الجمهورية العربية السورية
المعهد العالي للعلوم التطبيقية والتكنولوجيا
قسم النظم المعلوماتية – هندسة البرمجيات والذكاء الصنعي
العام الدراسي 2024/2023

مشروع سنة الرابعة

إدارة حالة مشاريع المعهد العالي

تقديم

حسن بهجت خضور

إشراف

د. مصطفى دقاق م. محمود الياس

8/10/201

الاهداء.

الخلاصة

(وتتضمن الهدف من المشروع) يهدف هذا المشروع إلى .

Abstract

Translate your abstract here.

المحتويات

قائمة الأشكال ..... **V**
الفصل الأول **1**
التعريف بالمشروع **1**
1.1- مقدمة 2
2.1- الهدف من المشروع 2
3.1- نطاق المشروع 2
4.1- المتطلبات الوظيفية 2
5.1- المتطلبات غير الوظيفية 5
الفصل الثاني **6**
الدراسة المرجعية **6**
1.2- مقدمة **7**
2.2 الحاجة إلى النظام 7
3.2 النظام الحالي ومشاكله / إن وجد 7
3.2 القيمة المضافة التي سيقدمها 7
الفصل الثالث **8**
دراسة آلية إدارة المشاريع **8**
1.2- مقدمة 9
1.2- دورة حياة المشروع (النشاط) 9
2.3 آلية إدارة المشروع المتبعة في المعهد 11
الفصل الرابع **13**
خطة إدارة المشروع **13**
1.4- مقدمة 14
2.4- الإجرائية المتبعة في تطوير المشروع 14
3.4- الخطة الزمنية 15
4.4- مخطط غانت Gantt Chart 16

الفصل الخامس **17**
الدراسة التحليلية **17**
1.6- مقدمة 18
2.6 مخطط أطوار المشروع 18
3.6- مدخلات ومخرجات النظام 19
4.6 مخططات حالات الاستخدام 20
6.6 مخططات التتالي لبعض حالات الاستخدام 39
7.6 عقود عمليات حالة الاستخدام متابعة مشروع 41
8.6 مخطط الصفوف 43
الفصل السابع **44**
الدراسة التصميمية **44**
1.7- مقدمة 45
2.7 البنية المعمارية النظيفة – Clean Architecture 45
6.7 نمط التصميم المستودع Repository Pattern 48
7.7 نمط التصميم Result Pattern 49
الفصل الثامن **50**
تصميم النظام **50**
1.8 مقدمة ..... **Error! Bookmark not defined.**
2.8 البنية المقترحة لتصميم النظام ..... **Error! Bookmark not defined.**
2.8مخطط العلاقات بين الكائنات Entity Relationship Diagram 52
3.8 تصميم قاعدة المعطيات ..... **Error! Bookmark not defined.**
الفصل التاسع **53**
الأدوات وأطر العمل المستخدمة **53**
1.1 إطار العمل Net Core. 54
2.2 إطار العمل Angular 54
3.2 قاعدة المعطيات SQL Server 54
4.2 نمط الخيارات Option Pattern 54
5.2 JSON Web Token 54

5.2المكتبة MediatR 55
الفصل العاشر **56**
تنجيز النظام **56**
1.5 مقدمة **57**
الفصل الحادي عشر **85**
الاختبارات **85**
1.6 مقدمة **86**

قائمة الأشكال

الشكل 1 : مخطط غانت لأنشطة خطة التنفيذ..... 16
الشكل 2 : مخطط أطوار المشروع..... 18
الشكل 3 : مخطط حالات الاستخدام لرئيس فريق العمل..... 20
الشكل 4 : مخطط حالات الاستخدام لمدير المشروع. 26
الشكل 5 : مخطط حالات الاستخدام للوكيل العلمي. 30
الشكل 6 : مخطط حالات الاستخدام للموظف..... 33
الشكل 7 : مخطط حالات الاستخدام للمستخدم. 35
الشكل 8 : مخطط حالات الستخدام لرئيس الفعالية..... 36
الشكل 9 : العلاقات بين الفاعلين 38
الشكل 10 : مخطط التتالي لحالة الاستخدام متابعة مشروع..... 39
الشكل 11 : مخطط التتالي لحالة الستخدام تخطيط مشروع. 40
الشكل 12 : مخطط الصفوف..... 43
الشكل 13 : مكونات البنية المعمارية النظيفة..... 45
الشكل 14 : مثال توضيحي عن النمط الوسيط 47
الشكل 15 : مثال توضيحي عن نمط الحالة..... 47
الشكل 16 : مخطط صفوف تجريدي يصف نمط الحالة..... 48
الشكل 17 : مخطط العلاقات بين الكياناتERD..... **Error! Bookmark not defined.**
الشكل 18 : مخطط قاعدة المعطيات..... **Error! Bookmark not defined.**

مقدمة عامة

مقدمة عن إدارة المشاريع وأهميتها في دم اتخاذ القرارات.

الفصل الأول

التعريف بالمشروع

نبيّن في هذا الفصل هدف المشروع ونطتقه كما نورد المتطالبات الوظيفية وغير الوظيفية.

1.1- مقدمة

مقدمة.

2.1- الهدف من المشروع

يهدف هذا المشروع إلى بناء نظام معلوماتي ندير من خلاله المشاريع ضمن المعهد العالي حيث يقدم هذا النظام إمكانية طرح مشروع وارفاق المستندات الخاصة بالمشروع واسناده للفعالية المنفذة ومتابعته.كما يمكن مديري المشاريع من تعريف مشروع وإضافة خطة الانفاق عليه و معلومات التمويل والتصنيف الخاصة بالمشروع , ويمكن روؤساء فرق العمل من إضافة مراحل المشروع , كما يمكنهم من إدارة عمليات المتابعة للمشاريع القائمين عليها من خلال متابعة نسب التنفيذ والتأخير ومدى انشغالية العاملين ومساهمتهم. ويؤمن هذا النظام إمكانية إصدار التقارير والاستمارات الخاصة بالمشاريع للإداريين المعنيين, كما سيسمح للموظفين بالاطلاع على المشاريع المشاركين بها.

.....

3.1- نطاق المشروع

ماهو نطاق المشروع وماهي الانظمة التي يتفاعل معها.

4.1- المتطلبات الوظيفية

نبين هنا المتطلبات الوظيفية التي يجب على النظام أن يحققها .

يجب على النظام أن يحقق مايلي

أ. أن يسمح للموظف بما يلي :

1. الاطلاع على المشاريع التي يشارك بها وأدواره بها.
2. إعلامه بالتبدلات على أدواره ومساهماته في المشاريع التي يشارك بها.
3. الاطلاع على نسبة انشغاليته في المشاريع التي يشارك به

ب. أن يسمح لرئيس فريق العمل بما يلي :

1. القيام بعمليات المتابعة للمشاريع التي يرأسها أي إدخال نسب الانجاز للمراحل وانشغالية العاملين (كما هو مبين في استمارة الانشغالية).
2. إصدار تقارير المتابعة للمشاريع (تقرير متابعة مرحلة, تقرير انشغالية العاملين).
3. استعراض عمليات المتابعة للمشاريع.
4. إدارة المشاركين في المشروع (إضافة مشارك, تعديل معلومات مشاركة,إزالة مشارك من المشروع).
5. إدارة مراحل المشروع (إضافة مرحلة ,تعديل معلومات مرحلة,إلغاء مرحلة).
6.

ت. أن يسمح لمدير المشروع بما يلي :

1. إدارة خطة الانفاق لمشروع.
2. إضافة المعلومات المالية للمشروع.
3. إضافة تصنيف المشروع .
4. إصدار تقرير عن حالة المشروع.
5. إصدار تقرير انجاز مشروع
6. إدارة المشاركين في المشروع

ث. أن يسمح لمراقبي المشاريع بما يلي :

1. إصدار تقارير المتابعة.
2. إصدار تقارير حالة المشاريع .
3. إصدار تقارير الخطة الزمنية.
4.

ج. أن يسمح لأعضاء المجلس العلمي المعنيين (الوكيل العلمي) :

1. استعراض المشاريع المتأخرة والمتوقفة.

2. إلغاء مشروع.
3. قبول مشروع.
4. طرح مشروع

ح. أن يسمح لرؤساء الفعاليات بما يلي :

1. استعراض المشاريع ضمن أقسامهم والاطلاع على حالتها.

خ. أن يسمح لجميع المستخدمين

1. إدارة حساباتهم .
2.

5.1- المتطلبات غير الوظيفية

نورد هنا المتطلبات غير الوظيفية أي القيود على النظم لكي يتم قبوله.حيث قمنا بتقسيمها إلى عدة محاور وهي متطلبات الأمان ومتطلبات الأداء ومتطلبات خاصة ومتطلبات التشغيل.

متطلبات الأمان

1. يجب أن يكون الدخول إلى النظام آمناً ؛ أي يجب أن يسمح فقط للمستخدمين المسجلين بالدخول إليه.
2. أن يسمح للمستخدمين القيام بالعليات وفقاً لما هم مخولون به من صلاحيات فقط.

متطلبات الأداء

1. أن

متطلبات خاصة

1. أن تكون الواجهات باللغة العربية.

متطلبات التشغيل

1. أن يتم تطوير تطبيق المخدم باستخدام إطار العمل ASP.Net وهو قيد من الطرف الذي سيشغل النظام.
2. أن يتم تطوير الواجهة الأمامية باستخدام إطار العمل Angular وهو قيد من الطرف الذي سيشغل النظام.
3.

الفصل الثاني

الدراسة المرجعية

نورد في هذا الفصل دراسة مرجعية لآلية العمل الحالية.

1.2- مقدمة

مقدمة.

2.2 الحاجة إلى النظام

3.2 النظام الحالي ومشاكله / إن وجد

3.2 القيمة المضافة التي سيقدمها

الفصل الثالث

دراسة آلية إدارة المشاريع

نبين في هذا الفصل دراسة نظرية حول آلية تعريف المشاريع ومتابعتها وإدارتها في المعهد العالي.

1.2- مقدمة

إدارة المشاريع هي عملية تنظيمية تهدف إلى تحقيق أهداف محددة ضمن قيود معينة الوقت، الكلفة، الجودة. تتضمن إدارة المشاريع تخطيطًا دقيقًا، وتنظيمًا للمهام، وإدارة للموارد، بالإضافة إلى متابعة دقيقة لضمان تحقيق الأهداف بكفاءة وفعالية.

تعد إجراءات إدارة المشاريع جزءًا حيويًا في دورة حياة المشروع، حيث تشمل سلسلة من الخطوات التي تُتبع لضمان نجاح المشروع من البداية حتى النهاية. هذه الإجراءات تشمل تحديد نطاق المشروع، وضع خطط تفصيلية، تنفيذ تلك الخطط، مراقبة التقدم، وإجراء التعديلات عند الحاجة.

أما دورة حياة المشروع، فهي الإطار الذي يمر من خلاله المشروع من بدايته وحتى اكتماله. تبدأ هذه الدورة بمرحلة الطرح، حيث يتم تعريف المشروع وتحديد أهدافه الأساسية. تليها مرحلة التخطيط التي يتم فيها وضع التفاصيل اللازمة لتنفيذ المشروع بنجاح، بما في ذلك تحديد الموارد المطلوبة ووضع جدول زمني.

تأتي بعد ذلك مرحلة التنفيذ، حيث يتم تنفيذ الخطط الموضوعة والعمل على تحقيق الأهداف المحددة. وأخيرًا، تأتي مرحلة الإغلاق، حيث يتم تقييم النتائج، وتوثيق الإنجازات، والتأكد من تحقيق أهداف المشروع بشكل كامل.

دورة حياة المشروع ليست مجرد خطوات متتابعة، بل هي عملية ديناميكية تتطلب متابعة مستمرة وتعديلات حسب الحاجة لضمان النجاح في تحقيق الأهداف المرجوة.

1.2- دورة حياة المشروع (النشاط)

دورة حياة المشروع هي عملية منظمة تتكون من مراحل متتابعة تهدف إلى ضمان تنفيذ المشروع بشكل فعال من البداية حتى النهاية. تبدأ هذه الدورة بفكرة أو مبادرة تحتاج إلى تحليل وتخطيط دقيق قبل الشروع في التنفيذ. يمر المشروع عبر عدة مراحل رئيسية، تشمل الطرح، التخطيط، التنفيذ، والمراقبة والتحكم، وأخيرًا الإغلاق.

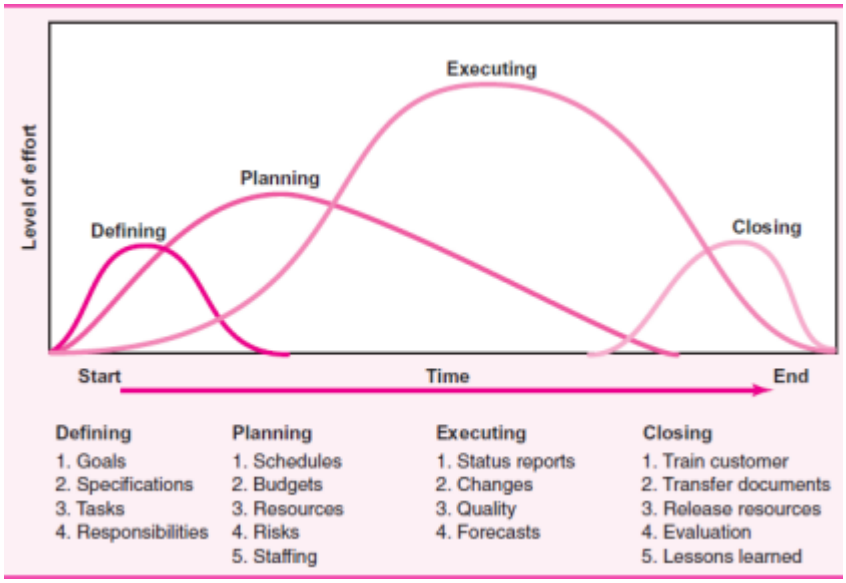
كل مرحلة من هذه المراحل تلعب دورًا حاسمًا في تحديد نجاح المشروع، حيث يتم في كل منها اتخاذ قرارات واستراتيجيات تساهم في تحقيق أهداف المشروع بأعلى قدر ممكن من الكفاءة.

ففي مرحلة الطرح يتم فيها تحديد فكرة المشروع والهدف منه . وتقييم جدواه من الناحية الاقتصادية والعملية. يتم تحليل الفكرة وتحويلها إلى مشروع قابل للتنفيذ.

وبالنسبة لمرحلة التخطيط فتُعتبر هذه المرحلة من أهم مراحل المشروع، حيث يتم وضع خطة تفصيلية تحدد كيفية تنفيذ المشروع لتحقيق أهدافه. ويتضمن التخطيط وضع جداول زمنية مفصلة، تحديد الموارد البشرية والمادية اللازمة، بالإضافة إلى وضع استراتيجية لإدارة المخاطر والجودة. وينتج عن هذه المرحلة خطة إنفاق , خطة لإدارة المخاطر ,خطة تنفيذ المشروع.

وفي مرحلة التنفيذ ، يتم تنفيذ الأنشطة والخطط الموضوعة في مرحلة التخطيط. تُدار الفرق وتُنسق العمليات والأنشطة لضمان تنفيذ المشروع وفقًا للجدول الزمني والمعايير المحددة. من المهم هنا مراقبة العمل بشكل مستمر وتقييم وتحديد أداء العاملين.لذلك تأتي على التوازي مرحلة التحكم والمراقبة فهذه المرحلة ترافق مرحلة التنفيذ وتتداخل معها. والهدف من هذه المرحلة هو التأكد من أن المشروع يسير وفقًا للخطة الموضوعة. يتم في هذه المرحلة مراقبة تقدم المشروع، قياس الأداء الفعلي مقابل الأداء المتوقع، وإجراء التعديلات اللازمة عند الضرورة. ويتنج عن هذه المرحلة تقارير تساهم في قياس الأداء والتقدم التي بدورها تستخدم لاطلاع الجهات الإدارية عليها لاتخاذ القرارات المناسبة.

وفي النهاية تأتي مرحلة الإغلاق أو كما يسميها البعض الانجاز. وتعتبر هذه المرحلة نهاية دورة حياة المشروع، حيث يتم فيها إغلاق جميع الأنشطة والتأكد من تحقيق جميع أهداف المشروع. يتم في هذه المرحلة تسليم المخرجات النهائية لأصحاب المصلحة، توثيق الدروس المستفادة، وتقييم أداء المشروع ككل.ويبين الشكل أدناه (x) دورة حياة المشروع بشكل مرئي أوضح.



الشكل 1: عمليات المشروع عبر دورة حياته

2.3 آلية إدارة المشروع المتبعة في المعهد

من خلال الاجتماعات مع المعنين والاطلاع على آلية سير العمل تبين لنا أن عملية إدارة المشاريع حاليا تتم وفق الإجرائيات الآتية

- إجرائية الطرح والتعريف :

يقوم الزبون (الجهة الطارحة) بتقديم المشروع الذي ير غب بأن ينفذ له للجهة المعنية.

بعد قبول فكرة المشروع يجب تعريف المشروع وتوصيفه ودراسة الجدوى منه وتصنيفه فتسند مهمة إدارة المشروع (مدير المشروع) للموظف المناسب كما تسند أيضا مهمة رئاسة فريق العمل للموظف المناسب (رئيس فريق العمل) .

- إجرائية التخطيط :

بعد قيام كل من مدير المشروع ورئيس فريق العمل بدراسة المشروع المراد طرحه وقيامهم بدراسة المراحل التي يتطلبها المشروع, يقوم مدير المشروع بإضافة تصنيف المشروع و الجدوى الاقتصادية منه والخطة المالية وعدد العمال اللازم.

يرفع المشروع بعد هذا التوصيف وارفاقه بالمفات التي يراها كل من مدير المشروع ورئيس فريق لعمل ضرورية إلى مجلس المعهد (الإدارة المعنية بالموافقة) للموافقة على المشروع أو رفضه.

في حال لم تر الإدارة أن المشروع يلبي متطلعاتها وأنه غير مجد وقررت رفضه يعلم المعنيين أن المشروع قد ألغي.

وفي حال الموافقة على المشروع واعطائه المعلومات الذاتية المناسبة (رقم كتاب الموافقة ,رمز ...):

يقوم رئيس فريق العمل بإضافة العاملين المشاركين إلى هذا المشروع ويسند لكل منهم نسبة انشغالية ومهمته المتوقعة ضمن المشروع. كما يقوم رئيس فريق العمل بإضافة المراحل ضمن المشروع وتاريخ بدئها ومدة كل منها وتوزعها على الربعيات.

ومن بعد ذلك ترفع هذه المعلومات إلى الجهة المعنية ويبدأ من بعدها العمل بالمشروع

- إجرائية المتابعة :

تنقسم السنة إلى أربع ربعيات، وفي كل ربعية يجب على رئيس فريق العمل القيام بعملية المتابعة حيث تنقسم هذه العملية إلى جزءين هما

- متابعة المراحل حيث يسجل رئيس فريق العمل نسبة الإنجاز في هذه المرحلة ويضيف ملاحظاته.
- متابعة عمل الموظفين حيث يسجل رئيس فريق العمل نسبة انشغال الموظفين وأدوارهم في كل مرحلة.

بعد القيام بعملية المتابعة يستطيع رئيس فريق العمل إصدار تقارير المتابعة.

- عملية الاطلاع :

في نهاية كل ربعية يجب أن ترفع تقارير المتابعة إلى مدراء المشاريع و إلى مجلس المعهد (الإدارة المعنية) للاطلاع عليها. والاطلاع على المشاريع المتأخر لتذليل الصعوبات أو إلغائها.

- إجرائية التسليم :

في هذه المرحلة وبعد انتهاء التسليمات الناتجة عن انهاء المراحل يتم تسليمها إلى الزبون وبعد انتهاء التسليم يتم انهاء المشروع وحل فريق العمل.

الفصل الرابع

خطة إدارة المشروع

نبيّن في هذا الفصل خطة إدارة المشروع والخطة الزمنية لتطوير النظام كما نبين أيضا الاجرائية المتبعة في تطوير النظام.

1.4- مقدمة

تعتبر خطة إدارة المشروع بمثابة الوثيقة الأساسية التي توضح استراتيجية تنفيذ المشروع، الجدول الزمني، والمنهجيات التي سيتم اعتمادها لتحقيق أهداف المشروع بكفاءة وفعالية. وتغطي هذه الخطة كافة مراحل المشروع بدءاً من جمع المتطلبات وحتى تسليم المنتج النهائي، مع التركيز على جودة التنفيذ والتزامه بالمعايير الزمنية والميزانية المحددة.

