

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

مشروع سنة الرابعة

تتبع حالة مشاريع المعهد العالي

تقديم

حسن بهجت خضور

إشراف

د. مصطفى دقاق م. محمود الياس

8/10/201

الإهداء.

الخلاصة

إنَّ بناء نظام يتيح تعريف المشاريع إلكترونياً ومتابعة سيرها ضمن المعهد العالي أصبح حاجة ملحة، انطلاقاً من مبداء مواكبة التحول الرقمي والاستغناء عن العمل الورقي، ولما تقدمه هذه الأنظمة من فوائد تنعكس على استراتيجيات اتخاذ القرارات. وانطلاقاً من ذلك قمنا في هذا العمل ببناء موقع لتتبع حالة المشاريع في المعهد العالي، يمكن من خلاله إدارة إجرائية تخطيط المشروع ومتابعته. وتم انجاز هذا الموقع باستخدام البنية المعمارية النظيفة وبعض مبادئ التصميم المقاد بالمجال وعدد من الأنماط التصميمية.

Abstract

Building a system that allows projects to be defined and tracked within the **HIAST** has become an urgent necessity, This is driven by the principle of embracing digital transformation and eliminating paper-based processes, as well as the benefits such systems offer for decision-making strategies. Therefore, in this work, we built a website to track the status of the projects at **HIAST**. This site facilitates project planning and monitoring through project management procedural. The website was developed using clean architecture, some domain-driven design principles, and several design patterns.

المحتويات

V	قائمة الأشكال
1	الفصل الأول
1	التعريف بالمشروع
2	1.1- مقدمة
2	2.1- الهدف من المشروع
2	3.1- نطاق المشروع
2	4.1- المتطلبات الوظيفية
5	5.1- المتطلبات غير الوظيفية
6	الفصل الثاني
6	الدراسة المرجعية
7	1.2- مقدمة
7	2.2 الحاجة إلى النظام
7	3.2 النظام الحالي ومشاكله / إن وجد
7	4.2 القيمة المضافة التي سيقدمها
7	5.2 بروتوكول الوصول إلى الدليل النشط LDAP
8	الفصل الثالث
8	دراسة آلية إدارة المشاريع
9	1.3- مقدمة
9	2.3- دورة حياة المشروع (النشاط)
11	3.3 آلية إدارة المشروع المتبعة في المعهد
13	الفصل الرابع
13	خطة إدارة المشروع
14	1.4- مقدمة
14	2.4- الإجراءات المتبعة في تطوير النظام
15	3.4- الخطة الزمنية

16	4.4- مخطط غانت Gantt Chart
17	الفصل الخامس
17	الدراسة التحليلية
18	1.5- مقدمة
18	2.5 مخطط أطوار المشروع
19	3.5- مدخلات ومخرجات النظام
20	4.5 مخططات حالات الاستخدام
39	6.5 مخططات التتالي لبعض حالات الاستخدام
41	7.5 عقود عمليات حالة الاستخدام متابعة مشروع
43	8.5 مخطط الصفوف
44	الفصل السادس
44	الدراسة التصميمية
45	1.6- مقدمة
45	2.6 البنية المعمارية النظيفة - Clean Architecture
48	6.6 نمط التصميم المستودع Repository Pattern
49	7.6 نمط التصميم Result Pattern
50	الفصل السابع
50	تصميم النظام
51	1.7 مبادئ التصميم
52	2.7 مخطط العلاقات بين الكائنات Entity Relationship Diagram
54	الفصل الثامن
54	الأدوات وأطر العمل المستخدمة
55	1.8 إطار العمل Net Core
55	2.8 إطار العمل Angular
55	3.8 قاعدة المعطيات SQL Server
55	4.8 نمط الخيارات Option Pattern
55	5.8 JSON Web Token
56	5.8 المكتبة MediatR

57	الفصل التاسع
57	تنجيز النظام
58	1.9 منهجية تصميم النظام
58	2.9 البنية المعمارية لطرف المخدم
82	3.9 بنية الواجهة الامامية Front-end
85	الفصل العاشر
85	الواجهات والاختبارات
86	1.10 مقدمة
86	2.10 اختبار البنية المعمارية
89	3.10 التقارير التي أنجزناها
96	4.10 بعض الواجهات في الموقع

