعلام من قاعدة بيانات أو إعراب ملف إكس إم إل أو البحث ضمن بنية معطيات بما يمكن من التركيز على هدف البرنامج المنطقي وتحسين مقروئيته وصيانته.

كان لدى سي# جالب للحظ اسمه آندي (سمي باسم أندرس هيلسبرغ) وقد أحيل إلى التقاعد في 29 كانون الثاني عام 2004.

عُرضت سي# على لجنة آيزو الفرعية JTC 1/SC 22 للمراجعة والتعيير، كان اسم المعيار ISO/IEC 23270:2003 وهو ملغى اليوم. تمت الموافقة فيما بعد على تعيير سي# وفق المعيار ISO/IEC 23270:2006.

III.1.3 ميزات لغة C#:

تتفرد لغة سي# عن غيرها من لغات برمجة إطار عمل دوت نت في كونها مرتبطة بشكل وثيق بمزايا البنية التحتية المشتركة للغات البرمجة (CLI)، فعلى سبيل المثال

تنسب معظم الأنماط الحقيقية إلى قرائنها من أنماط القيمة (بالإنجليزية: value-types) الموجودة في CLI. على الرغم من ذلك فإن توصيف اللغة لا يتطرق إلى متطلبات توليد الكود الخاصة بالمصرّف، بمعنى أن التوصيف لا يرغم مصرف سي# على توليد كود متوافق مع وقت التنفيذ المشترك للغات (CLR) أو توليد تعليمات متوافقة مع اللغة المشتركة الوسيطة (بالإنجليزية: Common Intermediate Language) أو أي تنسيق آخر. وهكذا يمكّن (نظرياً على الأقل) مصرّف سي# من توليد كود لغة الآلة مباشرة على غرار باقي المصرفات كمصرّف سي++ ومصرّف فورتران. فيما يلي بعض من المزايا الجديرة بالاهتمام كونها تميز لغة سي# عن لغتي سي وسي++ (بالإضافة إلى جافا لدى ذكر ذلك صراحة):

- تدعم سي# التصريح الضمني عن المتحولات قوية التنميط عن طريق الكلمة المفتاحية var بالإضافة إلى التصريح الضمني عن المصفوفات المنمطة عن طريق الكلمة المفتاحية متبوعة بمهيئ المجموعة.
- إن إمكانية الميتا-برمجة) بالإنجليزية(Meta Programming : من خلال استخدام السمات) بالإنجليزية(Attributes : هي جزءً من توصيف اللغة. توفر العديد من هذه السمات وظيفية مطابقة لتلك التي توفرها موجهات قبل التصريف المعتمدة على المنصة المدعومة من قبل جي.سي.سي وفيجوال سي.++
- يجب على مبرمجي سي# استخدام الكلمة المفتاحية virtual من أجل إتاحة تجاوز الطرق في الصفوف المشتقة على غرار لغة سي++ (وخلافاً للغة جافا).
- تتيح الدوال الملحقة) بالإنجليزية(Extension Methods : لمبرمجي سي# إمكانية استخدام الطرق الاستاتيكية كما لو أنها طرق خاصة بالصف، وهذا ما يمكن المبرمج من إضافة طرق جديدة إلى الصفوف عن الإحساس بضرورة وجودها في الصف نفسه وكافة الصفوف المشتقة منه.
- يتيح النمط dynamic إمكانية الربط مع الطرق أثناء التنفيذ بما يمكن من استدعاء الطرق وتركيب الكائنات وقت التنفيذ مثلما تفعل لغة جافا سكربت.
- لدى سي# الكلمة المفتاحية delegate التي تتيح تعريف مؤشرات قوية التنميط إلى توابع.
- لدى سي# وظيفية الإعلان عن الأحداث والتسجيل عليها من خلال استخدام delegateوغيرها بشكل يشابه وظيفية الإشارات والمقابس) بالإنجليزية : signals and slots) التي تضيفها بيئة تطوير كيوت إلى لغة سي.++
- خلافاً للغتي سي وسي++ لا تستطيع المتحولات المحلية إخفاء المتحولات الموجودة في الكتلة البرمجية المحيطة.