2.4- الإجرائية المتبعة في تطوير المشروع

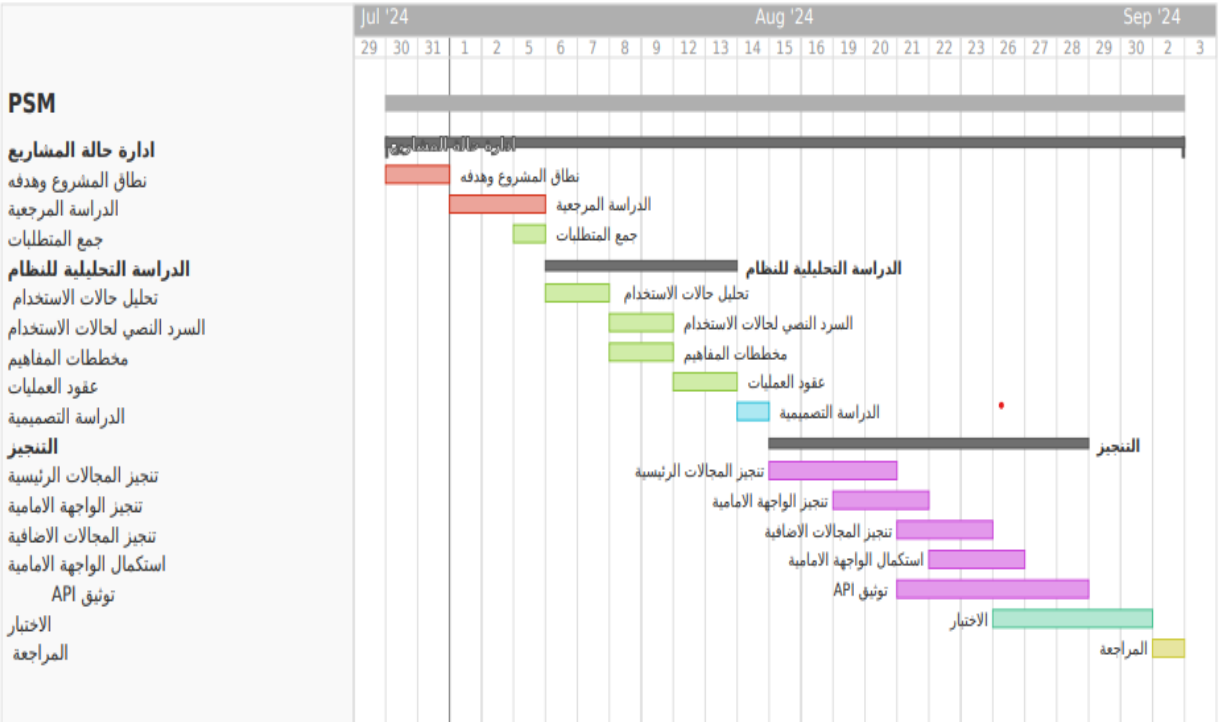
اتبعنا في تطوير النظام اجرائية التطوير الشلالي (Waterfall Process) والتي تتسم بكونها منهجية خطية متسلسلة، حيث يتم الانتقال من مرحلة إلى أخرى بعد إتمام المرحلة السابقة.ولتحقيق هذه المنهجية قسمنا مراحل العمل على النحو الآتي:

1. **مرحلة التواصل مع الزبون وجمع المتطلبات**: يتم في هذه المرحلة التفاعل مع الأطراف المعنية لجمع كافة المتطلبات التي يريدونها والتعرف على آلية العمل المتبعة.
2. **مرحلة تحديد المتطلبات وتوصيفها**: يتم تحويل المتطلبات المجمعة إلى مواصفات واضحة ومحددة تساعد في تصميم النظام وعرضها على الجهة الطارحة للتأكد من أن هذه هي المتطلبات التي يريدونها.
3. **مرحلة الدراسة المرجعية**: في هذه المرحلة يتم تحليل الالية المتبعة في النظام القديم إن وجد ودراسة آلية العمل لفهمها.
4. **مرحلة الدراسة التحليلية**:
5. **مرحلة التطوير**:
6. **مرحلة الاختبار** :
7. **مرحلة التسليم** :

3.4- الخطة الزمنية

مشروع إدارة حالة مشاريع المعهد العالي		
المرحلة	تاريخ البدء	المدة
تحديد نطاق المشروع وهدفه	31/7	1
جمع المتطلبات	1/8	2
الدراسة المرجعية	2/8	5
الدراسة التحليلية (تحليل المتطلبات)	7/8	7
تصميم النظام	13/8	2
تطوير النظام	16/8	14
الاختبار	29/8	2
المراجعة النهائية	1/9	2/9

4.4- مخطط غانت Gantt Chart



الشكل 1 : مخطط غانت لأنشطة خطة التنفيذ

الفصل الخامس

الدراسة التحليلية

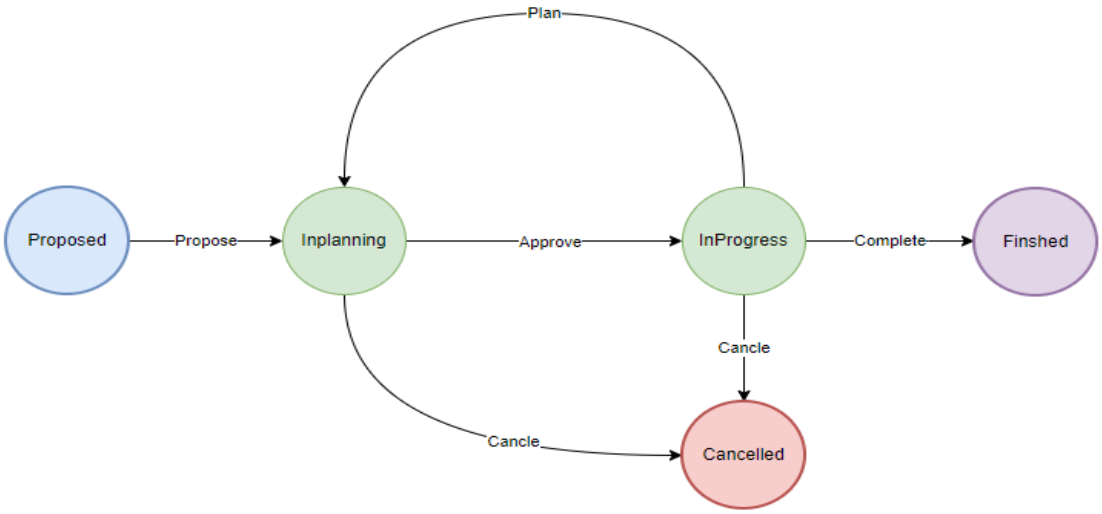
يقدم هذا الفصل تحليلا للمتطلبات التي أوردناها في الفصل الأول.

1.6- مقدمة

نقدم في هذا الفصل دراسة تحليلية للمسألة المطروحة ونبين حالات الاستخدام ووصفها السردي وبعض مخططات UML.

2.6 مخطط أطوار المشروع

يوضح الشكل المرفق أدناه مخططا لطوار حالة المشروع وتغيراتها بناءا على المؤثرات المطبقة على المشروع.



الشكل 2 : مخطط أطوار المشروع.

3.6- مدخلات ومخرجات النظام

تبين لنا من خلال الاجتماعات مع المعنين أنه يوجد حاليا عدد من الاستمارات التي يحتاجونها ويتعاملون معها أثناء إدارة ,طرح وتتبع المشاريع وهي على النحو الآتي :

1. استمارة طرح مشروع
2. استمارة تصنيف مشروع
3. استمارة الجدوى الاقتصادية من المشروع
4. استمارة انشغالية العاملين الربعية (عملية المتابعة)
5. استمارة متابعة المراحل (عملية المتابعة)
6. استمارة الاسناد التخطيطي (تهتم بالنسبة المخططة لانشغالية العامل في كل ربعية)
7. البطاقة الزمنية (توضح توزع مراحل المشروع على أشهر العام)
8. الخطة السنوية المتوقعة
9. استمارة انجاز مشروع
10. استمارة التكليف الربعي (العمل الفعلي للعاملين وهي ناتجة عن المتابعة)

تبين لنا أن الاستمارات من 1 وحتى 6 هي مدخلات للنظام يجب أن تدخل إلى النظام

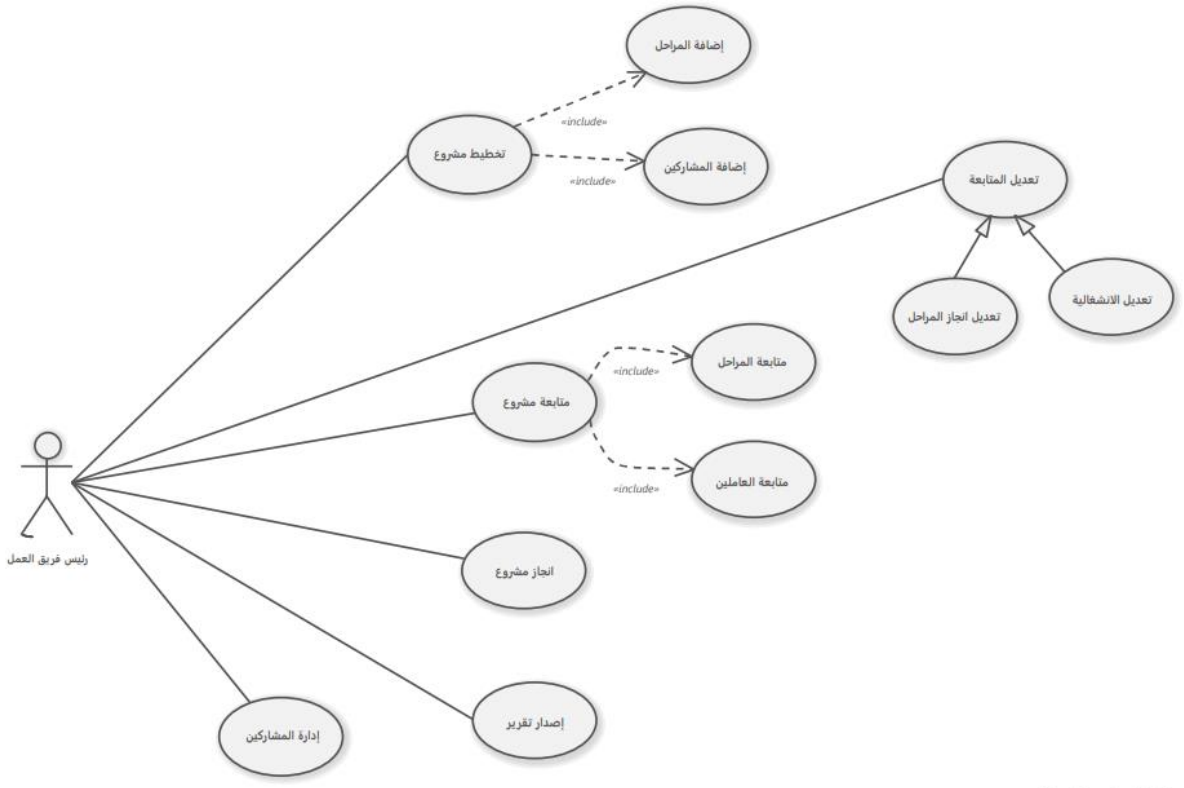
وأن الاستمارات من 7 وحتى 10 هي مخرجات للنظام يجب أن تصدر عن النظام وترسل للمعنين

4.6 مخططات حالات الاستخدام

نبين هنا حلات الاستخدام في النظام لتحقيق المتطلبات المبينة في الفصل الأول والسرد النصي الذي يصف آلية تحقيقها ,حيث نورد هذا السرد النصي بناءا على آلية سير العمل التي ذكرناها آنفا.

1.4.6 حالات الاستخدام الخاصة برئيس فريق العمل

حالات الاستخدام



الشكل 3 : مخطط حالات الاستخدام لرئيس فريق العمل.

الوصف السردي لحالات الاستخدام

أ- حالة الاستخدام متابعة مشروع

اسم حالة الاستخدام : متابعة مشروع	
الفاعلون الأوليون	رئيس فريق العمل
الشروط المسبقة	• المشروع معروف ومسجل لدى النظام. • المشروع في طور التنفيذ. • رئيس فريق العمل مسجل في النظام.
الشروط اللاحقة	• عملية المتابعة اكملت وسجلت لدى النظام. • تقرير المتابعة أرسل للوكيل العلمي. • انشغالية العاملين أسندت و وتقرير الانشغالية تم رفعه.
الوصف	يقوم رئيس فريق العمل بمتابعة المشروع حيث يسند نسب الانشغالية للعاملين.ويسند نسب الانجاز في المراحل.

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	رئيس فريق العمل
	1. يقوم رئيس فريق العمل ببدء عملية متابعة جديدة
2. يرسل النظام قائمة بالمراحل التي هي قيد الانجاز.	3. يقوم رئيس فريق العمل من أجل كل مرحلة بإدخال معلومات المتابعة (نسبة الانجاز , الملاحظات , وضع المرحلة).
	4. ييطلب تأكيد معلومات متابعة المراحل.
5. يسجل النظام معلومات متابعة المراحل.	.

6. يرسل النظام قائمة بالمشاركين في المشروع.	7. يختار رئيس فريق العمل المشاركين في هذه المتابعة ويدخل لكل منهم نسبة الانشغالية الخاصة به.
	8. يطلب تأكيد معلومات متابعة المشاركين.
9. يسجل معلومات المشاركين	
	10. يطلب تأكيد عملية المتابعة
11. يسجل عملية المتابعة , ويرسل تقرير المتابعة إلى الوكيل العلمي	
12. يرسل تقرير الانشغالية للمشاركين.	
13. يعلم رئيس فريق العمل بنجاح العملية.	

المسارات البديلة

A1: في المرحلة رقم 5 إذا كانت معلومات المرحلة غير صحيحة, يطلب النظام من رئيس فريق العمل إعادة إدخال المعلومة ويتابع من المرحلة رقم 4.

A2: في المرحلة رقم 9 إذا كانت نسبة انشغالية العامل لا تتوافق مع نسبة تفرغه للعمل يطلب النظام من رئيس فريق العمل إعادة الإددخال ويتابع من المرحلة رقم 8

A2: في المرحلة رقم 10 إذا أراد رئيس فريق العمل إرفاق مستندات, يقوم رئيس فريق العمل بإرفاق المستندات المطلوبة ويتابع من المرحلة رقم 10.

المسارات الخاطئة

لايوجد.

ب– حالة الاستخدام تخطيط مشروع

اسم حالة الاستخدام : تخطيط مشروع	
الفاعلون الأوليون	رئيس فريق العمل
الشروط المسبقة	• المشروع معروف ومسجل لدى النظام. • المشروع في طور التخطيط. • رئيس فريق العمل مسجل في النظام.
الشروط اللاحقة	• مراحل المشروع سجلت لدى النظام. • المشاركون في المشروع سجلوا لدى النظام. • تفرغ العاملين للمشروع سجل لدى النظام. • المشروع يصبح في حالة التنفيذ.
الوصف	يقوم رئيس فريق العمل بإضافة مراحل المشروع والمشاركين بالمشروع.

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	رئيس فريق العمل
	1. يقوم رئيس فريق العمل ببدء عملية تخطيط للمشروع
2. يرسل النظام معلومات المشروع وتوصيفه وعدد العمال المطلوب له.	3. يبدأ رئيس فريق العمل عملية إضافة المراحل
	4. يضيف رئيس فريق العمل من أجل كل مرحلة جديدة بإدخال معلوماتها حيث يحدد تاريخ بدئها ووزنها ومدة تنفيذها

	5. يطلب رئيس فريق العمل تأكيد عملية إضافة المراحل.
6. يسجل النظام مراحل المشروع.	
	7. يبدأ رئيس فريق العمل عملية إضافة المشاركين.
8. يرسل النظام قائمة بالموظفين المتاحين.	
	9. يختار رئيس فريق العمل المشاركين ويسند لكل منهم نسبة تفرغ.
	10. يطلب تأكيد عملية إضافة المشاركين.
11. يسجل المشاركين في المشروع	
	12. يطلب إنهاء عملية التخطيط.
13. يعلم كل مشارك باختياره ضمن المشروع.	
14. يعلم رئيس فريق العمل بنجاح العملية.	

المسارات البديلة

A1: في المرحلة رقم 5 إذا كانت معلومات أوزان المراحل غير متوافقة, يطلب النظام من رئيس فريق العمل إعادة إدخال المعلومة ويتابع من المرحلة رقم 4.

A2: في المرحلة رقم 9 إذا كانت نسبة التفرغ العامل لا تتوافق مع نسبة تفرغه العظمى للعمل يطلب النظام من رئيس فريق العمل إعادة الإدخال ويتابع من المرحلة رقم 8

A2: في المرحلة رقم 10 إذا أراد رئيس فريق العمل إرفاق مستندات, يقوم رئيس فريق العمل بإرفاق المستندات المطلوبة ويتابع من المرحلة رقم 10.

المسارات الخاطئة

لايوجد.

ت– حالة الاستخدام إصدار تقرير

اسم حالة الاستخدام : إصدار تقرير	
الفاعلون الأوليون	رئيس فريق العمل , مدير المشروع
الشروط المسبقة	• المشروع معروف ومسجل لدى النظام. • طالب الخدمة (رئيس فريق العمل, مدير المشروع) مسجل في النظام.
الشروط اللاحقة	لايوجد.
الوصف	يقوم رئيس فريق العمل بإصدار تقرير من التقارير المتوفرة عن المشروع.

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	رئيس فريق العمل
	1. يقوم ببدء عملية صدار تقرير للمشروع
2. يرسل النظام قائمة بأنواع التقارير المتوفرة.	
	3. يختار التقرير المطلوب
	4. يطلب تأكيد عملية الإصدار
5. يسجل النظام معلومات عملية الإصدار	
6. يرسل التقرير المطلوب	
	7. يطلب إنهاء عملية الإصدار.
8. يعلم رئيس فريق العمل بنجاح العملية.	

المسارات البديلة

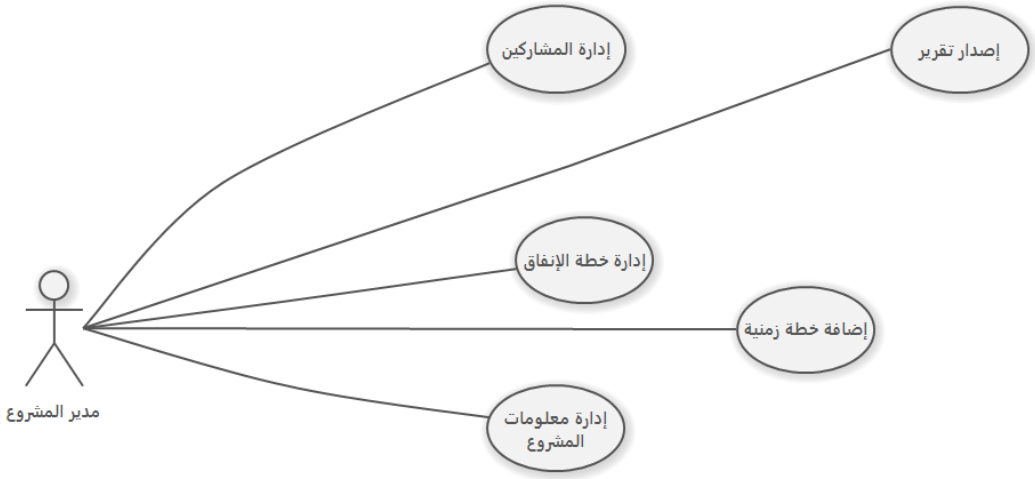
A1: في المرحلة رقم 7 إذا أراد إصدار تقرير آخر, فإنه يتابع من المرحلة رقم 3.

المسارات الخاطئة

لايوجد.

2.4.6 حالات الاستخدام الخاصة بمدير المشروع

حالات الاستخدام



الشكل 4 : مخطط حالات الاستخدام لمدير المشروع.

الوصف السردي لحالات الاستخدام

أ- حالة الاستخدام إضافة خطة الإنفاق

اسم حالة الاستخدام : إضافة خطة إنفاق	
الفاعلون الأوليون	مدير المشروع
الشروط المسبقة	• المشروع معروف ومسجل لدى النظام. • المشروع في طور التخطيط. • مدير المشروع مسجل في النظام.
الشروط اللاحقة	أ. خطة الانفاق أضيفت للمشروع.
الوصف	يقوم مدير المشروع بإضافة قائمة بالإنفاقات المتوقعة ضمن المشروع.