قائمة الأشكال

- الشكل 1: عمليات المشروع عبر دورة حياته 10
- الشكل 2: مخطط غانت لأنشطة خطة التنفيذ 16
- الشكل 3: مخطط أطوار المشروع 18
- الشكل 4: مخطط حالات الاستخدام لرئيس فريق العمل 20
- الشكل 5: مخطط حالات الاستخدام لمدير المشروع 26
- الشكل 6: مخطط حالات الاستخدام للوكيل العلمي 30
- الشكل 7: مخطط حالات الاستخدام للموظف 33
- الشكل 8: مخطط حالات الاستخدام للمستخدم 35
- الشكل 9: مخطط حالات الاستخدام لرئيس الفعالية 36
- الشكل 10: العلاقات بين الفاعلين 38
- الشكل 11: مخطط التالي لحالة الاستخدام متابعة مشروع 39
- الشكل 12: مخطط التالي لحالة الاستخدام تخطيط مشروع 40
- الشكل 13: مخطط الصفوف 43
- الشكل 14: مكونات البنية المعمارية النظيفة 45
- الشكل 15: مثال توضيحي عن النمط الوسيط 47
- الشكل 16: مثال توضيحي عن نمط الحالة 47
- الشكل 17: مخطط صفوف تجريدي يصف نمط الحالة 48
- الشكل 18: مخطط صفوف تجريدي للنمط المستودع 49
- الشكل 19: بنية عامة للتصميم المقترح 51
- الشكل 20: مخطط العلاقات بين الكيانات ERD 52
- الشكل 21: البنية المعمارية لطرف الواجهة الخلفية 59
- الشكل 22: هيكلية الحل المنجز 60
- الشكل 23: بنية طبقة النواة المشتركة 61
- الشكل 24: بنية طبقة التطبيق 62
- الشكل 25: البنية الضمنية في طبقة التطبيق 63
- الشكل 26: بنية مجال المشاريع 64
- الشكل 27: رسم توضيحي لمجال المشروع 64

الشكل 28: مثال على استخدام الصف الذي يحقق نمط الباني	65
الشكل 29: مثال على آلية تغليف الحالة لمؤثر الانتقال	66
الشكل 30: بنية طبقة التطبيق	67
الشكل 31: البنية الضمنية في كل مجال	68
الشكل 32: العقود ضمن طبقة التطبيق	68
الشكل 33: مثال على تنجيز حالة استخدام بالنمط CQRS	69
الشكل 34: Event Sourcing Example	70
الشكل 35: بعض معالجات الأحداث في مجال المشاريع	71
الشكل 36: مثال عن استخدام نمط النتيجة	72
الشكل 37: بنية مشروع التعاقدات	74
الشكل 38: بنية مشروع العرض	74
الشكل 39: مثال على حقن الاعتماديات واختيار النبية التحتية	75
الشكل 40: بنية مشروع مداومة البيانات	76
الشكل 41: مخطط العلاقات ضمن قاعدة المعطيات	77
الشكل 42: العلاقات بين المستخدمين والأدوار والصلاحيات	78
الشكل 43: بنية مشروع الخدمات	80
الشكل 44: الوحدات في تطبيق الواجهة الأمامية	82
الشكل 45: بنية وحدة النواة Core Module	83
الشكل 46: البنية الضمنية في الوحدة	84
الشكل 47: الصفحات الساكنة في التطبيق	84
الشكل 48: اختبار الاعتماديات	87
الشكل 49: اختبارات قواعد التسمية في طبقة التطبيق	87
الشكل 50: اختبار قواعد التسمية في طبقة المجال	88

مقدمة عامة

تعتبر إدارة المشاريع وتتبعها من العناصر الحيوية لدعم عملية اتخاذ القرارات في المؤسسات. تُسهم هذه العملية في تحديد أهداف المشروع بوضوح، وتوزيع الموارد بكفاءة، وتحديد المسؤوليات، مما يسهل مراقبة الأداء وتحليل البيانات المتعلقة بالتقدم المحرز. يساعد تتبع المشاريع في الكشف المبكر عن المشكلات المحتملة واتخاذ الإجراءات التصحيحية الفعّالة، مما يقلل من المخاطر ويزيد من فرص النجاح. إضافةً إلى ذلك، يوفر تتبع المشاريع رؤية مدعومة بالبيانات تسهم في اتخاذ قرارات استراتيجية مبنية على أسس قوية، وتحقيق الأهداف بفعالية وكفاءة.

الفصل الأول

التعريف بالمشروع

نبيّن في هذا الفصل هدف المشروع ونطاقه كما نورد المتطلبات الوظيفية وغير الوظيفية.