- يوفر فضاء التسمية namespace في لغة سي# نفس المستوى من العزل الذي توفره رزمة package جافا وفضاء التسمية namespace سي++، كما أنه يتمتع بالعديد من القواعد والميزات الشبيهة برزمة جافا.
- تتيح سي# إمكانية استخدام المؤشرات إلى عناوين الذاكرة فقط ضمن كتل مشار إليها بالكلمة المفتاحية unsafe (غير آمن)، وتتطلب البرامج التي تحتوي كوداً غير آمنٍ سماحيات ملائمة كي تعمل. غالباً ما تستخدم الدلائل) بالإنجليزية : (References الآمنة من أجل الولوج إلى الكائنات، حيث تشير هذه الدلائل دائماً لي كائنات فعالة في الذاكرة كما أن لديها قيمة فارغة null معرفة بشكل ملائم؛ من المستحيل الحصول على دليل إن كائن تم تدميره (من خلال جامع القمامة) أو الولوج إلى كتلة عشوائية في الذاكرة. يمكن للمؤشر غير الآمن الإشارة إلى متحول من نمط-قيمة أو مصفوفة أو سلسلة محرفية أو كتلة من الذاكرة محجوزة على المكدس. يمكن للكود غير الآمن أن يخزن ويتلاعب بالمؤشرات من خلال النمط System.IntPtr ولكن لا يمكنه الاطلاع على محتوياتها.
- لا يمكن تحرير الذاكرة المُدارة بشكل مباشر إذ يتم تحريرها بشكل أوتوماتيكي من قبل جامع القمامة. إن جمع القمامة يعالج معضلة تسرب الذاكرة من خلال إعفاء المبرمج من مسؤولية تحرير الذاكرة.
- توفر لغة سي# الخصائص) بالإنجليزية(Properties : كتجميل لغوي لنمط برمجي شائع يتم فيه تعريف زوج من الطرق: طريقة للحصول على القيمة) بالإنجليزية : (getter وأخرى لتعديل القيمة) بالإنجليزية(setter : بهدف تغليف عمليات الوصول إلى عضو من أعضاء الصف. نتيجة لذلك لم تعد هنالك حاجة لكتابة العديد من هذه الطرق بشكل تكراري، بالإضافة إلى ذلك يمكن الولوج إلى الخصائص بشكل موجز بنفس طريقة الولوج إلى الأعضاء ولا حاجة لاستخدام استدعاء الطريقة المعتاد للقيام بذلك.
- ابتداءً من الإصدار 3.0 تدعم لغة سي# تقنيات البرمجة الوظيفية من خلال الصفوف التابعية وتعابير لامبدا على الرغم من كونها لغة أمرية أصلاً.

2.III إطار العمل windows Forms:

ويندوز فورمز هي تقنية لإنشاء برامج ذات واجهات رسومية للمستخدم تعتمد على الأزرار وأدوات الإدخال مثل الـ(صندوق النص) (textbox) وتتميز بالألوان والصور وسهولة التعامل مع البرامج من نوعيتها التقنية من انتاج شركة مايكروسوفت بديلا للتعامل مع الكونسول أو الـ موجه الأوامر، أو التي تعرف بالشاشة السوداء وهي واجهة نظام التشغيل القديم نظام الدوس (Dos OS)

III.3 SQl Server : ميكروسوفت إس كيو إل سيرفر (بالإنجليزية: Microsoft SQL Server) هو برنامج لقواعد البيانات العلائقية من إنتاج مايكروسوفت، لغة الاستعلام الرئيسية فيه هي إس كيو إل وتي-سكيول.