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	مدير المشروع
	9. يقوم ببدء عملية إضافة خطة إنفاق
10. يرسل معلومات المشروع (ميزانية المشروع وتوصيفه).	
	11. يدخل مدير المشروع قائمة لمعلومات الإنفاق (نوع الكلفة , البيان,الشراء المحلي والخارجي, التاريخ المتوقع لإنفاق)
	12. يطلب تأكيد عملية الإضافة
13. يسجل النظام معلومات عملية الإنفاق	
14. يعلم رئيس فريق العمل بنجاح العملية.	

المسارات البديلة

A1: في المرحلة رقم 13 إذا كانت كلفة خطة الانفاق أعلى من ميزانية المشروع يطلب منه إعادة الإدخال أي يتابع من المرحلة رقم 11.

المسارات الخاطئة

لايوجد.

ب– حالة الاستخدام إدارة معلومات المشروع

اسم حالة الاستخدام : إضافة معلومات المشروع	
الفاعلون الأوليون	مدير المشروع
الشروط المسبقة	أ. المشروع معروف ومسجل لدى النظام. ب. المشروع في طور التخطيط. ت. مدير المشروع مسجل في النظام.
الشروط اللاحقة	ملعومات المشروع سجلت لدى الظام.
الوصف	يقوم مدير المشروع بإضافة معلومات المشروع .

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	مدير المشروع
	1. يقوم ببدء عملية إضافة معلومات للمشروع
2. يرسل النظام الاستمارات الواجب ملؤها (استمارة التصنيف , استمارة النوع, استمارة الجدوى)	

	3. يقوم مدير المشروع بملئ كل من الاستمارات
	4. يطلب تأكيد عملية الإضافة
5. يسجل النظام معلومات المشروع المضافة	
6. يعلمه بنجاح عملية الاضافة	
	7. يطلب إنهاء عملية الإضافة.
8. يعلمه بنجاح العملية.	

المسارات البديلة

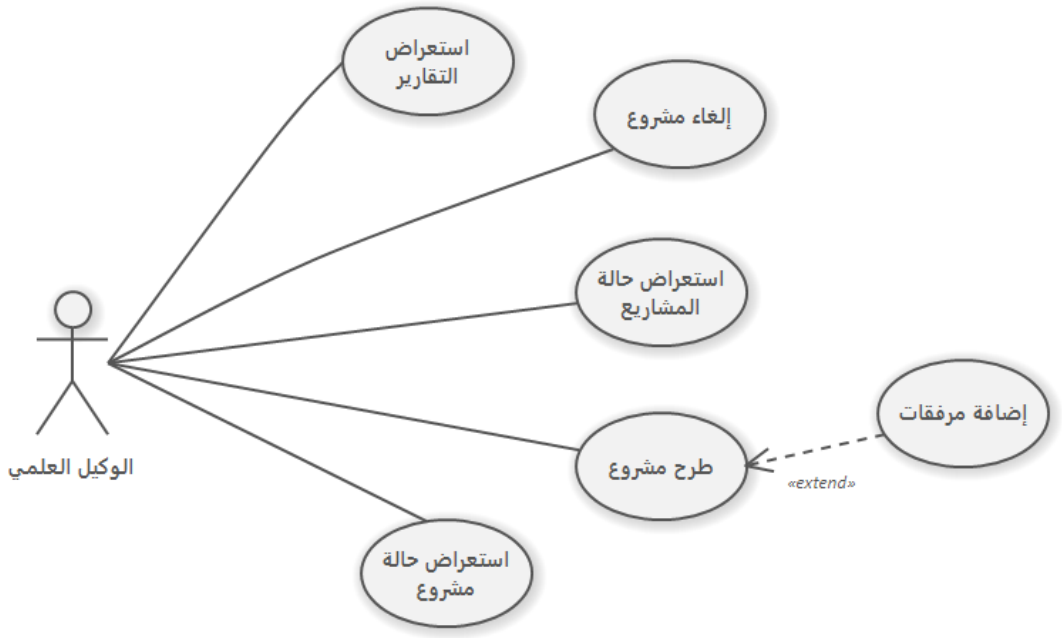
لايوجد.

المسارات الخاطئة

لايوجد.

3.4.6 حالات الاستخدام الخاصة بالوكيل العلمي

حالات الاستخدام



الشكل 5 : مخطط حالات الاستخدام للوكيل العلمي.

الوصف السردي لحالات الاستخدام

أ- حالة الاستخدام طرح مشروع

اسم حالة الاستخدام : طرح مشروع	
الفاعلون الأوليون	الوكيل العلمي
الشروط المسبقة	أ. طالب الخدمة مسجل في النظام. ب. المشروع في طور الطرح.
الشروط اللاحقة	المشروع سجل لدى النظام. رئيس فريق العمل ومدير المشروع أعلم باختياره ضمن المشروع.
الوصف	يقوم الوكيل العلمي بإضافة مشروع جديد ويسند مهمة الإدارة لمدير المشروع ويختار رئيس فريق العمل.

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	**رئيس فريق العمل**
	9. يقوم ببدء عملية طرح مشروع
10. يرسل النظام قائمة بأنواع المشاريع المتوفرة	
	11. يختار نوع المشروع.
12. يرسل قائمة بالموظفين المتاحين	
	13. يختار رئيس فريق العمل ومدير المشروع
	14. يحدد بيانات المشروع المطروح (الجهة الطارحة,معلومات التمويل ,...)
	15. يطلب إتمام عملية الطرح

16. يعلم رئيس فريق لعمل ومدير المشروع بإختيارهم	
17. يسجل معلومات المشروع.	
18. يعلم الوكيل بنجاح العملية.	

المسارات البديلة

A1: في المرحلة رقم 6 إذا أراد الوكيل إضافة مرفقات, فيختار المرفقات ويتابع من المرحلة رقم 7.

المسارات الخاطئة

لايوجد.

ب- حالة الاستخدام استعراض التقارير

اسم حالة الاستخدام : استعراض التقارير	
الفاعلون الأوليون	الوكيل العلمي
الشروط المسبقة	أ. طالب الخدمة مسجل في النظام.
الشروط اللاحقة	لايوجد.
الوصف	يقوم الوكيل العلمي بالاطلاع على تقارير المشاريع.

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	**رئيس فريق العمل**
	1. يقوم ببدء عملية الاطلاع على التقارير .
2. يرسل النظام قائمة بتقارير المشاريع المتوفرة.	
	3. يختار التقرير المطلوب
	4. يطلب تأكيد عملية الاطلاع

5. يسجل النظام معلومات عملية الإصدار	
6. يرسل التقرير المطلوب	
	7. يطلب إنهاء عملية الاطلاع.
8. يعلم الوكيل بنجاح العملية.	

المسارات البديلة

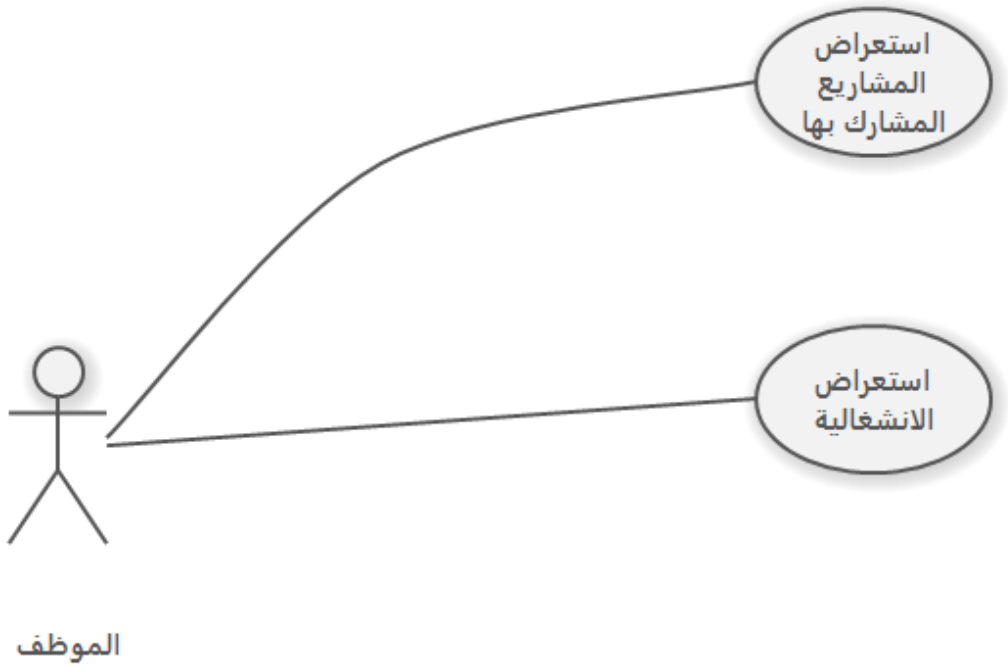
A1: في المرحلة رقم 7 إذا أراد الاطلاع على تقرير آخر, فإنه يتابع من المرحلة رقم 3.

المسارات الخاطئة

لايوجد.

4.4.6 حالات الخاصة بالموظف

حالات الاستخدام



الشكل 6 : مخطط حالات الاستخدام للموظف.

الوصف السردي لحالات الاستخدام

أ- حالة الاستخدام استعراض الانشغالية

اسم حالة الاستخدام : استعراض الانشغالية	
الفاعلون الأوليون	الموظف
الشروط المسبقة	أ. الموظف مسجل في النظام ومعروف لدى النظام.
الشروط اللاحقة	لايوجد.
الوصف	يقوم الموظف باختيار مشروع من المشاريع المشارك بها ويتطلع على بيانات انشغاليته ضمن المشروع.

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	الموظف
	1. يقوم ببدء عملية الاطلاع على الانشغالية.
2. يرسل النظام قائمة بالمشاريع االتي يشارك بها.	
	3. يختار المشروع المطلوب
	4. يطلب تأكيد عملية الاطلاع
5. يرسل ملعومات الانشغالية	
	6. يطلب إنهاء العملية.

المسارات البديلة

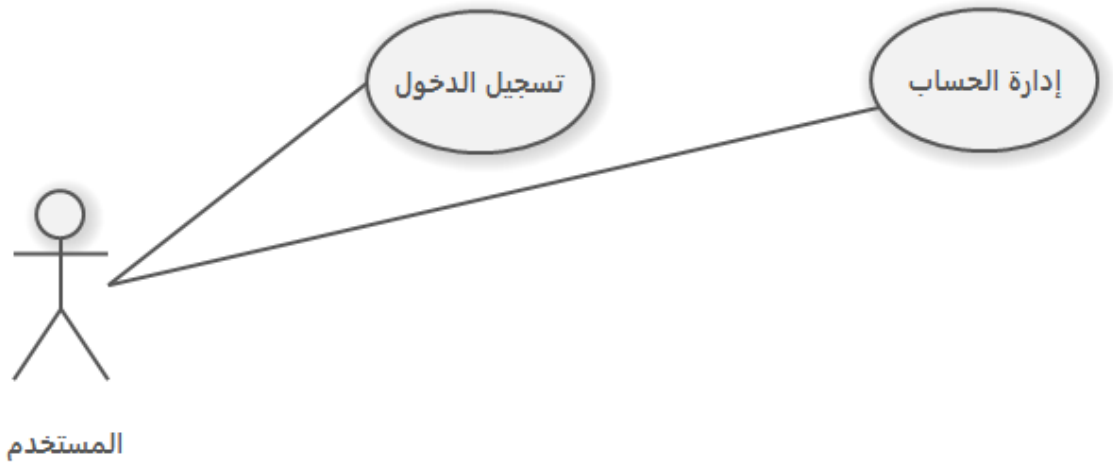
A1: في المرحلة رقم 4 إذا اختار مشروع ليس من المشاريع التي يشارك بها, يعلمه النظام بأن المشروع ليس من المشاريع المشارك بها ويتابع من المرحلة رقم 3.

المسارات الخاطئة

لايوجد.

5.4.6 حالات الاستخدام الخاصة بالمستخدم

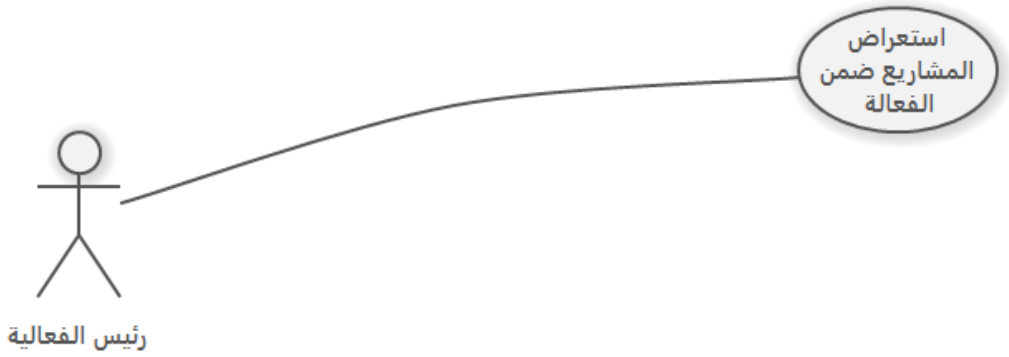
حالات الاستخدام



الشكل 7 : مخطط حالات الاستخدام للمستخدم.

6.4.6 حالات الاستخدام الخاصة برئيس الفعالية

حالات الاستخدام



الشكل 8 : مخطط حالات الستخدام لرئيس الفعالية.

الوصف السردي لحالات الاستخدام

أ- حالة الاستخدام استعراض المشاريع ضمن الفعالية

اسم حالة الاستخدام : استعراض المشاريع ضمن الفعالية	
الفاعلون الأوليون	رئيس الفعالية
الشروط المسبقة	أ. رئيس الفعالية مسجل في النظام.
الشروط اللاحقة	لايوجد.
الوصف	يقوم رئيس الفعالية بإصدار تقرير بالمشاريع ضمن الفعالية االتي يرأسها.

سير الأحاث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	رئيس فريق العمل
	1. يقوم بطلب عملية استعراض المشاريع
2. يرسل النظام قائمة بالمشاريع ضمن الفعالية	
	3. يختار المشروع الذي سيطلع على حالته
4. يسجل عملية الإصدار	
5. يرسل التقرير المطلوب مع المرفقات.	
	6. يطلب إنهاء العملية.
7. يعلم رئيس الفعالية بنجاح العملية.	

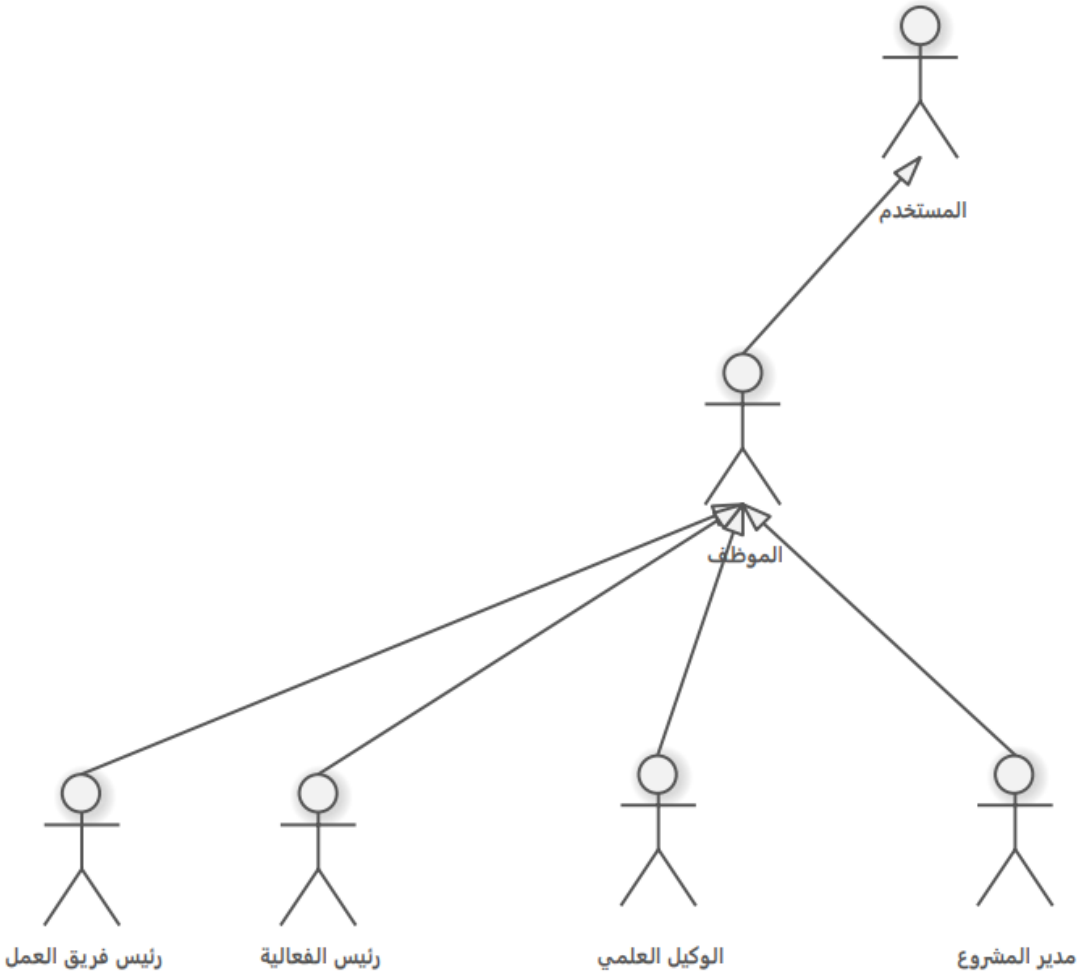
المسارات البديلة

لايوجد.

المسارات الخاطئة

لايوجد.