1.1- مقدمة

يهدف هذا الفصل إلى تقديم نظرة شاملة عن المشروع، بما في ذلك أهدافه، نطاقه، والمتطلبات الوظيفية وغير الوظيفية التي يجب تحقيقها لضمان نجاحه وفعاليته. سيتم استعراض جميع الجوانب المتعلقة بالمشروع، بدءًا من تعريفه وأهدافه وصولاً إلى القيود والمعايير التي يجب مراعاتها عند تطوير وتشغيل النظام.

2.1- الهدف من المشروع

يهدف هذا المشروع إلى بناء نظام معلوماتي نتبّع من خلاله حالات المشاريع ضمن المعهد العالي، حيث يقدم هذا النظام إمكانية طرح مشروع وإرفاق المستندات الخاصة بالمشروع وإسناده للفعالية المنقّدة ومتابعته. كما يمكن مديري المشاريع من تعريف مشروع وإضافة خطة إنفاقاً لإنفاق عليه ومعلومات التمويل والتصنيف الخاصة بالمشروع، ويمكن رؤساء فرق العمل من إضافة مراحل المشروع، كما يمكنهم من إدارة عمليات المتابعة للمشاريع القائمين عليها من خلال متابعة نسب التنفيذ والتأخير ومدى انشغالية العاملين ومساهماتهم. ويؤمن هذا النظام إمكانية إصدار التقارير والاستثمارات 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 بها.

ب. أن يسمح لرئيس فريق العمل بما يلي:

1. القيام بعمليات المتابعة للمشاريع التي يرأسها، أي إدخال نسب الإنجاز 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).

ت. أن يسمح لمدير المشروع بما يلي:

1. إدارة خطة الإنفاق لمشروع.
2. إضافة المعلومات المالية للمشروع.
3. إضافة تصنيف المشروع.
4. إصدار تقرير عن حالة المشروع.
5. إصدار تقرير إنجاز 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.

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 حالاتها.

5.1- المتطلبات غير الوظيفية

نورد هنا المتطلبات غير الوظيفية أي القيود على النظام 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 بالعمليات وفقاً لما هم مخولين به من صلاحيات فقط.

متطلبات الأداء

1. يجب أن يستجيب النظام لطلبات المستخدم في غضون زمن محدد لا يتجاوز 2 ثانية لمعظم العمليات.

متطلبات خاصة

1. أن تكون الواجهات باللغة العربية.

متطلبات التشغيل

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

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 4.2 القيمة المضافة التي سيقدمها

5.2 بروتوكول الوصول إلى الدليل النشط LDAP

بروتوكول الوصول إلى الدليل النشط (Lightweight Directory Access Protocol) هو بروتوكول مستخدم بنطاق واسع بحف الوصول إلى خدمات الدليل وإدارتها عبر الشبكة الحاسوبية. وخدمات الدليل هي قواعد بيانات مصممة لتخزين وإدارة المعلومات المتعلقة بالمستخدمين والأجهزة والتطبيقات والموارد الأخرى داخل شبكة معينة، ويتم تنظيم هذه المعلومات بشكل هرمي لتسهيل عملية البحث والإدارة. يُستخدم LDAP على نطاق واسع في العديد من التطبيقات، مثل أنظمة المصادقة (Authentication Systems)، وأنظمة التحكم في الوصول (Access Control Systems)، وأنظمة البريد الإلكتروني، ويُعد جزءاً أساسياً من البنية التحتية لتكنولوجيا المعلومات في المؤسسات الكبيرة.

تم تطوير LDAP كإصدار خفيف من بروتوكول أقدم يسمى (Directory Access Protocol) DAP. يتميز LDAP بكونه أخف وأبسط من DAP، حيث يعتمد على نموذج العميل/الخادم (Client/Server Model)، حيث يقوم العميل (Client) بإرسال طلبات استعلام أو تعديل إلى الخادم (Server) الذي يستضيف قاعدة بيانات الدليل. يتم تمثيل البيانات في LDAP باستخدام ما يُعرف بـ **شجرة معلومات الدليل** (Directory Information Tree - DIT)، التي تُبنى بشكل هرمي ابتداءً من جذر الدليل (Root) وتمتد لتشمل الكيانات الأخرى مثل النطاقات (Domains)، والوحدات التنظيمية (Organizational Units - OUs)، والمستخدمين (Users)، والمجموعات (Groups).