III.4 ***بعض نوافذ التطبيق:***

مقدمة :

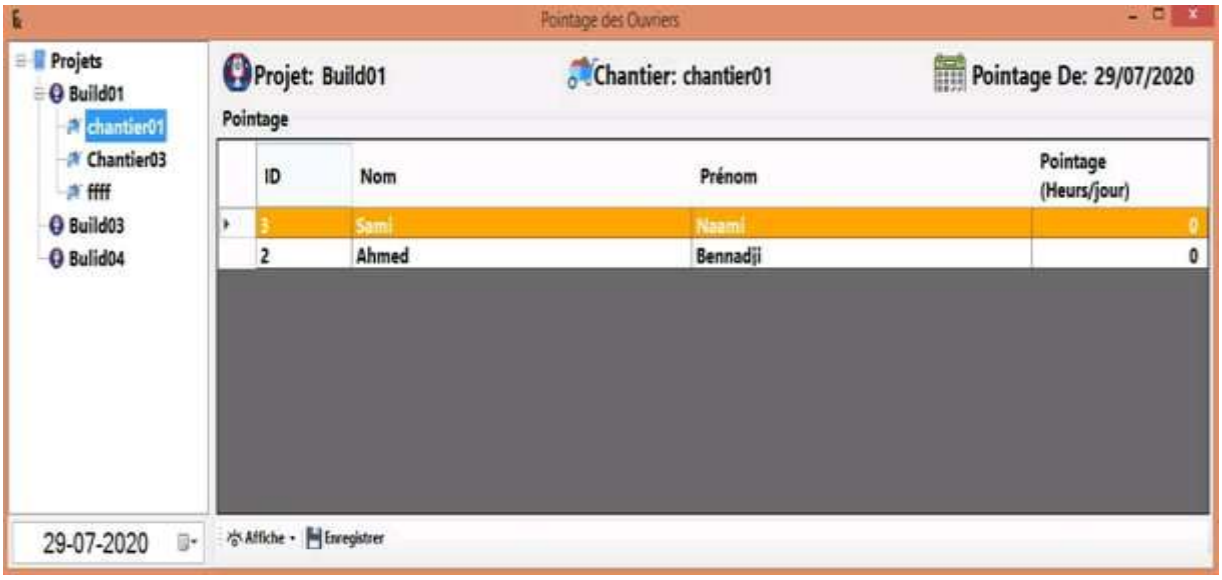
سنعرض في هذا الشطر بعض الواجهات للتوضيح كنموذج بما أن البرنامج كبير والعدد الهائل للنوافذ لا نستطيع عرضها كلها في هذا الصدد لذلك اخترنا بعضها والبقية سنعرضها أثناء الإلقاء.

- واجهة البرنامج الرئيسية:

الشكل 1: واجهة الرئيسية

❖ تحتوي النافذة الرئيسية على القوائم الرئيسية في البرنامج حيث كل قائمة يندرج تحتها أدوات تساعد في العمل على برنامج مثل قائمة عمال تحتوي على كل ما يخص العمال.