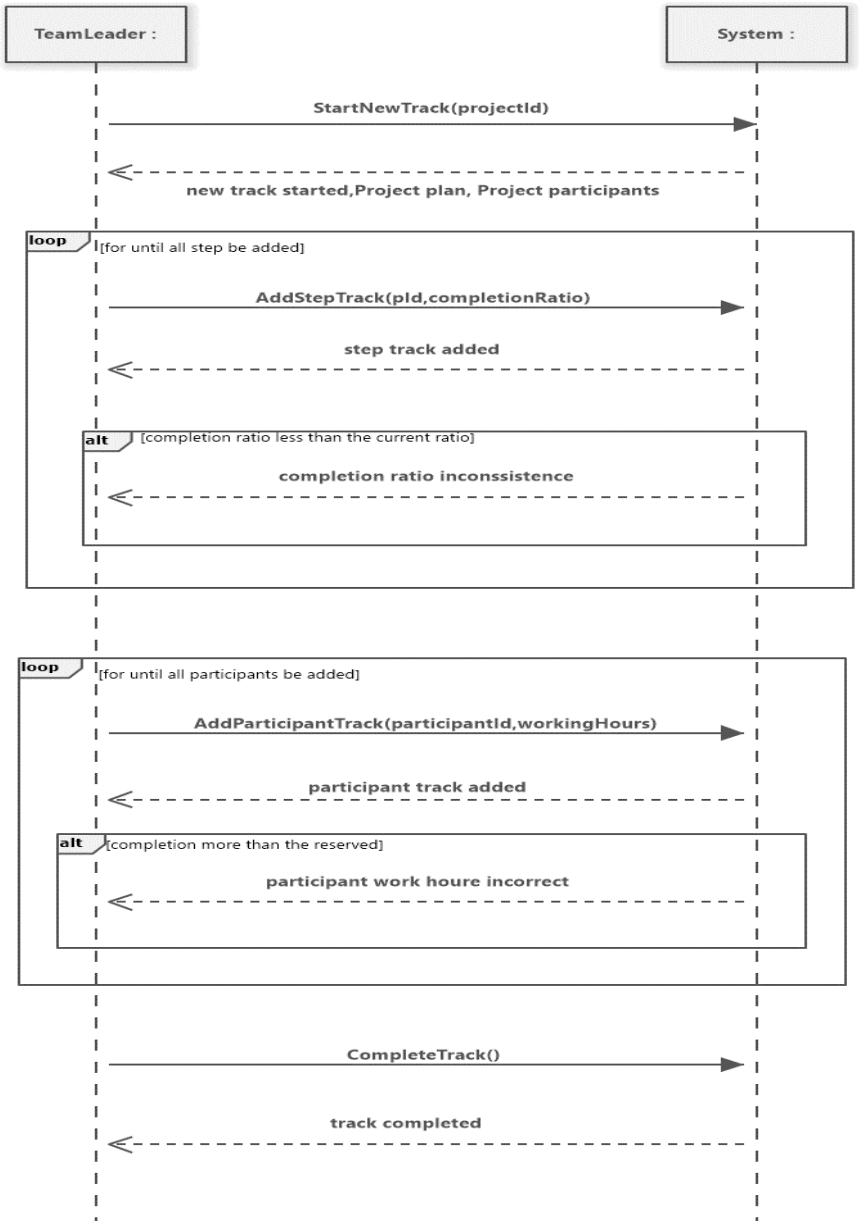
8.4.6 العلاقات بين الفاعلين



الشكل 9 : العلاقات بين الفاعلين

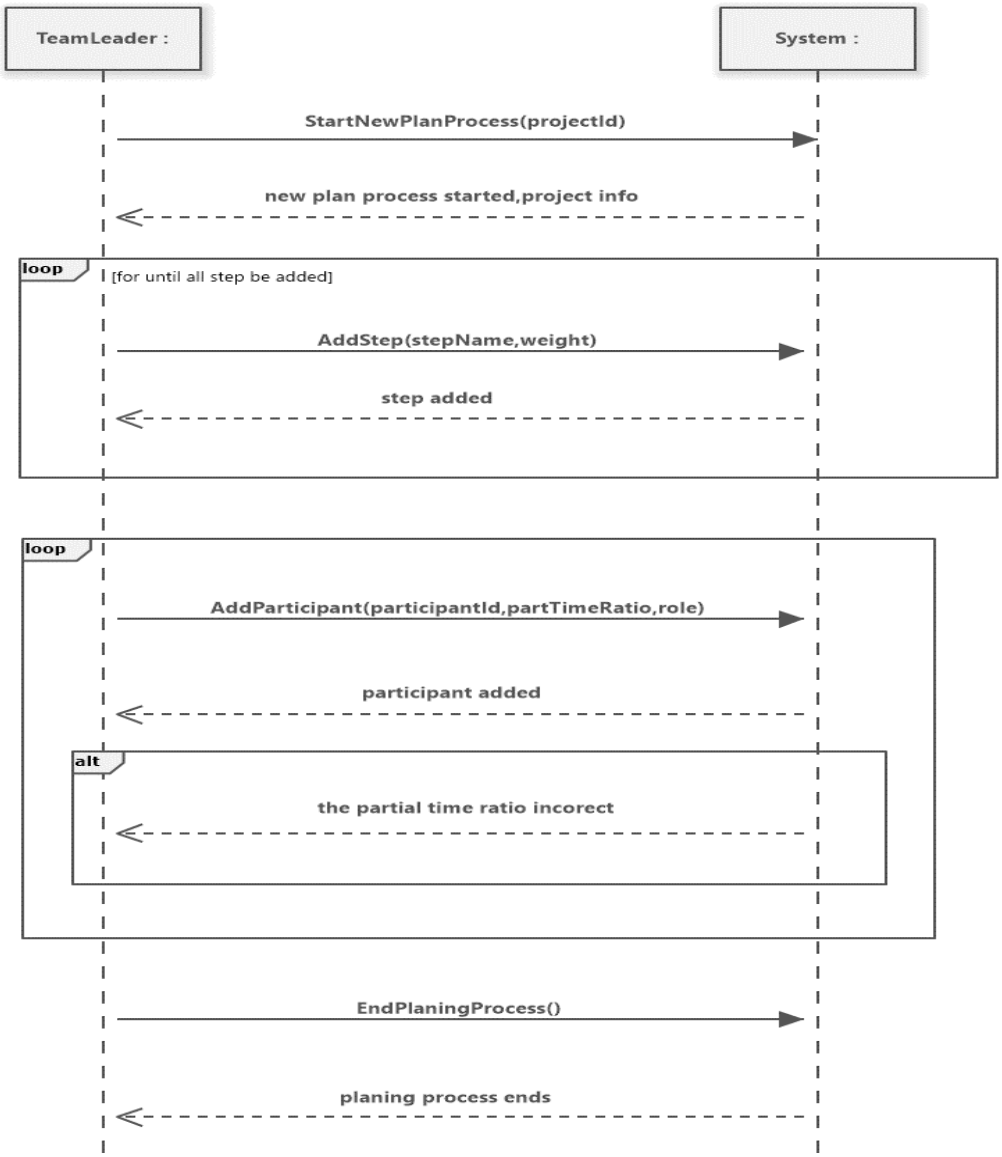
6.6 مخططات التتالي لبعض حالات الاستخدام

1.6.6 مخطط التتالي لحالة الاستخدام متابعة مشروع



الشكل 10 : مخطط التتالي لحالة الاستخدام متابعة مشروع

2.6.6 مخطط التتالي لحالة الاستخدام تخطيط مشروع



الشكل 11 : مخطط التتالي لحالة الستخدام تخطيط مشروع.

7.6 عقود عمليات حالة الاستخدام متابعة مشروع

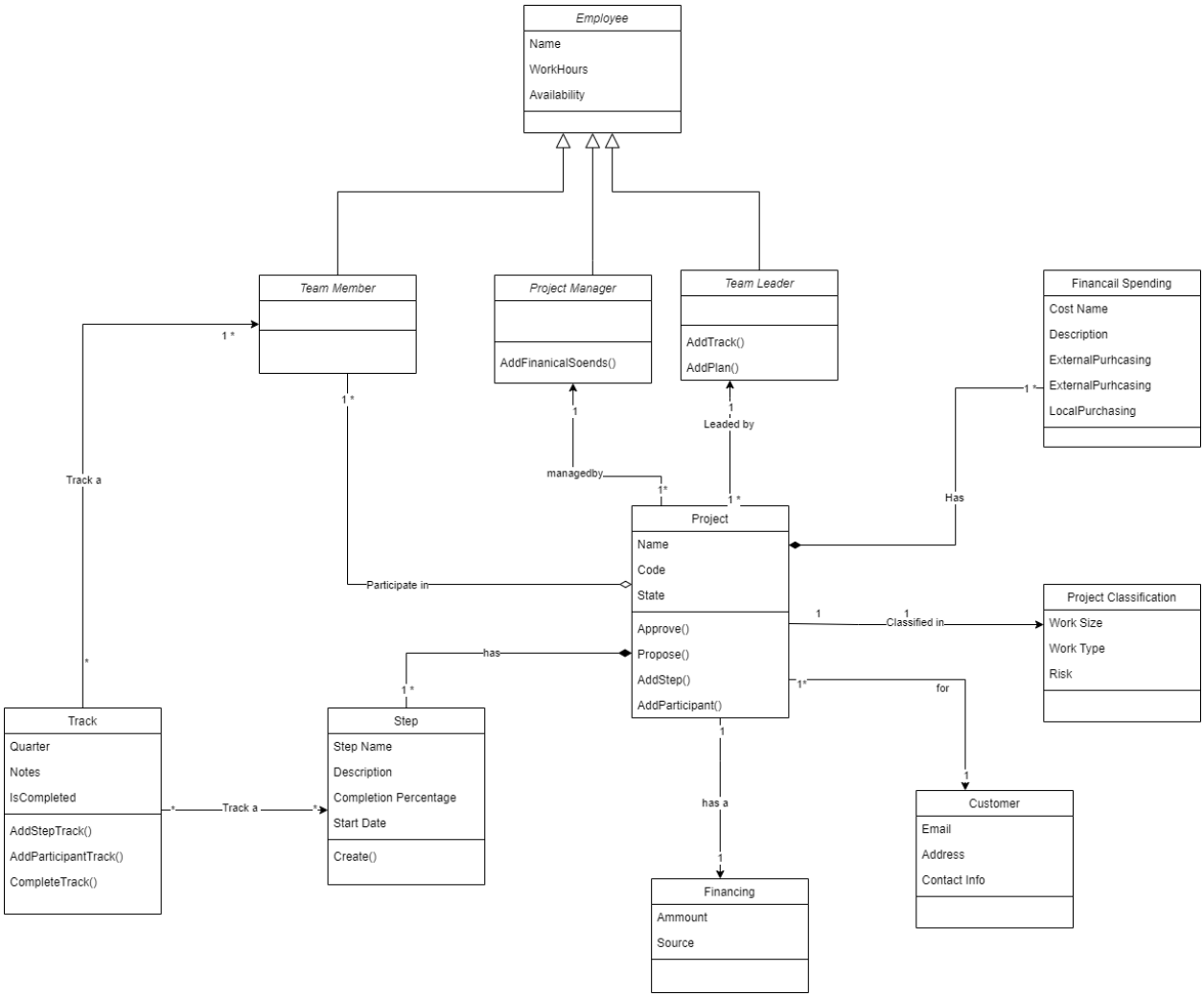
اسم العملية : StartNewTrack(projectId)	
Responsibilities	بدء عملية متابعة جديدة
Exception	لايوجد.
Precondition	الستخدم (رئيس فريق العمل) مسجل لدى النظام
Postcondition	عملية متابعة جديدة T أنشئت. نسخة من المشروع P أنشئت . عملية المتابعة T أضيفت إلى المشروع P.

اسم العملية : AddStepTrack(StepId , completionRatio)	
Responsibilities	إضافة متابعة مرحلة لعملية المتابعة.
Exception	نسبة الانجاز لاتتوافق مع النسبة الحالية.
Precondition	الستخدم (رئيس فريق العمل) مسجل لدى النظام عملية المتابعة Track التي ستضاف إليها معروفة لدى النظام. المرحلة Step التي ستتابع معروفة لدى النظام.
Postcondition	متابعة مرحلة StepTrack أنشئت . متابعة المرحلة StepTrack أضيفت لعملية المتابعة Tack . نسبة الانجاز للمرحلة Step أضيف أليه النسبة الجديدة completionrRatio .

اسم العملية : **AddParticipantTrack(participantId,workHours)**	
Responsibilities	إضافة عملية متابعة مشارك .
Exception	لايوجد.
Precondition	المستخدم (رئيس فريق العمل) مسجل لدى النظام المشارك Participant معرف لدى النظام. عمبية المتابعة Track معروفة لدى النظام.
Postcondition	متابعة مشارك ParticiapantTrack أنشئت. نسبة ساعات العمل أسندت إلى متابعة المشارك ParticipantTrack. متابعة المشارك ParticipantTrack أضيفت إلى عملية المتابعة Track .

اسم العملية : **CompleteTrack()**	
Responsibilities	إنهاء عملية المتابعة.
Exception	لايوجد.
Precondition	الستخددم (رئيس فريق العمل) مسجل لدى النظام عملية المتابعة Trackمعروفة لدى النظام.
Postcondition	لايوجد.

8.6 مخطط الصفوف



الشكل 12 : مخطط الصفوف

الفصل السابع

الدراسة التصميمية

نبين في هذا الفصل شرحا عن الأساليب والمنهجيات و الأنماط التصميمية المستخدمة في تنجيز النظام.

1.7- مقدمة

الأنماط التصميمية (Design Patterns) هي حلول قياسية ومكررة لمشاكل شائعة تظهر في تصميم البرمجيات. ظهرت هذه الأنماط كجزء من تطوير البرمجيات بهدف تحسين جودة ومرونة التطبيقات وتقليل التعقيد. يُستخدم مفهوم الأنماط التصميمية لتسهيل عملية التصميم والبرمجة من خلال تقديم حلول مجربة يمكن إعادة استخدامها.

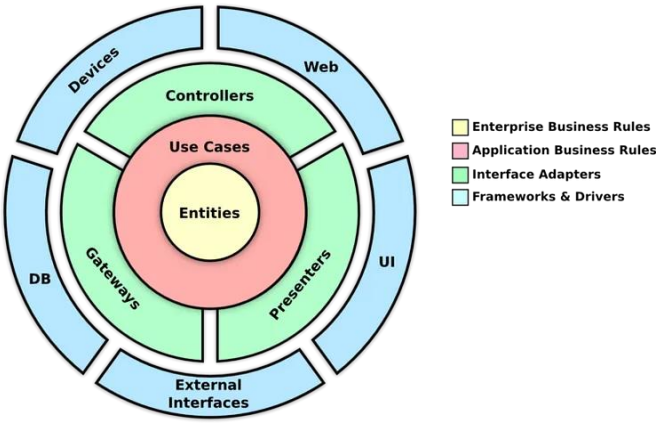
2.7 البنية المعمارية النظيفة – Clean Architecture

1.2.7 مفهوم البنية المعارية النظيفة

البنية المعمارية النظيفة هي فلسفة تصميم برمجيات تهدف إلى إنشاء أنظمة قابلة للصيانة والتوسعة، وتكون مستقلة عن تفاصيل التنفيذ. تعتمد هذه البنية على مبدأ فصل مكونات النظام إلى طبقات مستقلة،مركزة على قواعد العمل, مما يسهم في تقليل التبعيات بين هذه المكونات. يتيح هذا التصميم لكل مكون أن يتطور أو يتغير دون التأثير على باقي النظام، مما يؤدي إلى نظام أكثر مرونة وقابلية للتكيف مع المتغيرات.

وينتج عن استخدا مالبنية المعمارية النظيفة نظام له الخواص التالية :

- مستقل عن إطار العمل
- مستقل عن آلية تخزين البيانات
- نظام قابل للاختبار



الشكل 13 : مكونات البنية المعمارية النظيفة

2.2.7 الاعتمادية في البنية المعمارية النظيفة

التبعية في البنية المعمارية النظيفة هي علاقة اعتماد بين مكونات النظام المختلفة .تحدد هذه العلاقة كيفية تفاعل هذه المكونات مع بعضها البعض وكيف تؤثر التغييرات في مكون واحد على المكونات الأخرى .تعتبر قاعدة التبعية، التي تنص على أن التبعيات يجب أن تتجه من الخارج إلى الداخل، مبدأ أساسي في هذه البنية .هذا يعني أن المكونات الخارجية تعتمد على المكونات الداخلية، ولكن ليس العكس .تساهم هذه القاعدة في فصل الاهتمام، وزيادة مرونة النظام، وتسهيل صيانته وتطويره .من خلال إدارة التبعيات بشكل صحيح، يمكن للمطورين بناء أنظمة برمجية أكثر استقرارًا وقابلية للتوسع.

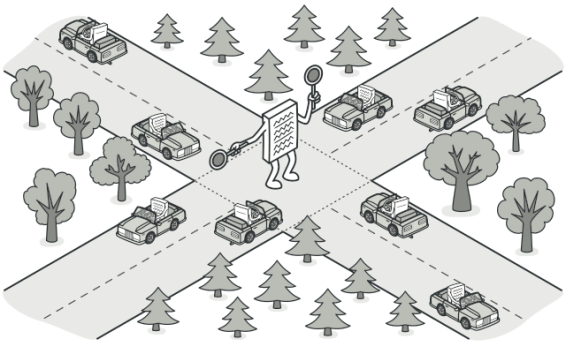
3.2.7 مكونات البنية المعمارية النظيفة

- الكيانات Entities : تعد الكيانات الأساس الذي يقوم عليه منطق الأعمال داخل النظام. الكيانات هي النماذج التي تعبر عن قواعد العمل الأساسية، وهي مصممة لتكون مستقلة عن التقنيات أو الأطر الخارجية.
- حالات الاستخدام Use Cases : تحدد حالات الاستخدام سلوك النظام من وجهة نظر المستخدم .وهي عبارة عن عمليات تجريبية تصف كيف يتفاعل المستخدم مع النظام لتحقيق هدف معين.وهي مستقلة عن الواجهات الخارجية والتقنيات.
- الواجهات Interfaces : تعمل الواجهات كعقود بين الطبقات المختلفة، حيث تحدد مجموعة من العمليات التي يمكن للطبقات الأخرى استخدامها للتفاعل مع هذه الطبقة دون الحاجة إلى معرفة التفاصيل الداخلية لتنفيذ هذه العمليات.
- أطر العمل والمحركات Frameworks and Drivers : تمثل أطر العمل والمحركات الطبقة الخارجية للبنية التي توفر الأدوات والبنية التحتية المكلوبة للتشغيل مثل أطر عمل الوب ومكتبات واجهة المستخدم وقواعد البيانات وغيرها.

3.7 النمط التصميمي الوسيط Mediator Design Pattern

النمط التصميمي **الوسيط (Mediator)** هو نمط تصميمي سلوكي (Behavioral Design Pattern) يهدف إلى تنظيم التفاعلات المعقدة بين الكائنات في نظام برمجي. في كثير من الأحيان، تكون العلاقات بين الكائنات في نظام برمجي معقدة، حيث يتواصل كل كائن مع العديد من الكائنات الأخرى. هذا يؤدي إلى وجود تشابك كبير بين الكائنات ويجعل من الصعب تعديل أو صيانة النظام.

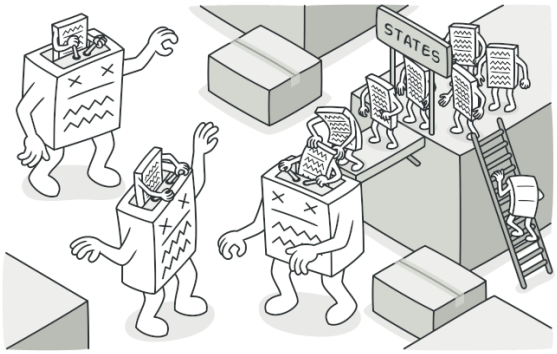
ويأتي دور النمط الوسيط لتبسيط هذا التعقيد عن طريق إدخال كائن وسيط واحد يتحكم في جميع التفاعلات بين الكائنات. بدلاً من أن تتواصل الكائنات مباشرة مع بعضها البعض، يتم توجيه جميع التفاعلات من خلال الوسيط. هذا يقلل من التبعيات بين الكائنات ويجعل النظام أكثر تنظيماً وأسهل في التعديل والتوسيع.



الشكل 14 : مثال توضيحي عن النمط الوسيط

4.7 النمط التصميمي الحالة State Design Pattern

و أحد أنماط التصميم السلوكية (Behavioral Design Pattern) التي تسمح لكائن بتغيير سلوكه بشكل ديناميكي بناءً على حالته الداخلية .بمعنى آخر، يتيح هذا النمط لكائن أن يكون له سلوكيات مختلفة في أوقات مختلفة، اعتمادًا على السياق الذي يوجد فيه.



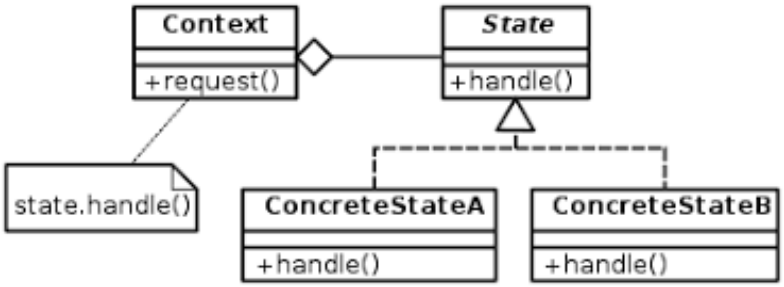
الشكل 15 : مثال توضيحي عن نمط الحالة

مكونات نمط التصميم حالة:

السياق :(Context) هو الكائن الذي يحافظ على الحالة الحالية ويوفّر واجهة للعمليات التي يمكن تنفيذها على الكائن.

الحالات :(States) تمثل كل حالة سلوكًا محددًا للكائن.

الواجهة :(Interface) تحدد واجهة مشتركة لجميع الحالات، وتصف العمليات التي يمكن تنفيذها في كل حالة.



الشكل 16 : مخطط صفوف تجريدي يصف نمط الحالة

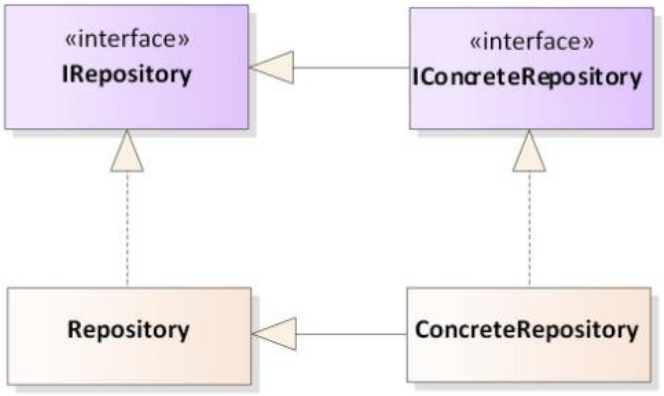
5.7 فصل مسؤوليات الأوامر والاستعلامات Command Query Responsibility Segregation

هو نمط تصميم يفصل بين العمليات التي تعدل البيانات (الأوامر) والعمليات التي تستعلم عنها (الاستعلامات) .بدلاً من استخدام نموذج واحد للبيانات للقراءة والكتابة، يقسم CQRS النظام إلى نموذجين منفصلين :نموذج للكتابة (Write Model) ونموذج للقراءة .(Read Model) هذا الفصل يوفر العديد من المزايا، بما في ذلك تحسين أداء القراءة والكتابة، زيادة قابلية التوسع، وتسهيل إدارة البيانات المعقدة .من خلال فصل المسؤوليات، يصبح من الممكن تحسين تصميم كل نموذج بشكل مستقل، وتطبيق استراتيجيات تخزين واسترجاع مختلفة لكل منهما.