تتألف بنية LDAP من **عناصر مميزة** تعرف بالتمييز المميز (Distinguished Names - DN) التي تحدد موقع كل كيان داخل الشجرة الهرمية، ويدعم LDAP العديد من العمليات الأساسية مثل البحث عن البيانات، والإضافة، والتعديل، والحذف، مما يجعله أداة قوية لإدارة موارد الشبكة بشكل مركزي.

يتمتع LDAP بمزايا عديدة تشمل الأداء العالي، وإمكانية التوسع، والمرونة في الوصول إلى البيانات وإدارتها، لكنه يتطلب أيضاً معرفة متخصصة لضبطه وصيانته بشكل صحيح. في البيئات المعقد.

الفصل الثالث

دراسة آلية إدارة المشاريع

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

1.3- مقدمة

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

تعد إجراءات إدارة المشاريع جزءاً حيوياً 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. وأخيراً، تأتي مرحلة الإغلاق، حيث يتم تقييم النتائج، وتوثيق الإنجازات، والتأكد من تحقيق أهداف المشروع بشكل كامل.

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

2.3- 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، وأخيراً الإغلاق.

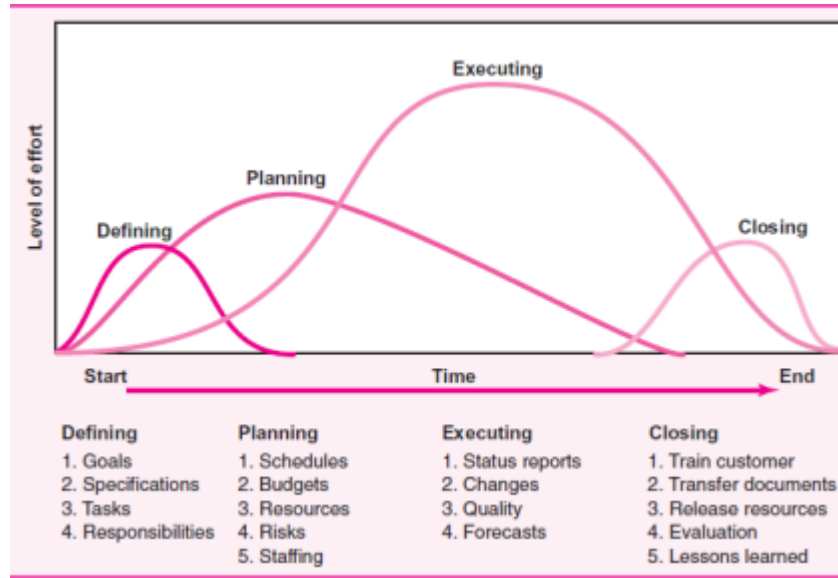
كل مرحلة من هذه المراحل تلعب دوراً حاسماً في تحديد نجاح المشروع، حيث يتم في كل منها اتخاذ قرارات تساهم في تحقيق أهداف المشروع بأعلى قدر ممكن من الكفاءة.

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 فتعتبر 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 خطة 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. وينتج عن هذه المرحلة تقارير تساهم في قياس الأداء والتقدم التي بدورها تستخدم لإطلاع الجهات الإدارية عليها لاتخاذ القرارات المناسبة.

وفي النهاية تأتي مرحلة الإغلاق أو كما يسميها البعض لإنجاز. 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

3.3 آلية إدارة المشروع المتبعة في المعهد

من خلال الاجتماعات مع المعنيين والاطلاع على آلية سير العمل، تبين لنا أن عملية إدارة المشاريع حالياً تتم وفق الإجراءات الآتية

- إجراءات الطرح والتعريف:

يقوم الزبون (الجهة الطارحة) بتقديم المشروع الذي يرغب 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 أيضاً مهمة رئاسة فريق العمل للموظف المناسب (رئيس فريق العمل).

- إجراءات 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 والجدوى الاقتصادية منه والخطة المالية وعدد العاملين اللازم. يرفع المشروع بعد هذا التوصيف وإرفاقه بالملفات التي يراها كل من مدير المشروع ورئيس فريق العمل ضرورية إلى لجنة الشؤون العلمية ومن ثم مجلس المعهد (الإدارة المعنية بالموافقة) للموافقة على المشروع أو رفضه. في حال لم تر الإدارة أن المشروع يلبي متطلباتها وأنه غير مجد وقررت رفضه يعلم المعنيين أن المشروع قد ألغي. وفي حال الموافقة على المشروع وإعطائه 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 وتوزيعها على الربيعات. ومن بعد ذلك ترفع هذه المعلومات إلى الجهة المعنية ويبدأ من بعدها العمل بالمشروع.