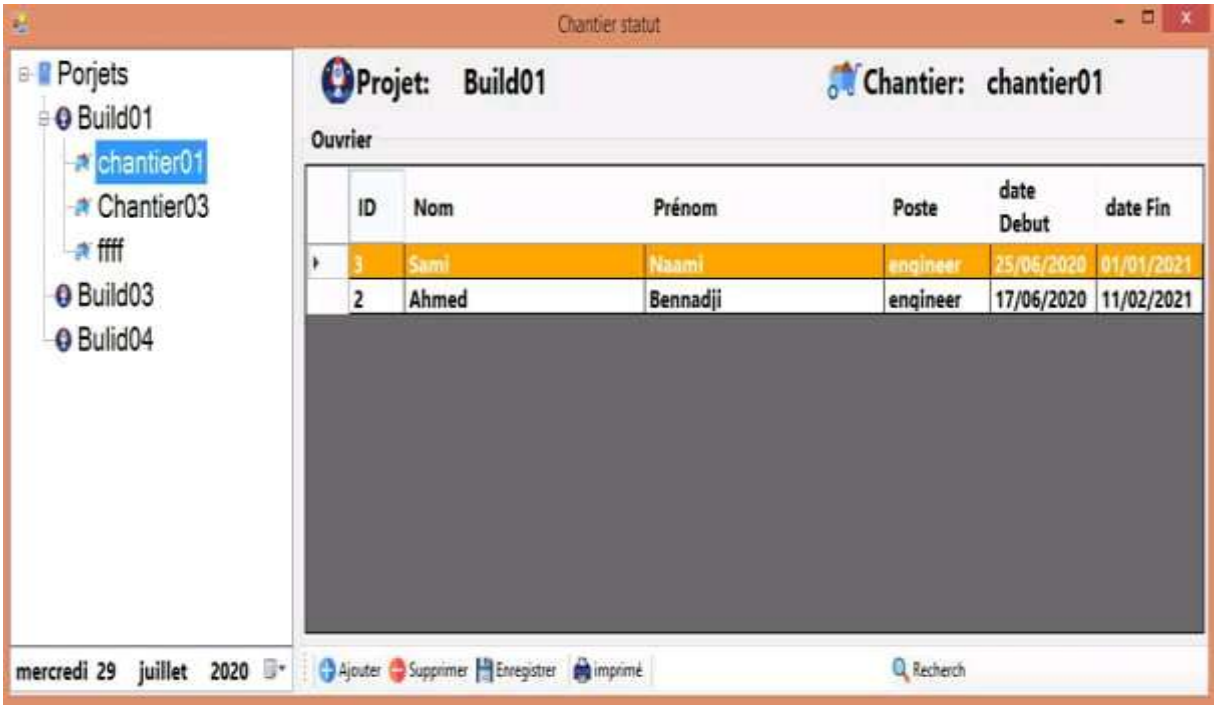
- واجهة التقييم اليومي لعمال الورشات:



الشكل 2: واجهة التقييم اليومي لعمال الورشات

❖ هذه الواجهة تسمح لنا بمتابعة وتقييم اليومي للعمال داخل ورشات العمل.

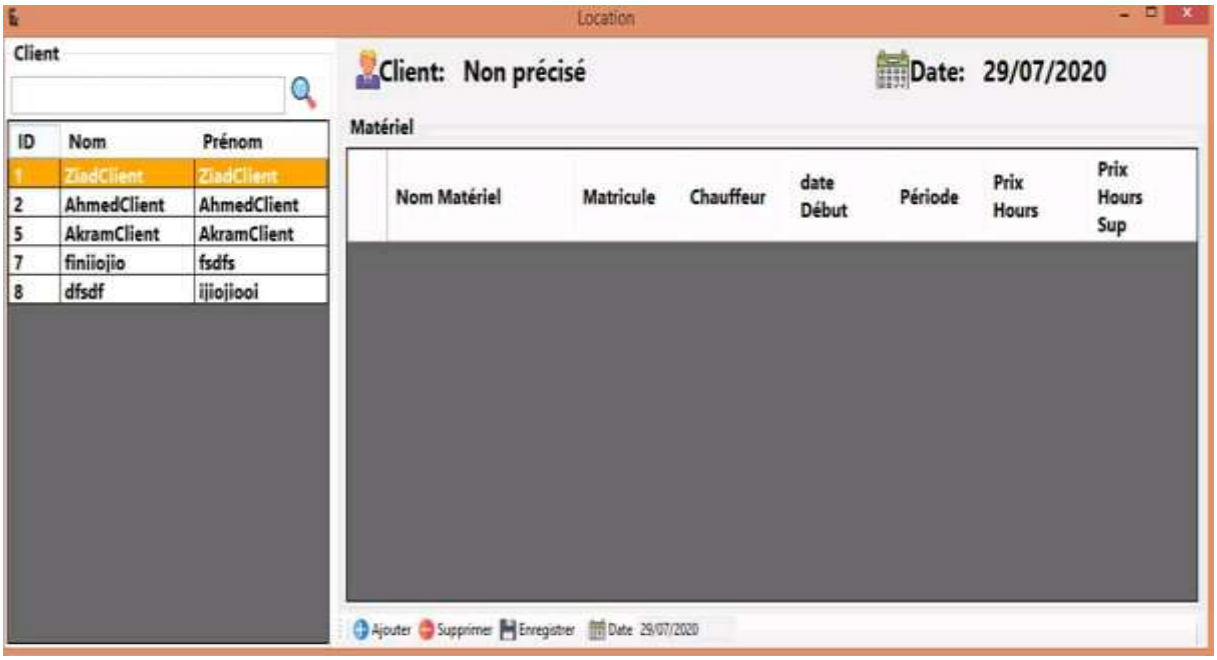
- واجهة إسناد العمال ومتابعتهم:



الشكل 3: واجهة إسناد العمال ومتابعتهم

❖ تسمح لنا هذه الواجهة بإسناد العمال لورشات وطبع قائمة العمال في الورشة.

- واجهة كراء العتاد للزبائن.



الشكل 4: واجهة كراء العتاد للزبائن

❖ تسمح لنا هذه الواجهة بكراء العتاد للزبائن المسجلين لدينا مسبقا ومتابعة العتاد لديهم.

- واجهة الدفع.

الشكل 5: واجهة الدفع

- تسمح لنا هذه الواجهة بمتابعة راتب العمال وعدد ساعات عملهم خلال الشهر في ورشة محددة ومعرفة كم كلفت الورشة الشركة.

III.5 الـخلاصة :

في هذا الفصل قدمنا البيئة المستخدمة في تطور مشروعنا، ومن ثما قمنا بتقديم بعض الواجهات الأساسية في برنامجنا.

خاتمة عامة

في عملنا هذا الذي قمنا بتحقيقه، حاولنا تطوير وسيلة من أجل تسهيل تسيير شؤون العمال والعتاد لشركة المقاولات من خلال إنشاء برنامج سطح مكتب مع قاعدة بيانات تجعل أمر الإدارة للإداريين بسيط بحيث يوفر البرنامج واجهات سهلة وغير معقدة، ولقد قمنا بإنجاز الحد الأدنى من المشروع بسبب حجم كبير له وبسبب ضيق الوقت لذلك نأمل أن نجد فرصة سامحة لتطويره أكثر وأكثر.

- **أهم الصعوبات المواجهة:**

✓ تدهور الأوضاع بسبب جائحة كورونا مما جعلنا ننقطع على الأستاذ المشرف.

✓ ضعف سرعة شبكة الأنترنت في بلادنا الذي جعلنا ننتظر لساعات طويلة في البحث عن حل للمشاكل التي صادفتنا.

✓ قلة المصادر العلمية بسبب غلق الجامعة.

في الخاتمة نتمنى من الله عز وجل أن يوفقنا ويسدد خطانا في دربنا، وأن نكون قد وفقنا في إضافة شيء ولو بسيط لهذا العالم، كما نأمل أن يسركم عملنا

قـائـمـة الـمـصـادر و الـمـراجـع :

https://en.wikipedia.org/wiki/C_Sharp_(programming_language)

https://en.wikipedia.org/wiki/Microsoft_SQL_Server

https://www.uml-diagrams.org/

https://visualstudio.microsoft.com/

https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15

https://en.wikipedia.org/wiki/Class_diagram

https://en.wikipedia.org/wiki/Use_case

https://en.wikipedia.org/wiki/Sequence_diagram

- كتاب "لغة النمذجة الموحدة "" UML" إعداد الطالب "محمد أحمد سالم الوصابي"قسم التكنولوجيا المعلومات.
- C#7.0 in a Nutshell by *Joseph Albahari and Ben Albahari*
- Database Modeling& Design **Fourth Edition** by Toby Teorey Sam Lightstone Tom Nadeau.
- **Agile Principles, Patterns, and Practices in C#** By Martin C. Robert, Martin Micah.

,