6.7 نمط التصميم المستودع Repository Pattern

هو نمط تصميمي يهدف إلى فصل منطق الوصول إلى البيانات (Data Access Logic) عن منطق العمل في التطبيق (Business Logic). يوفر هذا النمط طبقة تجريدية تفصل بين مصدر البيانات (مثل قواعد البيانات أو خدمات الويب) وبين الطبقات التي تعتمد على البيانات، مما يعزز مبدأ فصل الاهتمامات(Separation of Concerns) .

يقوم المستودع (Repository) بتجميع منطق الوصول إلى البيانات في مكان واحد، مما يسهل إدارة التغييرات على مصدر البيانات دون الحاجة إلى تعديل منطق العمل.



الشكل 2 : مخطط صفوف تجريدي للنمط المستودع

7.7 نمط التصميم Result Pattern

الفصل الثامن

تصميم النظام

بعض المخططات لبنية النظام.

1.5 مبادئ التصميم

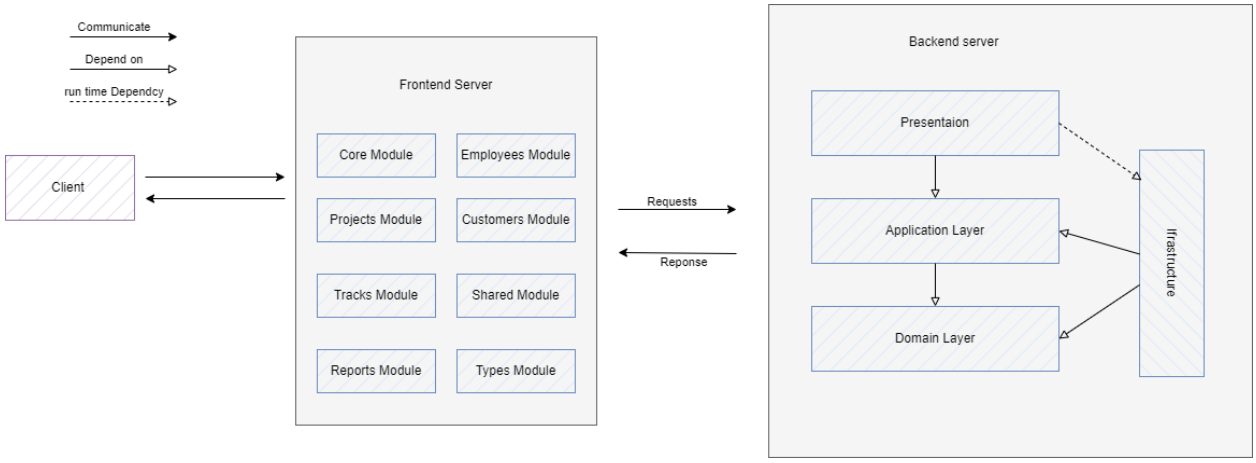
إن بناء تصميم جيد وقابل للتوسع يتحقق عندما نضع أسسا ومبادئ واضحة لهيكلة الحل وتقسيمه وطريقة واضحة للشروع في تنجيزه.

بناء على الدراسة التحليلية, وفهم آلية سير العمل تبين لنا أن النظام يمكن فصله إلى عدد من المجالات المستقلة التي تتكامل مع بعضها البعض لتحقق المطلوب لذلك إن الحل الذي سنعرضه وننجزه قائم على مبداء فصل المكونات ضمن كل الطبقات إلى ستة مجالات رئيسية هي مجال المشاريع ومجال المتابعة ومجال إدارة المراحل ومجال إدارة خطة الانفاق ومجال الجهات الطارحة و الموظفين. إذن إن المبداء الأساس الذي انطلقنا مه هو بفصل هذه المجالات في كل طبقات التصميم ومن خلال هذا الفصل بين المجالات سنستطيع بناء نظام مرن قابل للتوسع.

2.1.5 بنية عامة للتصميم

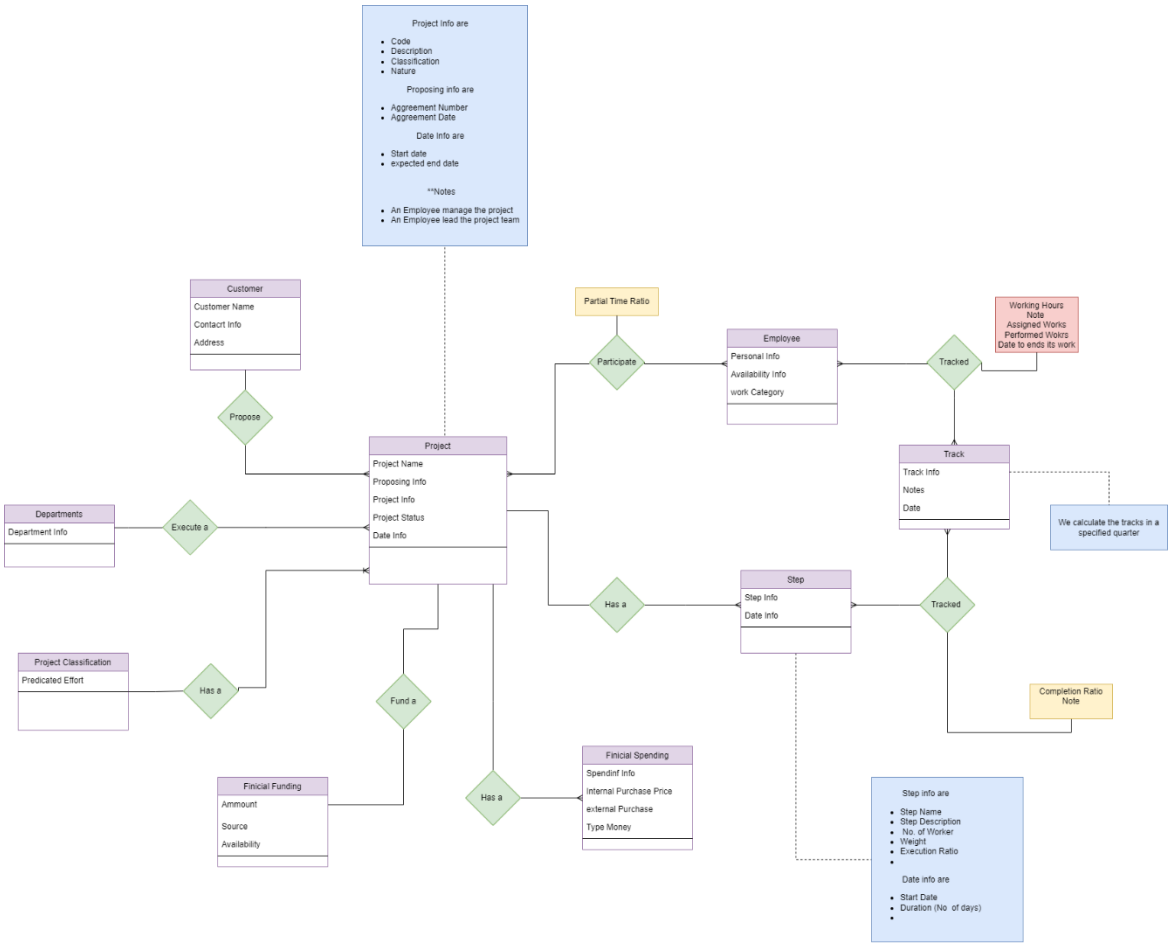
يبين الشكل (X) شكل مبسطا يوضح مكونات النظام, حيث يتكون النظام من جزئين تطبيق الواجهةالخلفية (Backend) الذي اعتمدنا في تصميمه وتنجيزه على البنية المعمارية النظيفة (Clean Architecture) وبعض مبادئ التصميم المقاد بالمجال (Domain Driven Design), والجزء الثاني هو طرف لواجهة الأمامية واعتدمنا في تصميمه على مبداء التقسيم وفق الميزات (Feature Based Decomposition) حيث قسمنا طرف الواجهة الأمامية إلى مجموعة من الوحدات (Modules) المستقلة بعضها عن بعض.

يتواصل طرف الواجهة الأمامية من الخلفية عن طريق النقاط الطرفية API Endpoints عبر طلبات HTTP وتنقل الحالة من وتمثل عبر REST API



الشكل 3 : بنية عامة للتصميم المقترح

2.8 مخطط العلاقات بين الكائنات Entity Relationship Diagram



الشكل 4 : مخطط العلاقات بين الكيانات**ERD**

الفصل التاسع

الأدوات وأطر العمل المستخدمة

نبين في هذا الفصل ماستخدمناه من أدوات وأطر عمل.

1.1 إطار العمل Net Core.

هو إطار عمل مفتوح المصدر ومتعددة المنصات، تم تطويرها بواسطة Microsoft لإنشاء تطبيقات حديثة وعالية الأداء وقابلة للتطوير. يتميز NET. بتحسينات كبيرة من حيث السرعة واستخدام الذاكرة مقارنةً بـ NET Framework. التقليدي، مما يجعله خيارًا مثاليًا لتطبيقات السحابة، الويب، والأنظمة الموزعة.

2.2 إطار العمل Angular

هو إطار عمل مفتوح المصدر لتطوير تطبيقات الويب تم إنشاؤه وصيانته من قبل Google. يعتمد على لغة TypeScript ويوفر بنية قائمة على المكونات، حيث يتم تقسيم التطبيق إلى وحدات صغيرة قابلة لإعادة الاستخدام، مما يسهل إدارة وتوسيع التطبيقات. يتيح Angular ربط البيانات ثنائي الاتجاه، مما يضمن التحديث المتزامن بين النموذج وواجهة المستخدم. كما يتضمن نظام توجيه متقدم لتطوير تطبيقات الصفحة الواحدة (SPAs) وخدمات حقن التبعية التي تعزز من قابلية الصيانة واختبار التطبيق. بالإضافة إلى ذلك، توفر أداة Angular CLI دعمًا متكاملاً لإعداد المشروع، بناء التطبيق، وإجراء الاختبارات، مما يعزز من إنتاجية المطورين.

3.2 قاعدة المعطيات SQL Server

هو نظام إدارة قواعد بيانات علائقية (Relational Database Management System) تم تطويره وتسويقه من قبل شركة Microsoft. يعمل SQL Server على نظامي التشغيل Windows وLinux، ويوفر سعة تخزين عالية مع أداء متميز في استعادة البيانات. يتميز SQL Server بقدرته على التعامل مع كميات كبيرة من البيانات بكفاءة عالية، مما يجعله مناسبًا للاستخدام في المؤسسات التي تتطلب إدارة بيانات قوية وموثوقة.

4.2 نمط الخيارات Option Pattern

في إطار العمل .NET يُستخدم لإدارة إعدادات التطبيق (appsettings) داخل بيئة العمل. يساعد هذا النمط في استخراج وتعديل القيم الثابتة المضمنة بشكل مباشر في الكود (hard-coded) بطريقة منظمة ومرنة. من خلال استخدام هذا النمط ، يمكن تحميل إعدادات التطبيق من ملفات التكوين، في كائنات ذات بنية منظمة. هذا يسهل تعديل الإعدادات دون الحاجة إلى تعديل الكود البرمجي مباشرة، مما يعزز من قابلية صيانة وتحديث الإعدادات بشكل أكثر أمانًا وفعالية.

5.2 JSON Web Token

هو معيار مفتوح يُستخدم لنقل المعلومات بين طرفين بشكل آمن عبر الويب باستخدام غرض JSON. يتميز JWT بقدرته على تقديم آلية مصادقة لا تعتمد على الجلسات (stateless authentication)، مما يجعله مناسبًا للتطبيقات الحديثة. على عكس ملفات تعريف الارتباط (cookies)، كما يمكن استخدام رموز JWT في المصادقة عبر تطبيقات الويب، تطبيقات

الهواتف المحمولة، وتطبيقات سطح المكتب. بفضل هذه المرونة، أصبح JWT خيارًا شائعًا لضمان أمان الاتصال بين المستخدمين والخوادم في بيئات متعددة.

5.2 المكتبة MediatR

هي أداة فعّالة تمكّن من تطبيق نمط التصميم الوسيط (Mediator Pattern) بشكل سهل وبسيط في بيئة .NET. فهي تعتد على مبدأ فصل إرسال الطلبات عن معالجتها، مما يسهم في تقليل التبعيات بين مكونات النظام ويسهل إدارة الرماز وتطويره . MediatR تسهل تنفيذ نمط CQRS (Command Query Responsibility Segregation) أيضًا، حيث يمكن فصل الأوامر (Commands) عن الاستعلامات (Queries) بطريقة منظمة وفعّالة.

الفصل العاشر

تنجيز النظام

نقدم في هذا لافصل شرحا مفصلا لمنهجية تنجيز النظام.

1.5 منهجية تصميم النظام

في تصميم الأنظمة البرمجية، تُعد منهجية التصميم الجيدة أمرًا حيويًا لضمان بناء نظام قوي ومرن وقابل للتوسع والصيانة. أحد هذه المنهجيات هي البنية المعمارية النظيفة (Clean Architecture)، التي تعتمد على مبدأ فصل منطق العمل (Business Logic) عن طرق العرض وآليات التخزين, مما يوفر بنية قابلة لإعادة الاستخدام والصيانة بسهولة.

تهدف هذه المنهجية إلى بناء نظام يكون مستقلًا عن البنى التحتية مثل قواعد البيانات أو واجهات المستخدم، حيث يركز التصميم على المجالات الأساسية للنظام (Core Domains). يتم تنظيم النظام في عدة طبقات، بحيث تكون كل طبقة مستقلة عن الأخرى وتعتمد فقط على الطبقات الداخلية لها، مما يسهل عملية التطوير ويعزز من مبدأ التماسك العالي (High Cohesion) والترابط المنخفض (Low Coupling) بين مكونات النظام.

وبالنسبة لطرف الواجهة الأمامية فاعتمدنا على منهيجة التقسيم وفق الميزات مما يساعد على جعل التطبيق قابللا للفهم والصيانة ومرنا.

2.5 البنية المعمارية لطرف المخدم

1.2.5 طبقات الحل المقترح

لقد اعتمدنا في تصميم طرف المخدم على البنية المعمارية النظيفة, لذلك نجد أن الحل لامقترح هو حل مركز حول المجال ومنطق العمل ومستقل عن طرق العرض و طرق تخزين البيانات.

يتكون طرف المخدم لدينا من خمسة طبقات هي :

1. طبقة النواة المشتركة

تحتوي هذه الطبقة على مفاهيم تجريدية عامة صالحة للاستخدام في أي مشروع يعتمد على البنية المعمارية النظيفة. وتمثل هذه الطبقة أساس التجريد الذي سيستخدم من قبل بقية الطبقات.

2. طبقة المجال

تحتوي هذه الطبقة على المجالات الستة التي ذكرناها في مبادئ التصميم وفي كل مجال تغلف فيه قواعد العمل (Business Rules) وهذه الطبقة لاتعتمد عى أي طبقة أخرى سوى طبقة النواة المشتركة التي تحوي المفاهيم التجريدية.

3. طبقة التطبيق

تعرف هذه الطبقة حالات الاستخدام حيث تعرف لكل مجال حالات الاستخدام ضمنه أي تعرف الاستعلامات والأوامر ضمن المجال, كما تحوي على تعاقدات (Interfaces) لعمليات يتم حقن تنجيزها الفعلي وقت الطلب. وتعتمد هذه الطبقة فقط على طبقة المجال.

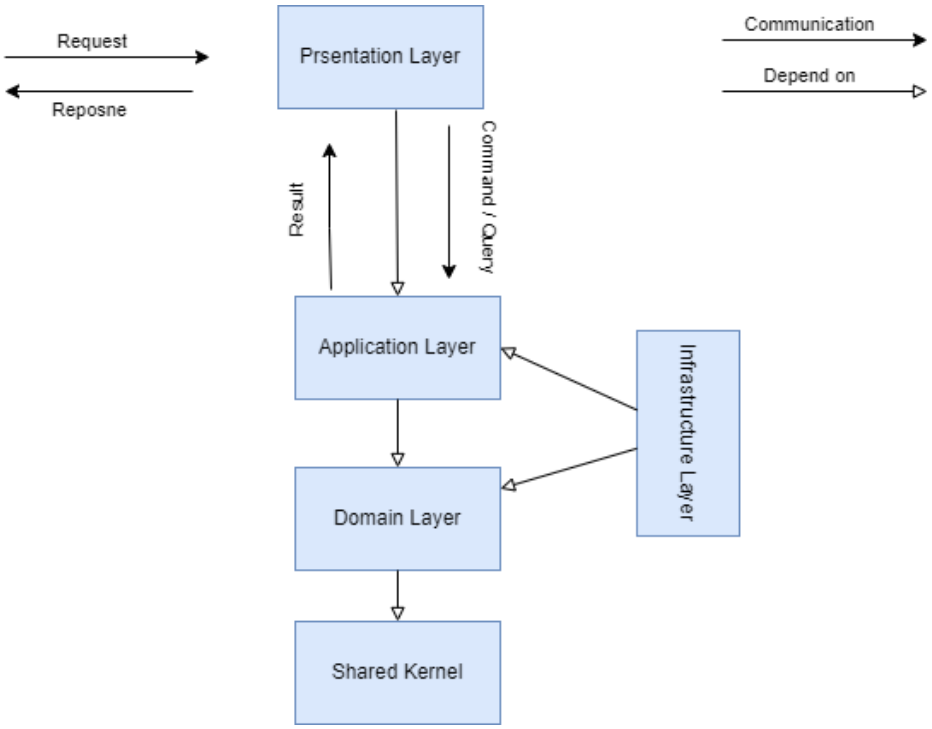
4. طبقة العرض

تعرف هذه الطبقة التعاقدات للطلبات والردود التي يتعامل معها الطرف الخارجي, وهي تعتمد على طبقة التطبيق

5. طبقة البنية التحتية

تعتمد هذه الطبقة على طبقة المجال وطبقة التطبيق وفيها يتم تنجيز آلية تخزن البيانات ضمن قاعدة المعطيات أو ضمن أي طريقة تخزين أخرى, كما تعرف هذه الطبقة تنجيزا للعقود في طبقة التطبيق.

ويبين الشكل أدناه توضيحا لتوضع الطبقات والاعتمادات بينها وطرق التواصل فيما بينها

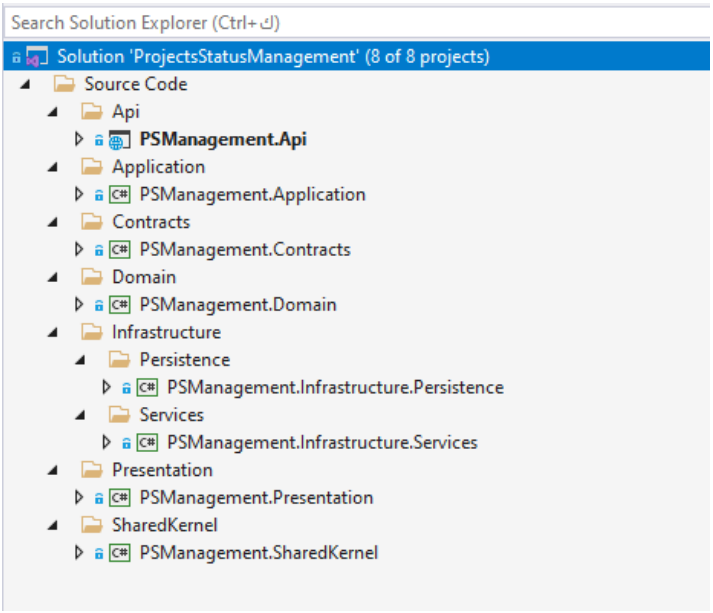


الشكل 5 البنية المعمارية لطرف الواجهة الخلفية

2.2.5 هيكلية مجلدات طرف المخدم

يتكون الحل لدينا من مجلد الرماز المصدري و مجلد الاختبارات "لم ينجز بعد يمكن أن أنجزه"

ويتكون الرماز المصدري للحل من ثمانية مشاريع تحقق المطلوب وهي كما هو مبين في الشكل أدناه من خلالها تم تحقيق الحل المطلوب.