- إجرائية المتابعة:

تنقسم السنة إلى أربع ربعيات، وفي كل رعية يجب على رئيس فريق العمل القيام بعملية المتابعة حيث تنقسم هذه العملية إلى جزئين 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 رعية يجب أن تُعرض تقارير المتابعة على لجنة الشؤون العلمية وعلى مجلس المعهد (الإدارة المعنية) للاطلاع عليها.

والاطلاع على المشاريع المتأخرة لتذليل الصعوبات واتخاذ الإجراءات المناسبة.

- إجرائية التسليم:

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

الفصل الرابع

خطة إدارة المشروع

نبيّن في هذا الفصل خطة إدارة المشروع والخطة الزمنية لتطوير النظام كما نبيّن أيضا الإجراءات 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 النظام

اتبعنا في تطوير النظام إجرائية التطوير الشلالي (Waterfall Process) والتي تتسم بكونها منهجية خطية متسلسلة، حيث يتم الانتقال من مرحلة إلى أخرى بعد إتمام المرحلة السابقة.

لتحقيق 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 لأخذ الموافقة عليها.

3. مرحلة الدراسة المرجعية: في هذه المرحلة يتم تحليل الآلية المتبعة في النظام القديم إن وجد ودراسة آلية العمل لفهمها.

4. مرحلة الدراسة التحليلية:

5. مرحلة التطوير:

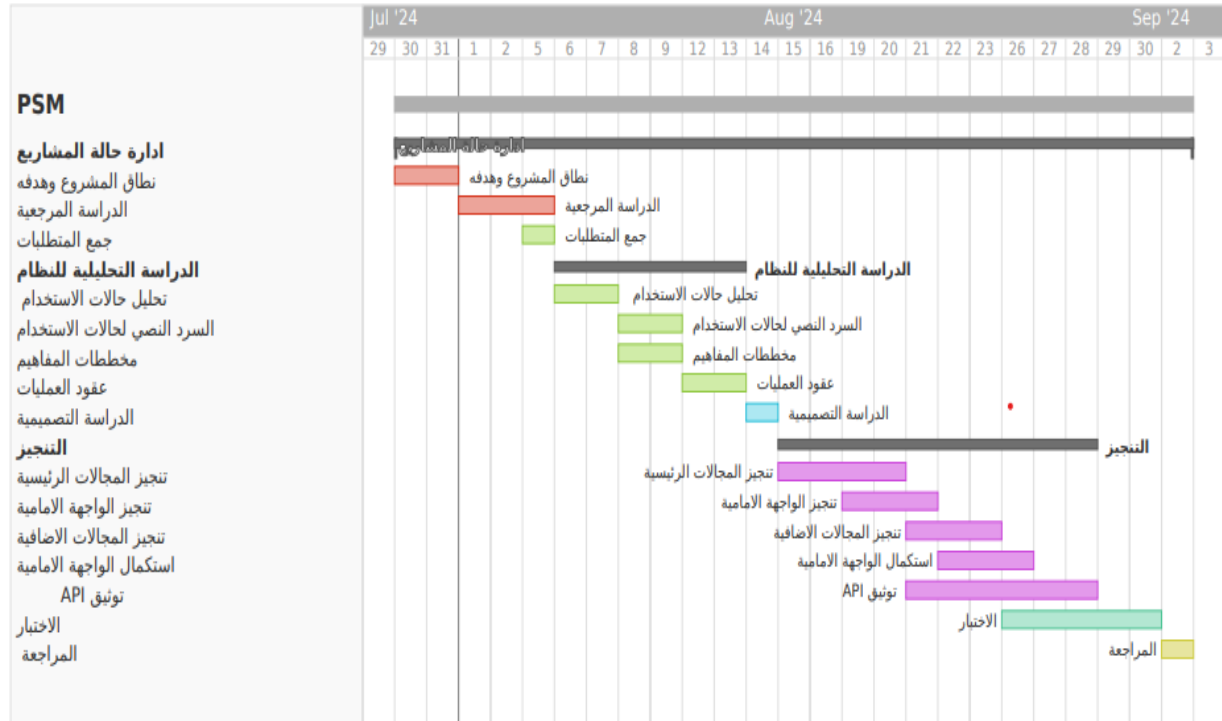
6. مرحلة الاختبار:

7. مرحلة التسليم:

3.4- الخطة الزمنية

مشروع إدارة حالة مشاريع المعهد العالي		
المرحلة	تاريخ البدء	المدة (بالأيام)
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 2: مخطط غانت لأنشطة خطة التنفيذ

الفصل الخامس

الدراسة التحليلية

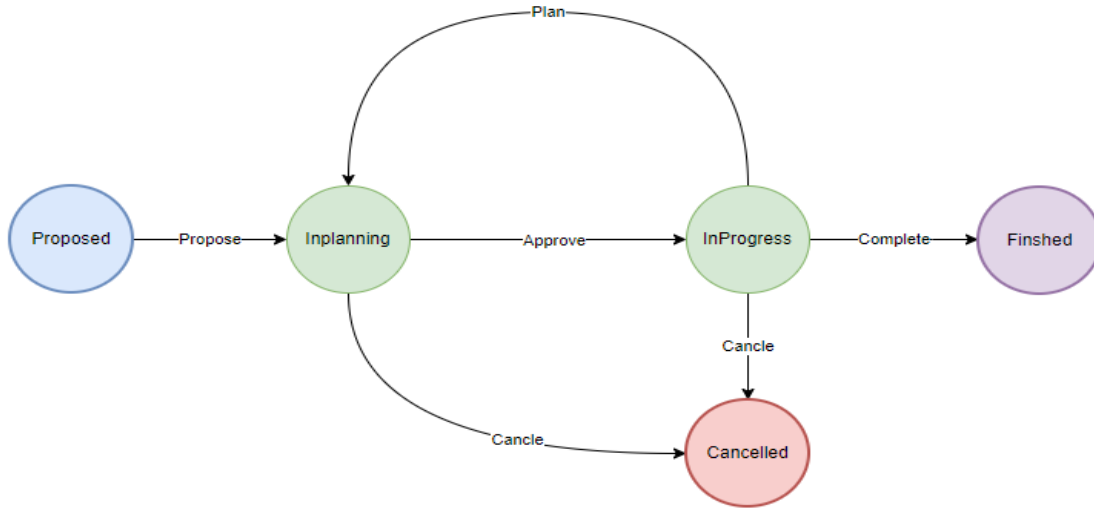
يقدم هذا الفصل تحليلاً للمتطلبات التي أوردناها في الفصل الأول.

1.5- مقدمة

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

2.5 مخطط أطوار المشروع

يوضح الشكل المرفق أدناه مخططاً لأطوار حالة المشروع وتغيراتها بناءً على المؤثرات المطبقة على المشروع.



الشكل 3: مخطط أطوار المشروع.

3.5- مدخلات ومخرجات النظام

تبين لنا من خلال الاجتماعات مع المعنيين أنه يوجد حالياً عدد من الاستثمارات التي يحتاجونها ويتعاملون معها أثناء إدارة، طرح وتتبع المشاريع وهي على النحو الآتي:

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

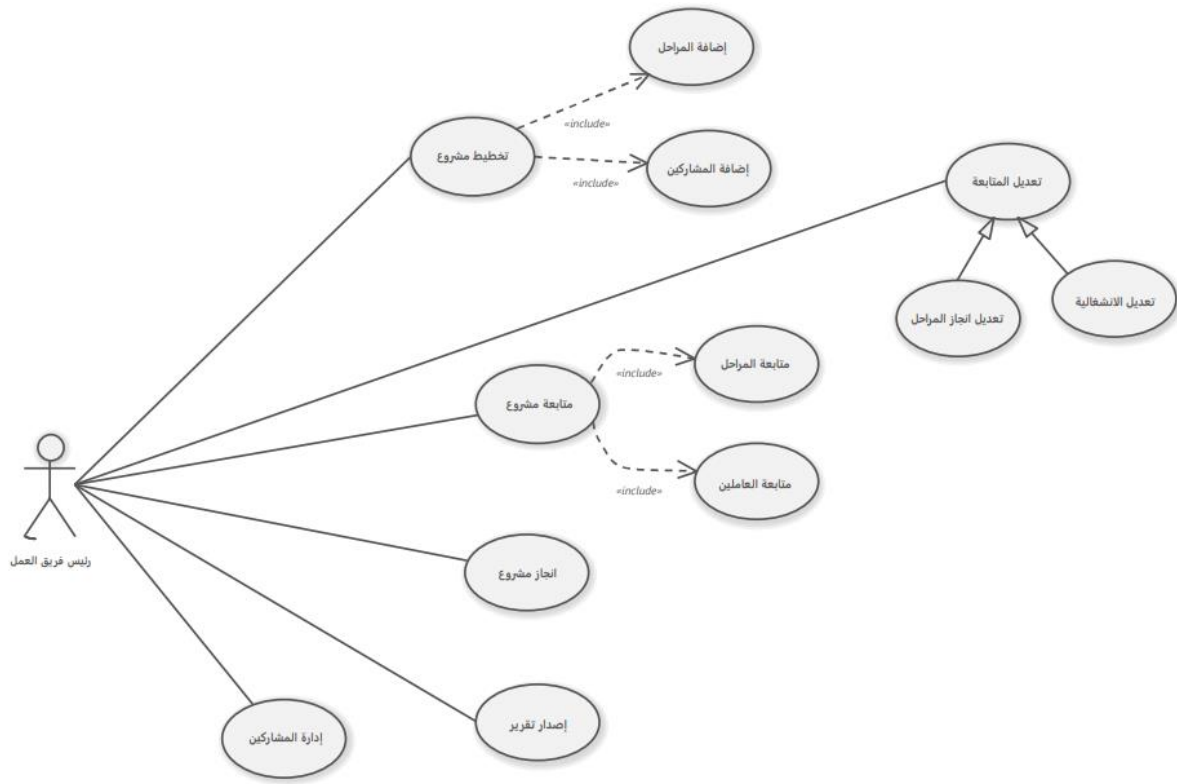
تبين لنا أن الاستثمارات من 1 وحتى 6 هي مدخلات للنظام يجب أن تدخل إلى النظام.
وأن الاستثمارات من 7 وحتى 10 هي مخرجات للنظام يجب أن تصدر عن النظام وتُعرض على المعنيين.

4.5 مخططات حالات الاستخدام

نبين هنا حالات الاستخدام في النظام لتحقيق المتطلبات المبينة في الفصل الأول والسرد النصي الذي يصف آلية تحقيقها، حيث نورد هذا السرد النصي بناءً على آلية سير العمل التي ذكرناها آنفاً.

1.4.5 حالات الاستخدام الخاصة برئيس فريق العمل

حالات الاستخدام



الشكل 4: مخطط حالات الاستخدام لرئيس فريق العمل

الوصف السردى لحالات الاستخدام

أ- حالة الاستخدام متابعة مشروع

اسم حالة الاستخدام: متابعة مشروع	
الفاعلون الأوليون	رئيس فريق العمل
الشروط المسبقة	<ul style="list-style-type: none"> المشروع معروف ومسجل لدى النظام. المشروع في طور التنفيذ. رئيس فريق العمل مسجل في النظام.
الشروط اللاحقة	<ul style="list-style-type: none"> اكتملت عملية المتابعة وسجلت لدى النظام. تقرير المتابعة أرسل للوكيل العلمي. انشغالية العاملين أسندت وتقرير الانشغالية تم رفعه.
الوصف	يقوم رئيس فريق العمل بمتابعة المشروع حيث يسند نسب الانشغالية للعاملين، ويسند نسب إنجاز الإنجاز في المراحل.

سير الأحداث

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 الإنجاز.	3. يقوم رئيس فريق العمل من أجل كل مرحلة بإدخال معلومات المتابعة (نسبة الإنجاز، الملاحظات، وضع المرحلة).
	4. يتطلب 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 الانشغال 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 الانشغال 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 انشغال العامل لا تتوافق مع نسبة تفرغه 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

لا يوجد.

ب- حالة الاستخدام تخطيط مشروع

اسم حالة الاستخدام: تخطيط مشروع	
الفاعلون الأوليون	رئيس فريق العمل
الشروط المسبقة	<ul style="list-style-type: none"> المشروع معروف ومسجل لدى النظام. المشروع في طور التخطيط. رئيس فريق العمل مسجل في النظام.
الشروط اللاحقة	<ul style="list-style-type: none"> 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 الأحداث

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

لا يوجد.

ت- حالة الاستخدام إصدار تقرير

اسم حالة الاستخدام: إصدار تقرير	
الفاعلون الأوليون	رئيس فريق العمل، مدير المشروع
الشروط المسبقة	<ul style="list-style-type: none"> المشروع معروف ومسجل لدى النظام. طالب الخدمة (رئيس فريق العمل، مدير المشروع) مسجل في النظام.
الشروط اللاحقة	لا يوجد.
الوصف	يقوم رئيس فريق العمل بإصدار تقرير من التقارير المتوفرة عن المشروع.