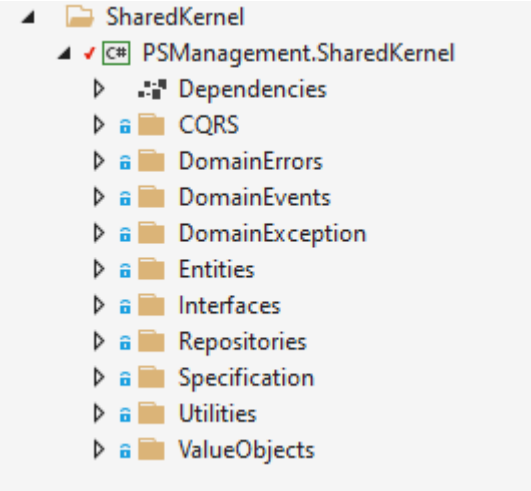
الشكل 6 هيكلية الحل المنجز

3.2.5 طبقة النواة المشتركة Shared Kernel

تحوي هذه الطبقة على مفاهيم تجريدية صالحة للاستخدام في أي مشروع يتبع البنية المعمارية النظيفة ومبادئ التصميم المقاد بالمجال. فهي تحوي على مفاهيم مجردة ولاتحوي على منطق عمل وأنما تجريد للمفاهيم فقط.

ويوضح الشكل أدناه مكونات الطبقة حيث تحوي على

- أساس الخطاء الذي يستخدم في نمط النتيجة
- أساس الكائنات Entity Base
- واجهات الأحداث ومعالجاتها IDomain Events & IEvents Handler
- تجريد النمط CQRS فهي تحوي الوجهات ICommand & ILoggable Command وكذلك بالنسبة للاستعلامات.
- أساس المستودع الذي يعرف عقود العمليات الأساسية CRUD



الشكل 7 : بنية طبقة النواة المشتركة

3.2.5 طبقة المجال Domain Layer

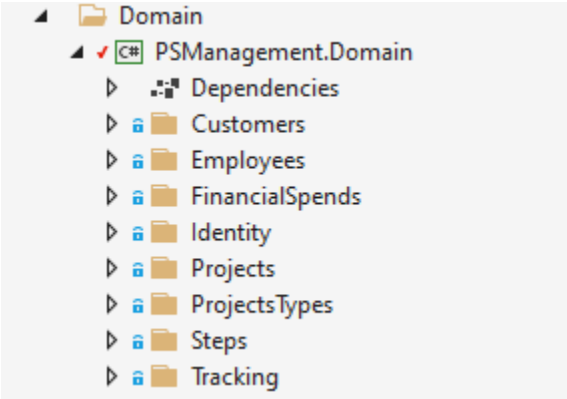
1.3.2.5 الهيكلية العامة للطبقة

تعتبر طبقة المجال الطبقة الأهم والأساس في هذه البنية المعمارية حيث تتمحور الاعتماديات عليها وهي لاتعتمد إلا على طبقة النواة المشتركة وفي هذه الطبقة نعرف قواعد العمل والكائنات و الاخطاء الأحداث. وكما ذكرنا في فقرة مبادئ التصميم التي سنتبعها قسمنا هذه الطبق إلى مجالات منفصلة وهي كما يلي:

- مجال المشاريع نهتم فيه بتعريف المشروع وإجرائيته وإدارة مرفقاته و إدارة أطواره.
- مجال أنواع المشاريع نعرف فيه أنواع المشاريع.
- مجال الجهات الطارحة ندير فيه الجهات الطارحة ومعلومات التواصل معها.
- مجال الموظفين نعرف فيه البيانات التي تهمنا ضمن مسألتنا عن الموظف والتي نكاملها ونأخذها من نظام الذاتية.
- مجال المتابعة نهتم فيه بمتابعة المشروع و المراحل وعمل الموظفين ونغلف فيه إجرائية المتابعة.
- مجال المراحل نهتام فيه بتعريف المراحل وإدارتها.
- مجال خطة الانفاق ندير فيه خطة الانفاق ومعلوماتها.

كما يوجد مجلد الهوية Identity نعرف في المستخدمين وصلاحياتهم وأدوارهم.

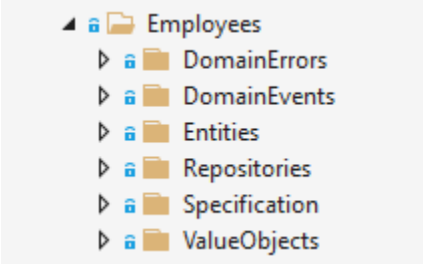
ويبين الشكل أدناه هذه المجالات.



الشكل 8 : بنية طبقة التطبيق.

وفي كل مجال من هذه المجالات يتكون من ستة مجلدات وهي كما يلي

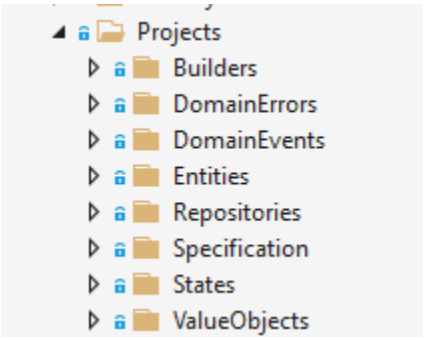
- الكائنات Entities : نعرف فيه الكائنات ضمن المجال وهي تغلف قواعد العمل وإصدار الأحداث.
- الأحداث Events : حيث نعرف الأحداث ضمن هذا المجال و ندع للكائنات إصدارها وفق لقواعد العمل وتساعد هذه الأحداث باستخراج عمليات النظام التي لاتنتمي لسيناريو العمل الأساسي ومعالجتها في مرحلة لاحقة كما في حالة تسجيل تبدلات العامل عند تغير مشاركته.
- الاخطاء Errors : نعرف فيه الاخطاء ضمن المجال وهي اخطاء تحقق Validation Errors تستخدمها طبقة التطبيق.
- المستودعات Repositories : وهو يجرد عقود عمليات تخزين البيانات واسترجاعها فهو يمكننا من الوصول إلى البيانات والتعامل معها بغض النظر عن طريقة تخزينها الفعلية.
- أغراض القيمة Value Objects : يحتوي على أغراض القيمة التي تساعد في تجميع واصفات الكائن على أسس مشتركة وجعلها كقيمة واحدة تتغير سوية.
- التخصيص Specification : يحوي هذا المجلد على تعاريف لاسترجاع البيانات من المستودع كيفما نريد أي ترتيبها وفق خاصية ما أو تجميعها أو تطبيق التصفيح عليها (pagination).



الشكل 9 : البنية الضمنية في طبقة التطبيق

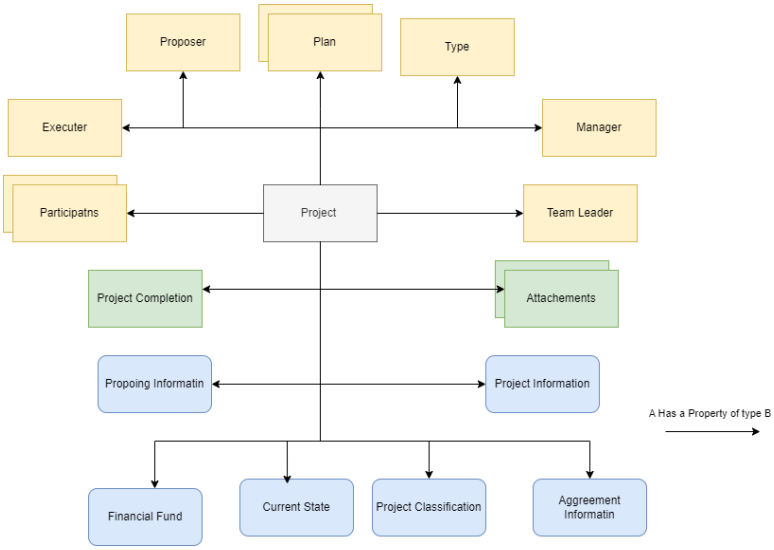
2.3.2.5 مجال المشاريع

إن هذا المجال يمثل أساس النظام أي هو المحور الأكبر في هذا النظام. وعلى عكس بقية المجالات فهو يحوي عددا منالمجلدات لاتوجد في البقية. سنقوم في شرحها هنا.



الشكل 10 : بنية مجال المشاريع

هيكلية الصف Project يبين الشكل أدناه الصف Project وواصفاته من أغراض قيمة (ذات اللون اأزرق) و كائنات ضمن المجال (ذات اللون الأخضر) وكائنات من مجال آخر (ذات اللون الأصفر)



الشكل 11 : رسم توضيحي لمجال المشروع.

ويتكون هذا من عدد من أغراض القيمة مثل معلومة طرح المشروع Project Proposing ويتكون غرض القيمة هذا من خاصيتين هما رقم كتاب الطر و تاريخ كتاب الطرح, لماذا هذا غرض قيمة لأن على الرغم منأن كتاب الطرح وتاريخه معلوماتان مختلفتان ولكن لهما دالة وهي معلومة الطرح التي تتغير كليا بتغير أحداهما أي ليس غرضا ذا هوية على عكس كائن المرفق الذي يعرف بمعرف (Id) وتغيير عنوان المرفق لايغير المرفق بذاته.وهكذا نميز بين أغراض القيمة و الكائنات.

يحوي هذا المجال أيضا على مجلد الباني,

ففي هذا المجال أضطررنا إلى استخدام النمط التصميم الباني (Builder Design Pattern) يساعدنا هذا النمط في تقديم واجهة موحدة لبناء غرض من هذا الصف فهذا الصف يحوي على عدد كبير من الواصفات ونحن في طبقة المجل لانعلم أي الواصفات في الطبقات الأعلى تحتاجها لبناء الصف أي ليس من المنطقي كتابة بان constructor لكل مجموعة من الخواص التي نتوقع أن تحتاجها الطبقة الأعلى. لذلك استخدمنا هذا النمط الذي يساعد على أخذ غرض من كائن لديه واصفات عديدة لانعلم أي يحتاج الكائن الذي سيتعامل مع الغرض المنشاء. لذلك يكون الحل هو بكتابة بان واحدة يأخد جميع الواصفات وترك عملية اسناد القيم للصف Project Builder ويوضح الشكل أدناه طريقة استخدمنا لهذا البان.

```
Project project = _projectBuilder
    .WithProjectAggreement(request.ProjectAggreement)
    .WithProjectInfo(request.ProjectInfo)
    .WithProposalInfo(request.ProposalInfo)
    .WithExecuter(request.ExecuterId)
    .WithFinancialFund(request.FinancialFund)
    .WithProjectManager(request.ProjectManagerId)
    .WithTeamLeader(request.TeamLeaderId)
    .WithProposer(request.ProposerId)
    .WithClassification(request.ProjectClassification)
    .Build();
```

الشكل 12 : مثال على استخدام الصف الذي يحقق نمط الباني

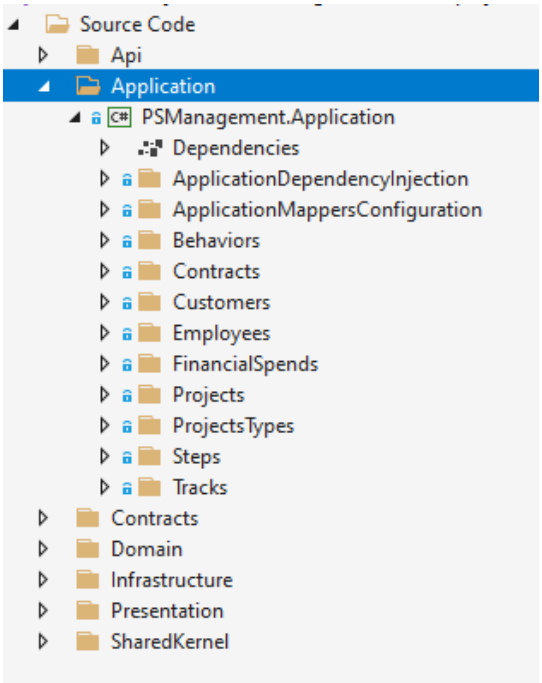
كما في هذا المجال استخدمنا النمط التصميمي الحالة لإدارة أطوار المشروع والانتقالات بينها.حيث بين الشكل (x) مخطط الصفوف المستخدم.حيث عرفنا صف مجردا يحوي الطرائق التي تعرف الانتقال بين الحالات Complete, Plan , Approve, Propose ,Cancel ,Complete وتترك مسؤولية التنجيز الفعلي إلى الحالة المعنية التي يقع على عاتقها استخدام طرائق الصف Project لتحقيق العملية.

```
1 reference
public Result Complete(ProjectCompletion projectCompletion)
{
    return State.Complete(this, projectCompletion);

}
1 reference
public Result Plan()
{
    return State.Plan(this);

}
```

الشكل 13 : مثال على آلية تغليف الحالة لمؤثر الانتقال

4.2.5 طبقة التطبيق Application Layer

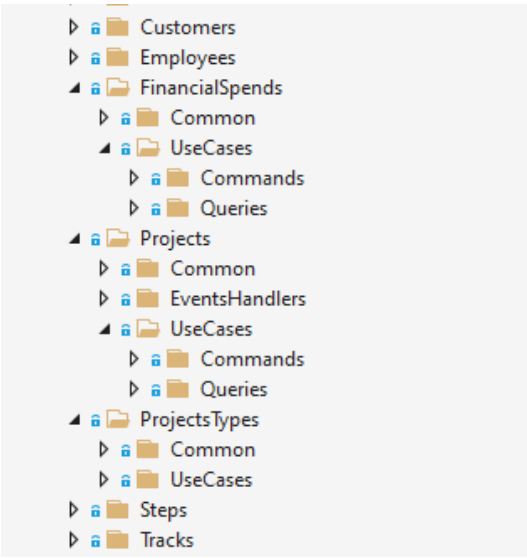
مسؤولية هذه الطبقة هي تغليف طبقة المجال وإضافة منطق العمل كما تعرف حالات الاستخدام التي تتيحها للطبقات الأعلى.



الشكل 14 :بنية طبقة التطبيق.

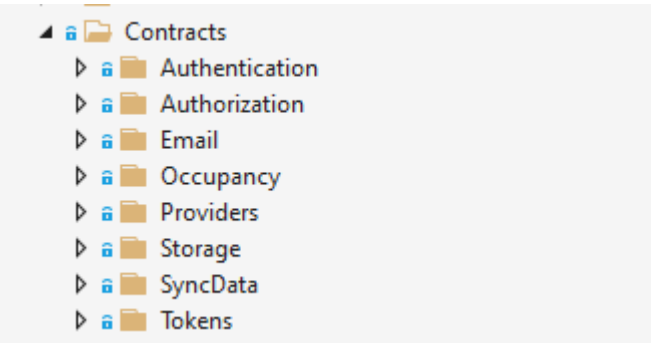
في هذه الطبقة نجد لدينا شقين أساسيين هما

الشق تغليف طبقة المجال بحالات الاستخدام التي تغلف منطق العمل حيث نجد سبعة مجلدات تمثل المجالات ضمن التطبيق وهي المتابعة, المرحل, المشاريع, أنواع المشاريع, الجهات الطارحة, إدارة الانفاق, الموظفين. ويحوي كل مجال من هذه المجالات ثلاث مجلدات وهي مجلد المشترك (Common) الذي يعرف الرد أي النتيجة التي تحصل عن استعلام ما. ومجلد معالجات الأحداث (Event Handlers) الذي يحوي معاجة لبعذ الأحداث التي يجب معالجتها في هذه الطبقة والتي تصدر عن طبقة التطبيق. وأيضا يوجد مجلد حالات الاستخدام (Use Cases) الذي يحوي الاستعلامات والأوامر وفيه نغلف منطق العمل.



الشكل 15 : البنية الضمنية في كل مجال.

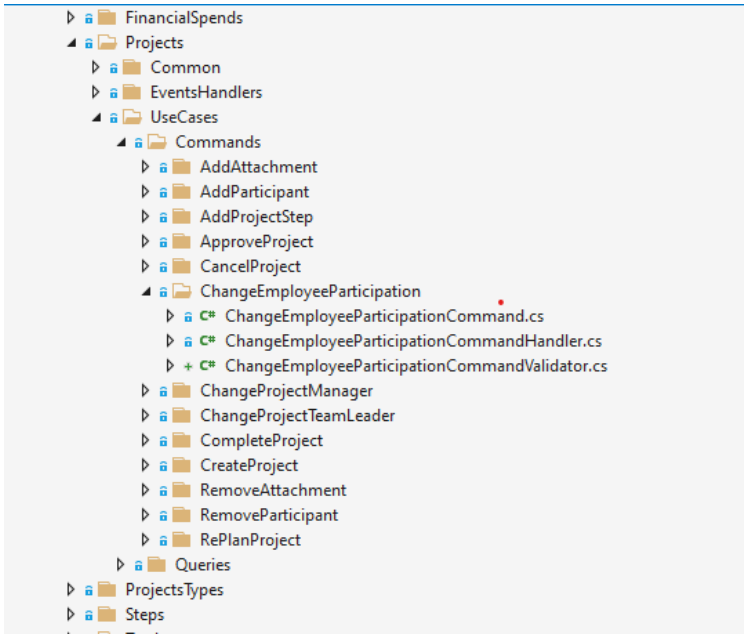
والشق الثاني تعريف لعقود خدمات (Interfaces) يتم التعامل معها دون الاهتمام بالتنجيز الفعلي مما يساعد على جعل منطق العمل في بعض الحالات مستقلا عن طريقة التنجيز الفعلية.



الشكل 16 : العقود ضمن طبقة التطبيق.

استخدمنا في هذه الطبقة لتحقيقها النمط التصميمي Command Query Responsibility Segregation (CQRS) حيث فصلنا حالات الاستخدام إلى جزئين أوامر Commands تغير حالة الكائنات واستعلامات Queries تعطي معلومات دون تغيير على أي حالة لأي كائن.

ويوضح الشكل أدناه مثالا على تحقيقه في حالة الاستخدام تغيير معلومات مشاركة موظف



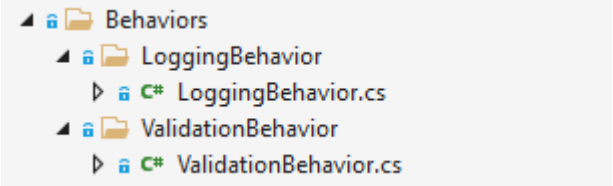
الشكل 17 : مثال على تنجيز حالة استخدام بالنمط **CQRS**

يتم تنجيز حالة استخدام ما بفصلها إلى شقين الأمر ومعالج هذا الأمر. يتم استخدام هذه الحالة من الطبقة الأعلى من خلال أرسال الطب ChangeEmployeeParticipationCommand بعد تحديد قيمه المطلوبة ضمنه (الرقم المعرف للموظف ومعرف الزبون ومعلومات المشاركة الجديدة) ويترتب على عاتق الطبقة من خلال النمط الوسيط (Mediator) أن يفعل الطريقة Handle الموجودة ضمن الأوامر ليتم تنفيذ المعالج. ولتحقيق النمط الوسيط استخدمنا المكتبة MediatR التي تحقق هذا النمط ويوضح الشكل أدناه آلية إرسال طلب باستخدامه من طبقة العرض.

```
GetProjectParticipantsQuery query = new(id);

var result = await _sender.Send(query);
```

وفي سياق النمط التصميمي الوسيط علينا أن نذكر أيضا أن المكتبة MediatR عندما قامت بتنجيزه أتاحت لنا تعريف خط تسلسل من السلوك Pipeline Behavior يقاطع الطلب ويحصل على سياق كائن الأمر أو الاستعلام قبل تسليمه للمعالج المسؤول عن معالجة الأمر أي باختصار قبل أن يصل الأمر ChangeEmployeeParticipationCommand إلى المعالج

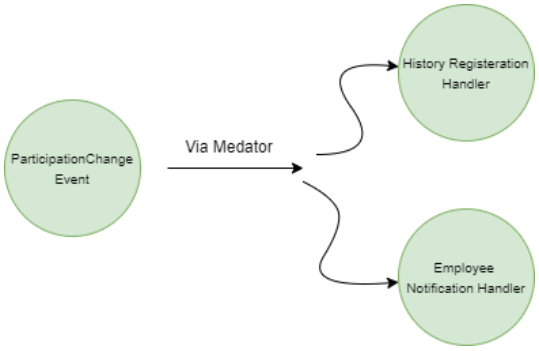
خاصته نستطيع تعريف تسلسل من السلوك الذي نريده أن ينفذ بسياق الطلب وفي حالتنا نحن قمنا بتعريف سلوكين عامين الأول هو سلوك التسجيل Logging Behavior المسؤول عن تسجيل الاستعلامات والأوامر, وسلوك أخر وهو سلوك التحقق Validation Behavior المسؤول عن التحقق من أهلية البيانات المدخلة أي لكي نمنع في حالة تعير مشاركة موظف من إدخال قيم سالبة لنسبة التفرغ وبعض الشروط الأخر التي عرفناها في الصف ChangeEmployeeParticipationCommandValidator ويوضح الشكل أدناه السوكين اللذان عرفناهما.



سلوك التسجيل مسؤوليته التحقق من أهلية البيانات الواردة. وسلوك التسجيل الذي يسجل الاستعلامات والأوامر المهمة.

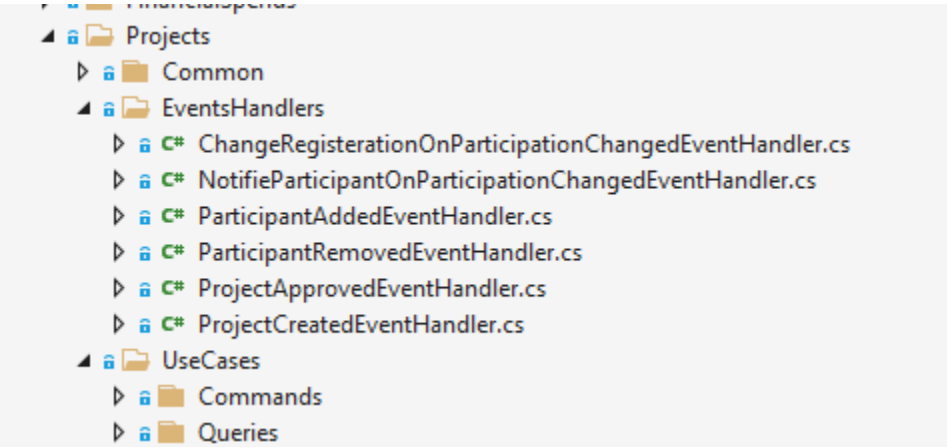
ومن أجل عدم تسجيل جميع الاستعلامات والأوامر لأن ذلك يخفف الأداء وليست جميع الأوامر والاستعلامات مهمة قمنا بفصل الاستعلامات إلى نوعين Loggable Commands و Commands وكذلك الأمر بالنسبة للاستعلامات وبهذه الطريقة استطعنا تعريف الاستعلامات والأوامر المهمة.

وفي هذه الطبقة وكما ذكرنا سابقا يوجد في كل مجال مجلد لمعالجات الأحداث.حيث وفي سياق حالة الاستخدام ذاتها ChangeEmployeeParticipation إن معالج هذه الحالة يستدعي طريقة التعديل التي يغلفها الكائن Project في طبقة المجال ومن ضمن تغليفه لآلية التعديل فهو ينشر حدث وهو EmployeeParticipationChanged وكل من يهمه أمر معالجته يستقبله من خلال أيضا النمط الوسيط الذي بدوره يسلمه للمعالج المناسب الذي يعلمه من خلال تنجيزه للواجهة IEventHandler وفي سياق هذا نوضح ماذا ستفدنا منه.



الشكل 18 : **Event Sourcing Example**

إن حدث تغيير معلومات مشاركة موظف في مشروع يتطلب منا إعلامه بأن مشاركته قد تغيرت ويتطلب من أيضا تسجيل معلومات المشاركة القديمة لنقدم للمشارك وللإدارة تقريرا بتاريخ مشاركته وتبدلاته. وعندما نريد فعل هذا يوجد طريقة بديهية ألا وهي القيام بهذه الأمور في معالج حالة الاستخدام ذاته قد يبدو هذا الحل وجيها ولكنه لايصح لأن جمع إجرائية إرسال الإعلام و إجرائية تسجيل تاريخ المشارك ضمن حالة الاسنخدام ذاتها يجعلها غير مترابطة فيما بينها وليست ذات مسؤولية واحدة أي يخرق مبداءان تصميميان هم High Cohesion و Single Responsibility لذلك قمنا بهذا الفصل الذي يجعمل مكونات معالجة حالة الاستخدام مترابطة ومستقلة عن الإجرائيات التي تتم عند وقوع الحدث.



الشكل 19 :بعض معالجات الأحداث في مجال المشاريع.

وأيضا من الأنماط المستخدمة في هذه الطبقة هو النمط Unit Of Work المعرف في طبقة النواة المشتركة ويستخدم هذا النمط لجعل حالة الاستخدام تنفذ أو تفشل كقطعة واحدة أي إما يتم تنفيذ كامل حالة الاستخدام بشكل صحيح أو تلغى كاملة.

يتم هذا النمط من خلال تعريف ثلاثة طرائق Commit, RoleBack, BeginTransaction حيث تكون العملية Begin Transaction عن بدء مناقلة معزولة وعند الانتهاء من حالة الاستخدام ننهي المناقلة أي تتطبق وهذا المقهوم مشابه لمفهوم المناقلات في قواعد المعطيات.

وأيضا تم استخدام نمط النتيجة Result Pattern المقدم من المكتبة Ardalis.Result للتحكم بسير العمل في هذه الطبقة عوضا عن استخدام الاستثناءات Exceptions حيث يقوم هذا النمط بتغليف القيمة المعادة وحالة النتيجة ويحوي على أخطاء قمنا بتعريفها بكل مجال. ويوضح الشكل أدناه آلية استخدامه.

```
if (!project.HasParticipant(request.ParticipantId)) {
    return Result.Invalid(ProjectsErrors.ParticipantUnExistError);
}
```

الشكل 20 : مثال عن استخدام نمط النتيجة

5.2.5 طبقة العرض

1.5.2.5 الهيكلية العامة

تتكون هذه الطبقة من ثلاثة مشاريع هي على النحو الآتي :

1. مشروع التعاقدات Contracts Project
2. مشروع العرض Presentation Project
3. مشروع واجهة التخاطب API Project

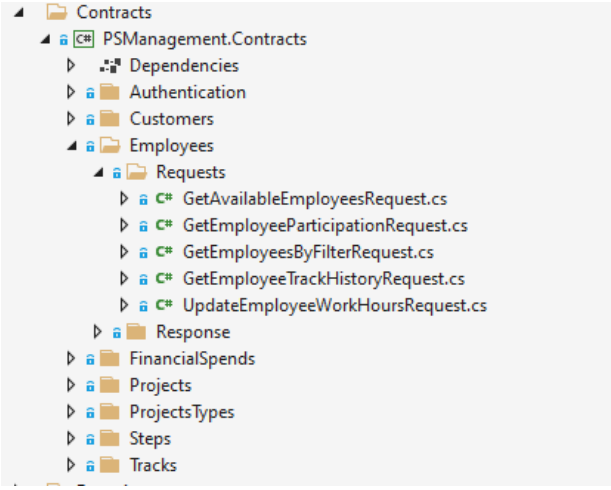
إن لكل من هذه المشاريع دوره الواضح والحاجة إليه والفائدة التي يقدمها,فمشروع التعاقدات يعرف الطلبات التي ترد إلى النظام والردود عليها, بينما يعرف مشروع العرض النقاط الطرفية في كل مجال ومهمته هي فقط أخذ الطلب وتحويله إلى أمر أو استعلام وتسليمه عن طريق النمط الوسيط إلى معالج الحدث ومن ثم إعادة النتيجة وتحويلها إلى رد.

بينما يقع على عاتق مشروع واجهات التخاطب تعريف الطبقات الوسيطة Middleware's وحقن الاعتمدايات المطلوبة حيث هو فقط الذي يستطيع أن يأخذ مرجعا من كل المشاريع فهو الذي يضيف التنجيز الفعلي للواجهات ممن خلال حقنه لاعتماديات طبقة النبية التحتية.

أ. مشروع التعاقدات

إن تعريفنا للطلبات الممكن إرسالها والردود النتاجة عنها يساعد في توثيق واجهة التخاطب بشكل جيد ويساعد على فهم النظام دون الحاجة والطرفيات دون الحاجة إلى الاطلاع على منطق العمل ممايسهل مكاملته مع طرف الواجهة الأمامية.

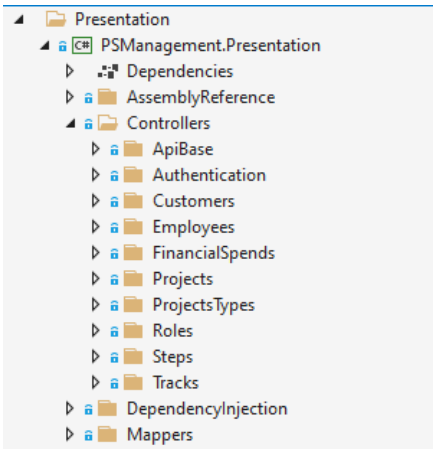
وهذا المشروع يحوي على ستة مجلدات تمثل المجالات الموضحة في مبادئ التصميم وفي كل مجلد يوجد مجلد للطلبات وآخر للردو كما هو مبين في الشكل أدناه.حيث يساعد هذا المشروع على تقديم طريقة واضحة لتوثيق النظام أي نستطيع من خلاله أن نعرف بشكل دقيق دون اطلاع أحد على طبقة التطبيق بجميع الطلبات الممكن إرسالها.



الشكل 21 :بنية مشروع التعاقدات

ب. مشروع العرض

يحوي على المتحكمات Controllers ويعرف التقابلات مع طبقة التطبيق ومهمته تختصر باستقبال الطلب وتحويله إلى أمر أو استعلام وإرساله عن طريق الوسيط إلى معالجه ومن ثم أخذ النتيجة وتحويلها إلى رد وإعادتها إلى الزبون.



الشكل 22 : بنية مشروع العرض

ت. مشروع واجهة التخاطب API Project

يمثل هذا المشروع آلية تشكيل وتهيئة النظام كاملا حيث يتم فيه حقن الاعتماديات التي عرفت في الطبقات الاخرى واختيار البنية التحتية الفعلية ويوجد فيه الطبقات الوسيطة التي تعترض الطلب قبل وصوله إلى المتحكم Controller ويوضح الشكل (X) مثالا على حقن الاعتماديات المعرفة في كل طبقة من الطبقات السابقة. حيث كل طبقة تحوي على صف حقن الاعتماديات (Dependency Injection) تعرف فيه الاعتماديات التي تستطيع تعريفها وفقا للكائنات الموجودة لديها وتقدم من خلاله واجهة للطبقات الأخرى التي تريد استخدامها

```
0 references
public void ConfigureServices(IServiceCollection services)
{
    // adding dependency injection
    services
        .AddAPI()
        .AddPresentation()
        .AddApplication()
        .AddPersistence(Configuration)
        .AddInfrastructureSevices(Configuration);
}
```

الشكل 23 : مثال على حقن الاعتماديات واختيار النبية التحتية

حيث كل طبقة من الطبقات السابقة عرفت فيها الاعتماديات التي تستطيع إضافتها, وبدوره نختار هنا الاعتماديات التي نريدها.

6.2.5 طبقة البنية التحتية

1.6.2.5 الهيكلية العامة

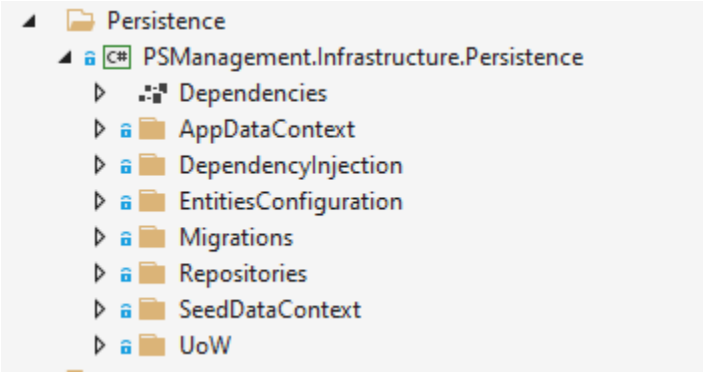
تم تحقيق هذه الطبقة من خلال مشروعين أولا مشروع Persistence Project وهو مسؤول عن التنجيز الفعلي لمستودعات البيانات وتخزينها. والثاني مشروع الخدمات Services Project حيث يتم فيه تعريف التنجيز الفعلي للعقود المعرفة في طبقة التطبيق.

2.6.2.5 مداومة البيانات Data Persistance

أ. هيكلية عامة

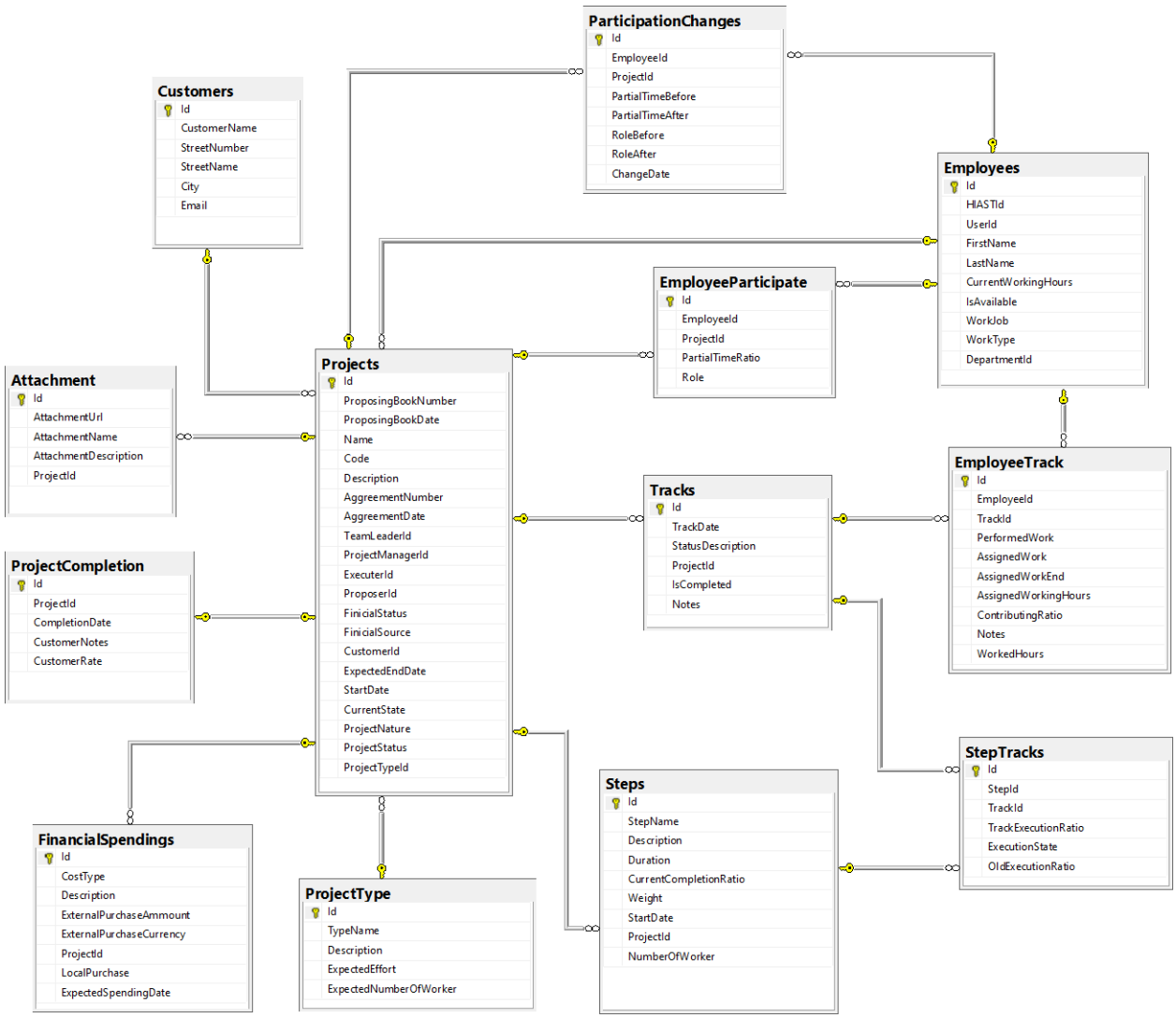
في هذا المشروع تم تعريف طريقة التخزين الفعلية للبيانات كما قمنا فيه بتنجيز المستودعات الموجودة عقودها في طبقة المجال.

ويوضح الشكل أدناه هيكلية المشروع.

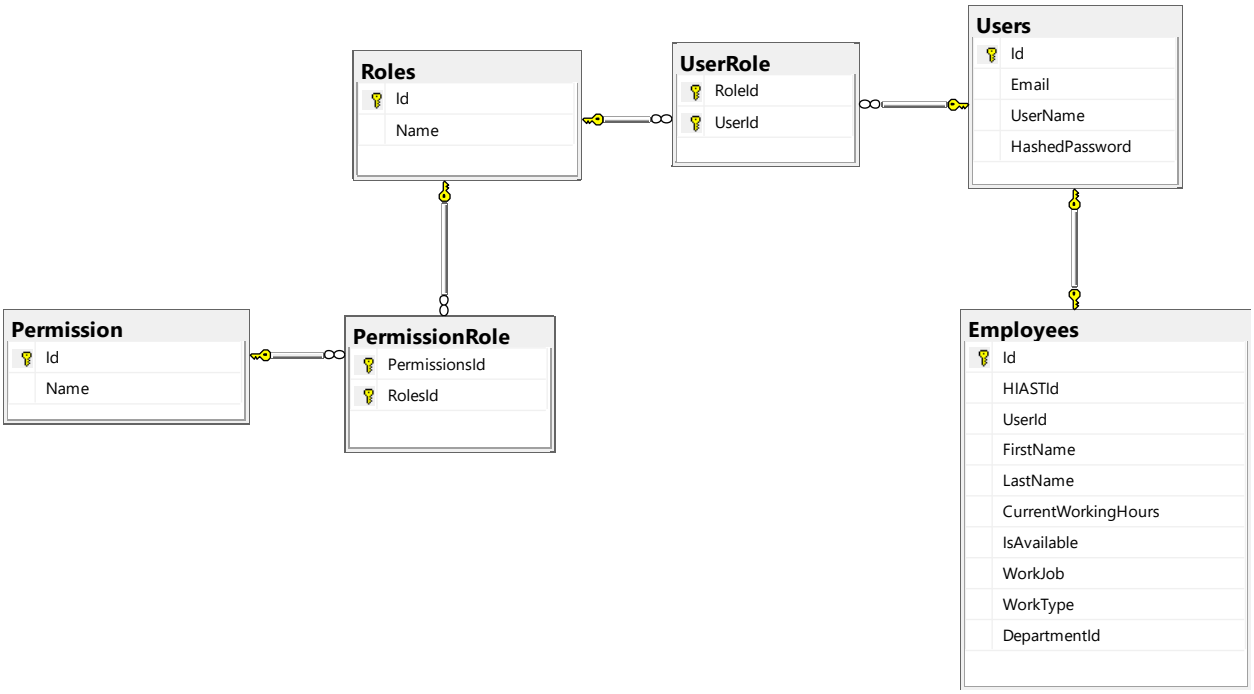


ب. تخزين المعطيات

جرى استخدام إطار العمل (Entity Framework) لتحويل الكائنات في طبقة المجال (Entities) إلى جداول علائقية (Relational Tables) ضمن قاعدة المعطيات التي استخدمناها وهي من النوع (SQL-Server) ويوضح الشكل أدناه مخطط العلاقات في هذه القاعدة.



الشكل 24 : مخطط العلاقات ضمن قاعدة المعطيات



الشكل 25 :العلاقات بين المستخدمين والأدوار والصلاحيات

من أجل تحقيق إدارة المستخدمين والصلاحيات يوجد نمطين من التحكم بالوصول Role Based Access Control و Permission Based Access Control

في النمط الأول نعرف مجموعة من الأدوار الثابتة ونسند لكل مستخدم مجموعة من الأدوار حسب الحاجة ونضع لكل طرفية اتصال API Endpoint الدور الذي يجب أن يمتلكه المستخدم الذي يطلب الطرفية وعلى أساسه بتم التحقق من إمكانية استخدامها.

وفي النمط الثاني نعرف مجموعة من الصلاحيات ونضع لكل طرفية اتصال الصلاحية المطلوبة لاستخدامها ونجمع الصلاحيات في أدوار أي كل دور له مجموعة من الصلاحيات أي نستطيع إضافة أدوار جديدة وإضافة الصلاحيات التي نريدها لها ونسند الأدوار للمستخدم.

مزايا وعيوب استخدام النمط الأول :

يسمح لنا بتعريف أدوار واضحة وثابتة, سهل الاستخدام والتنجيز

مساوؤه أنه يعرف أدوار ثابتة لانستطيع تعديل إمكانياتها

مزايا وعيوب النمط الثاني :

يسمح لنا بتعريف أدوار ديناميكية حسب الحاجة وتعديل الصلاحيات المسموحة لكل دور أي هو مرن كيفما نريد.

مساوؤه أننا لانتسطيع فهم الأدوار بشكل واضح حيث أنها تشكل بشكل تلقائي كيفما نشاء.

وفي حالتنا ولأن مشروعنا يحوي على أدوار واضحة لاحاجة لأن تكون مرنة قمنا باستخدام النمط الأول واتحنا لمن يريد التطوير عليه إمكانية استخدام النمط الثاني.

حيث عرفنا لدينا الأدوار الآتية :

المخطط وهو مسؤول عن إدارة أنواع المشاريع و الجهات الطارحة.

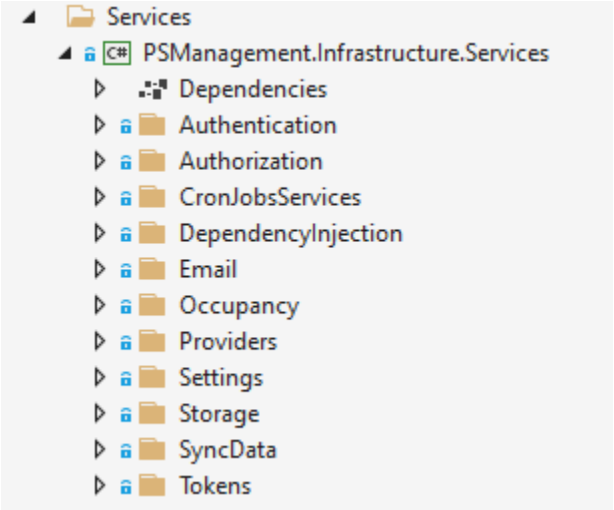
الموظف وهو يتمتع بصلاحيات تخطيط ومتابعة المشاريع.

الوكيل العلمي وهو يتمتع بصلاحيات طرح المشاريع إلغاءها واستعراضها جميعا.

مدير النظام وهو مسؤول عن إدارة المستخدمين وأدوارهم.

2.6.2.5 الخدمات Services

في هذا المشروع تم تنجيز العقود المعرفة في طبقة التطبيق.



الشكل 26 : بنية مشروع الخدمات

من الخدمات الاساسية التي نجد تنجيزها هنا هي :

أ. تنجيز خدمة حفظ الملفات واسترجاعها File Storage Service :

حيث عرفنا في طبقة التطبيق واجهة للتعامل مع حفظ الملفات واسترجاعها من أجل إدارة مرفقات المشروع. وفي هذه الطبقة قمنا بتنجيز التخزين الفعلي لها على نظام الملفات.

إن فائدة هذا الفصل بين الواجهة وتنجيزها هي أننا في طبقة التطبيق لانهتم في كيفية حفظ الملفات وأنما نهتم بحفظها واستعادتها فقط. لذلك هنا عرفنا طريق الحفظ في نظام ملفات ولو أردنا تغيير طريقة التنجيز ليتم الحفظ على السحابة فإننا يجب علينا التعديل هنا فقط دون التأقير على منطق العمل في بقية الطبقات.

ب. تنجيز خدمة تزويد بيانات الموظفين Employees Data Provider :

هذه الخدمة مسؤول عن احضار بيانات الموظفين التي تهمنا ومكاملتها مع مع قاعدة البيانات التي لدينا. إن تنجيز هذه الخدمة وهمي أي نقراء من مصد غير حقيقي لأن المصدر الرئيسي عن طريق ال LDAP لايتاح لنا كون مشروعنا طلابي. لذلك عند الرفع الفعلي للتطبيق الأمر المطلوب فقط هو تنجيز إحضار البيانات فقط لأن آلية مكاملتها معرفة في خدمات أخرى تنجيزها حقيقي. أي ليست من مسؤولية هذه الخدمة سوى احضار البيانات ويقع على عاتق خدمة أخرى فعلية مكاملتها.

ت. تنجيز خدمة مزامنة البيانات Sync Employee Data Service :

يقع على عاتقه تحقيق المكاملة الفعلية للبيانات التي نحضرها من خلال خدمة تزويد بيانات الموظفين التي تحدثنا عنها. حيث تقوم هذه الخدمة بجلب البيانات عن طريقة واجهة تزويد البيانات وتكاملها مع ماهو موجود لدينا عن طريق واجهة مستودع الموظفين. أي عرفنا آلية المكاملة ونجزناها تنجيزا فعليا والمطلوب هو فقط تحقيق الخدمة (ب)

ث. خدمة اشعار انشغالية العاملين Employee Occupancy Notification :

هذه الخدمة مسؤولة عن مزامنة بيانات انشغال الوظفين لدى نظامنا مع البيانات الموجودة على مخدم LDAP وهي أيضا بحاجة إلى تنجيز فعلي ولكن منطق العمل لدى نظامنا من إرسال اشعارات المزامنة مستقا عن التنجيز أي فقط في هذا المكان المستقل عن منطق وقواعد العمل يجب التنجيز.

ج. خدمات الفويض وتوليد المفردات Authorization and Token Generation :

تم تحقيق المصادقة Authentication من خلال رموز JWT التي تحدثنا عنها في فقرة الأدوات المستخدمة. حيث قمنا بتحميل بيانات الموظف من اسم وشهرة وأدوار في مفردة JWT ويتم التحقق من الصلاحيات عبر طبقة وسيطة تمر عبرها الطلبات قبل الوصول إلى المتحكم.

ومن أجل ضبط تهيئة توليد المفردة من مدة صلاحية ومفتاح تشفير تم استخدام نمط الخيارات لجعلها متغيرة يتم ضبطها من خلال ملف appsitteng.json.

كما يحوي هذا المشروع على أعمال مجدولة تنفذ كل فترة معين تحدد أيضا من خلال ملف appsitteng.json ومن هذه الاعمال مزامنة بيانات الموظفين.

3.5 بنية الواجهة الامامية Front-end

1.3.5 منهجية التصميم

تم بناء الواجهة الأمامية باستخدام إطار العمل Angular. ومن أجل تحقيق مبادئ التصميم التي ذكرناها سابقا ولفصل هذه المجالات في تطبيق الواجهة الأمامية لذلك اعتمدنا منهجية التقسيم القائم على الميزات (Feature Based Decomposition) حيث يقوم هذا النهج على تقسيم التطبيقات المعقدة والكبيرة إلى وحدات (Modules) مستقلة, تمثل كل منها ميزة محددة في التطبيق وهي تقابل في حالتنا مجالات عملنا حيث كل مجال هو ميزة مستقلة عن الآخر.

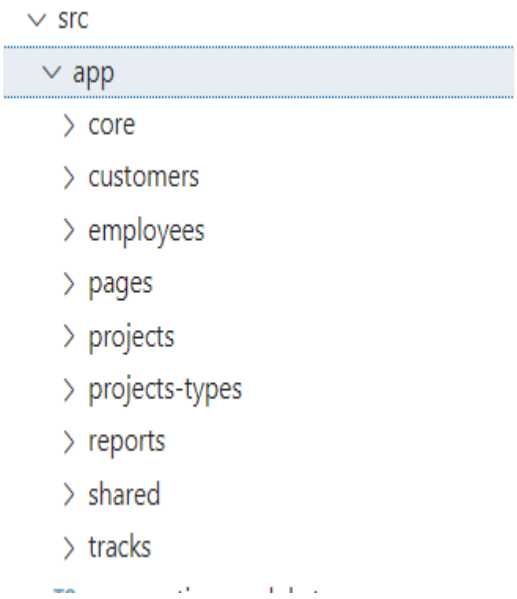
يهدف هذا النوع من التقسيم إلى تعزيز مبداء التماسك العالي High Cohesion والترابط المنخفض Low Coupling.

عند استخدام هذا التقسيم، يتم تحميل الوحدات بشكل ديناميكي(Lazy Loading) ، مما يعزز من أداء التصبيق.

2.3.5 بنية تطبيق الواجهة الأمامية

يحوي التطبيق على مجلدين اساسيين هما مجلد البيئة (Environment) الذي نعرف فيها ثوابت البيئة مثل عنوان طرف المخدم ومجلد المصدر (SRC) الذي يحوي الرماز المصدري.

في مجلد المصدر لدينا عدد من المجلدات كل مجلد هو عبارة عن وحدة (Module) مستقلة تحوي مكوناتها (Components) وخدماتها (Services) ونماذجها (Models) والتوجيه (Routing) والصفحات المعرفة بها. وهذه الهيكلية نفسها تتكرر لجميع الوحدات. ويوضح الشكل أدناه الوحدات الموجودة لدينا.

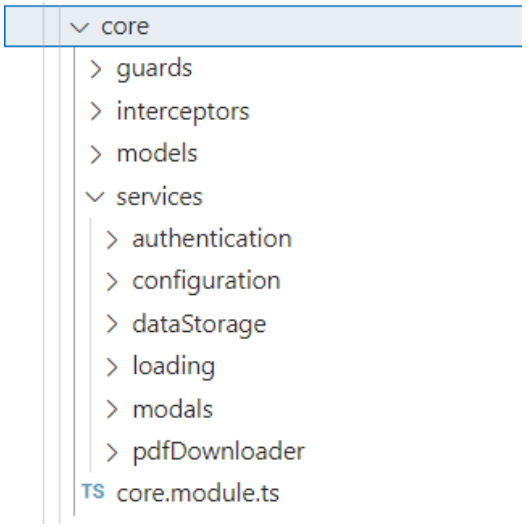


الشكل 27 : الوحدات في تطبيق الواجهة الأمامية

نلاحظ أنه لدينا وحدتين هما يشكلان مدخل العمل أي هما اللتان تحملان أولا ومن ثم حسب الطلب يتم تحميل الوحدات الأخرى وهما وحدة النواة (Core Module) و الوحدة المشتركة (Shared Module) .

تحوي الوحدة المشتركة على المكونات المشتركة بين الوحدات الأخرى و قالب الصفحة أي تحوي فقط مكونات للعرض.

تحوي النواة على مجلد الخدمات الاساسية في التطبيق من تخزين للبيانات ضمن المتصفح والتحقق من المستخدم والخدمات الرئيسة المشتركة بين الوحدات. وتحوي على مجلد الحراس الذي يحوي على حارس Guard يمنع المستخدم غير المسجل من تنسيط مكون يحتاج إلى أن يكون المستخجم مسجلا وآخر يسمح فقط للمستخدم الذي يملك الصلاحيات المناسبة من تنشيط المكون المطلوب. كما تحوي على معترضات لإضافة مفردة JWT للطلبات المرسلة.



الشكل 28 : بنية وحدة النواة **Core Module**

وبالنسبة لبقية الوحدات فهي كما ذكرنا تتكون من المجلدات الآتية :

أ. المكونات Components

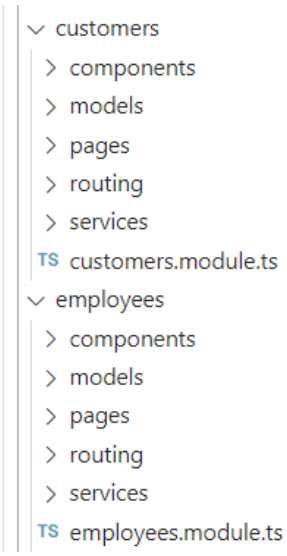
ب. النماذج Models : يحوي طلبات الإرسال و الردود

ت. الصفحات Pages : يحوي الصفحات المستقلة

ث. التوجيه Routing

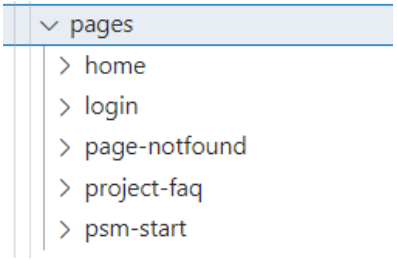
ج. الخدمات Services : يتحوي على خدمات تحقق الاتصال مع تطبيق المخدم وتجلب البيانات.

ويوضح الشكل أدناه هذه البينة.



الشكل 29 : البنية الضمنية في الوحدة

كما يوجد لدينا مجموعة من الصفحات الثابتة مثل صفحة ملامح المشروع والمساعدة وهي كما هو موضح في الشكل أدناه.



الشكل 30 : الصفحات الساكنة في التطبيق

الفصل الحادي عشر

الواجهات والاختبارات

نعرض في هذا الفصل الوجهات والتقارير التي يعرضها النظام كما نبين بعض الاختبارات التي قمنا بها.

1.6 مقدمة

2.6 اختبار البنية المعمارية

3.6 التقارير التي أنجزناها

أ. بطاقة تعريف مشروع

مشروع إدارة حالة المشاريع

بناء موقع ويب لأتمتة اجراءات إدارة مشاريع المعهد العالي

تاريخ البدء Aug 19, 2024

تاريخ الانتهاء المتوقع Sep 19, 2024

ذاتية المشروع

رمز المشروع: QWER

كتاب الطرح : 12 / Aug 19, 2024

كتاب الموافقة : 112 / Aug 19, 2024

التمويل : غير ممول / لايوجد

المعلومات التنفيذية

الجهة المنفذة: قسم النظم الإلكترونية

رئيس فريق العمل: حسن خضور

مدير المشروع : حسن خضور

ينفذ لصالح : شؤون الطلاب

معلومات حالة المشروع

المرحلة التطويرير : يخطط

نوع المشروع : مشروع تطويري

طبيعة المشروع : تطويري

مراحل المشروع

#	اسم المرحلة	الوصف	نسبة الانجاز	النسبة التقديرية	تاريخ البدء	مدة التنفيذ
1	مرحلة التطوير	تطوير النظام المطلوب	20	21	Aug 26, 2024	5
2	مرحلة الاختبار	يتك في هذه المرحلة اختبار النظام	0	19	Sep 7, 2024	10
3	مرحلة التصميم	يتم في هذه المرحلة تصميم النظام المطلوب	10	50	Aug 31, 2024	10
4	مرحلة التوثيق	يتم في هذه المرحلة توثيق النظام	10	10	Sep 30, 2024	2

المشاركين في المشروع

#	الرقم الذاتي	اسم المشارك	صفة العمل	الهيئة	نسبة التفرغ	المساهمة
1	1	حسن خضور	مبرمج	فني	15	مصمم واجهات

خطة الانفاق

#	نوع الكلفة	البيان	الشراء المحلي	الشراء الخارجي	نوع القطع	تاريخ الانفاق المتوقع
1	مواد أولية	شراء مواد أولية للمشروع	120123	12233	USA	Aug 29, 2024
2	شراء مخدمات	شراء مخدمات لتشغيل النظام المطلوب	100000	10000	USA	Sep 6, 2024
3	أجور عمال	دفع نفقات العمال	100000	10000	EUR	Oct 10, 2024

مدير المشروع	رئيس فريق العمل	الجهة الطارحة	الجهة المنفذة
حسن خضور	حسن خضور	شؤون الطلاب	قسم النظم الإلكترونية

ب. تقرير متابعة مراحل

بطاقة متابعة مراحل

طباعة | تقرير متابعة العمالين | تقرير خطة زمنية

مشروع إدارة حالة المشاريع

بناء موقع وب لاتمتة اجراءات إدارة مشاريع المعهد العالي

تاريخ البدء Aug 19, 2024

تاريخ الانتهاء المتوقع Sep 19, 2024

ذاتية المشروع

رمز المشروع: QWER

كتاب الطرح : 12 / Aug 19, 2024

كتاب الموافقة : 112 / Aug 19, 2024

التمويل : غير ممول / لايوجد

المعلومات التنفيذية

الجهة المنفذة: قسم النظم الإلكترونية

رئيس فريق العمل: حسن خضور

مدير المشروع : حسن خضور

ينفذ لصالح : شؤون الطلاب

معلومات حالة المشروع

المرحلة التطويرير : يخطط

نوع المشروع : مشروع تطويري

طبيعة المشروع : تطويري

قائمة متابعة المراحل لعملية المتابعة التي حدثت بتاريخ Aug 29, 2024

الوضع الراهن : وفق الخطة

وضع عملية المتابعة : غير مكتملة

الملاحظات : يسير العمل في المشروع وفق الخطة

#	اسم المرحلة	النسبة التقديرية	نسبة التنفيذ السابقة	نسبة تنفيذ المتابعة	نسبة التنفيذ الكلية	طبيعة االتنفيذ
1	مرحلة الاختبار	19	0	10	10	وفق الخطة
2	مرحلة التصميم	50	10	15	25	وفق الخطة

مدير المشروع	رئيس فريق العمل	الجهة الطارحة	الجهة المنفذة
حسن خضور	حسن خضور	شؤون الطلاب	قسم النظم الإلكترونية

ت. تقرير انشغالية عاملين

بطاقة معلومات مشروع

طباعة | بطاقة التعريف | تقرير متابعة المراحل | تقرير انجاز | تقرير خطة زمنية

مشروع أتمتة أعمال شؤون الطلاب

بناء موقع وب لمتابعة اعمال شؤون الطلاب

تاريخ البدء Aug 17, 2024

تاريخ الانتهاء المتوقع Aug 17, 2024

ذاتية المشروع

رمز المشروع: DTDG

كتاب الطرح : 10 / Aug 17, 2024

كتاب الموافقة : 9455 / Aug 17, 2024

التمويل : ممول / المركز

المعلومات التنفيذية

الجهة المنفذة: قسم المعلوميات

رئيس فريق العمل: حسن خضور

مدير المشروع : احمد الحسن

ينفذ لصالح : شؤون الطلاب

معلومات حالة المشروع

المرحلة التطويرير : مكتمل

نوع المشروع : مشروع تطويري

طبيعة المشروع : تطويري

قائمة متابعة العاملين لعملية المتابعة التي حدثت بتاريخ Aug 30, 2024

الوضع الراهن : وفق الخطة

وضع عملية المتابعة : غير مكتملة

الملاحظات : يسير العمل في المشروع وفق الخطة

الرقم الذاتي	الاسم والشهرة	ساعات العمل المسندة	ساعات العمل المنفذة	نسبة المساهمة	العمل المسند	العمل المنجز	التاريخ المتوقع لانهاء الاعمال	الملاحظات
1	حسن خضور	7	8	50	كل الأعمال	كل الأعمال	Aug 31, 2025	يعمل وفق ماهو مسند إليه
3	احمد الحسن	9	8	50	كل الأعمال	لم يقم ببعض الاعمال المسندة	Aug 31, 2025	يعمل وفق ماهو مسند إليه

مدير المشروع
احمد الحسن

رئيس فريق العمل
حسن خضور

الجهة الطارحة
شؤون الطلاب

الجهة المنفذة
قسم المعلوميات

ث. بطاقة تبدلات موظف

ج. بطاقة تاريخ متابعة مشروع

بطاقة متابعات مشروع

طباعة | تقرير خطة زمنية

مشروع إدارة حالة المشاريع

بناء موقع ويب لاتمتة اجراءات إدارة مشاريع المعهد العالي

تاريخ البدء Aug 19, 2024

تاريخ الانتهاء المتوقع Sep 19, 2024

ذاتية المشروع

رمز المشروع: QWER

كتاب الطرح : 12 / Aug 19, 2024

كتاب الموافقة : 112 / Aug 19, 2024

التمويل : غير ممول / لايوجد

المعلومات التنفيذية

الجهة المنفذة: قسم النظم الإلكترونية

رئيس فريق العمل: حسن خضور

مدير المشروع : حسن خضور

ينفذ لصالح : شؤون الطلاب

معلومات حالة المشروع

المرحلة التطويرير : يخطط

نوع المشروع : مشروع تطويري

طبيعة المشروع : تطويري

قائمة عمليات متابعة المشروع

م	الوضع الراهن	تاريخ المتابعة	الملاحظات	حالة المتابعة
1	وفق الخطة	Aug 24, 2024	يسير العمل في المشروع وفق الخطة	مكتملة
2	وفق الخطة	Aug 29, 2024	يسير العمل في المشروع وفق الخطة	مكتملة
3	وفق الخطة	Aug 29, 2024	يسير العمل في المشروع وفق الخطة	غير مكتملة

مدير المشروع	رئيس فريق العمل	الجهة الطارحة	الجهة المنفذة
حسن خضور	حسن خضور	شؤون الطلاب	قسم النظم الإلكترونية

ح. تقرير خطة انفاق سنوية

خ. تقرير خطة زمنية

الخاتمة

يجب أن يختتم التقرير بخاتمة

المراجع

[1] T. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley, 1991

[2]