سير الأحداث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	رئيس فريق العمل
	1. يقوم ببدء عملية إصدار 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

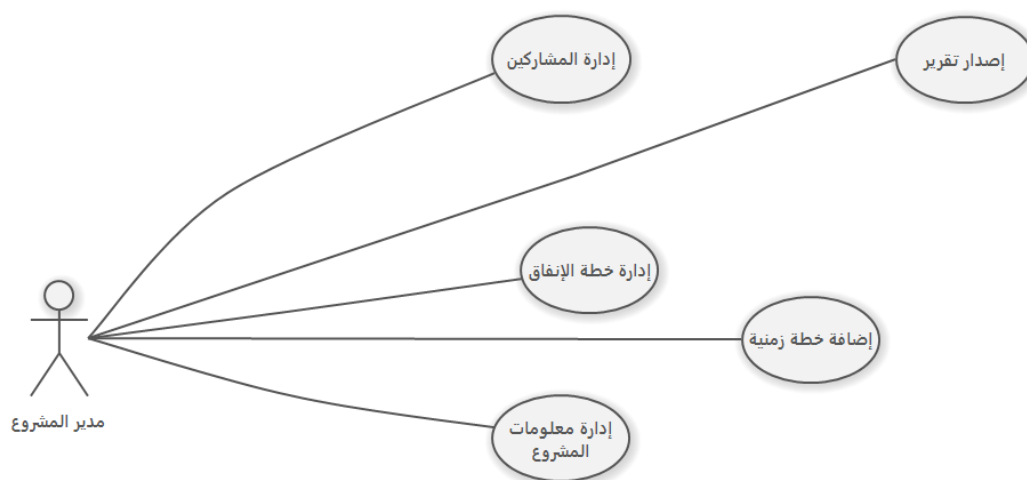
A1: في المرحلة رقم 7 إذا أراد إصدار تقرير آخر، فإنه يتابع من المرحلة رقم 3.

المسارات الخاطئة

لا يوجد.

2.4.5 حالات الاستخدام الخاصة بمدير المشروع

حالات الاستخدام



الشكل 5: مخطط حالات الاستخدام لمدير المشروع.

الوصف السردى لحالات الاستخدام

أ- حالة الاستخدام إضافة خطة الإنفاق

اسم حالة الاستخدام: إضافة خطة إنفاق	
الفاعلون الأوليون	مدير المشروع

الشروط المسبقة	<ul style="list-style-type: none"> المشروع معروف ومسجل لدى النظام. المشروع في طور التخطيط. مدير المشروع مسجل في النظام.
الشروط اللاحقة	أ. خطة الإنفاق أضيفت للمشروع.
الوصف	يقوم مدير المشروع بإضافة قائمة بالإنفاقات المتوقعة ضمن المشروع.

سير الأحداث

Main Success scenario – السيناريو الرئيسي الناجح

النظام	مدير المشروع
	1. يقوم ببدء عملية إضافة خطة إنفاق
2. يرسل معلومات المشروع (ميزانية المشروع وتوصيفه) (.	
	3. يدخل مدير المشروع قائمة لمعلومات الإنفاق (نوع الكلفة، البيان، الشراء المحلي والخارجي، التاريخ المتوقع لإنفاق)
	4. يطلب تأكيد عملية الإضافة
5. يسجل النظام معلومات عملية الإنفاق	
6. يعلم رئيس فريق العمل بنجاح العملية.	

المسارات البديلة

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

المسارات الخاطئة

لا يوجد.

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	معلومات المشروع سجلت لدى النظام.
الوصف	يقوم مدير المشروع بإضافة معلومات المشروع.

سير الأحداث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	مدير المشروع
	1. يقوم ببدء عملية إضافة معلومات للمشروع
2. يرسل النظام الاستثمارات الواجب ملؤها (استمارة التصنيف، استمارة النوع، استمارة الجدوى)	
	3. يقوم مدير المشروع بملء الاستثمارات
	4. يطلب تأكيد عملية الإضافة
5. يسجل النظام معلومات المشروع المضافة	
6. يعلمه بنجاح عملية الإضافة	
	7. يطلب إنهاء عملية الإضافة.

	8. يعلمه بنجاح العملية.
--	-------------------------

المسارات البديلة

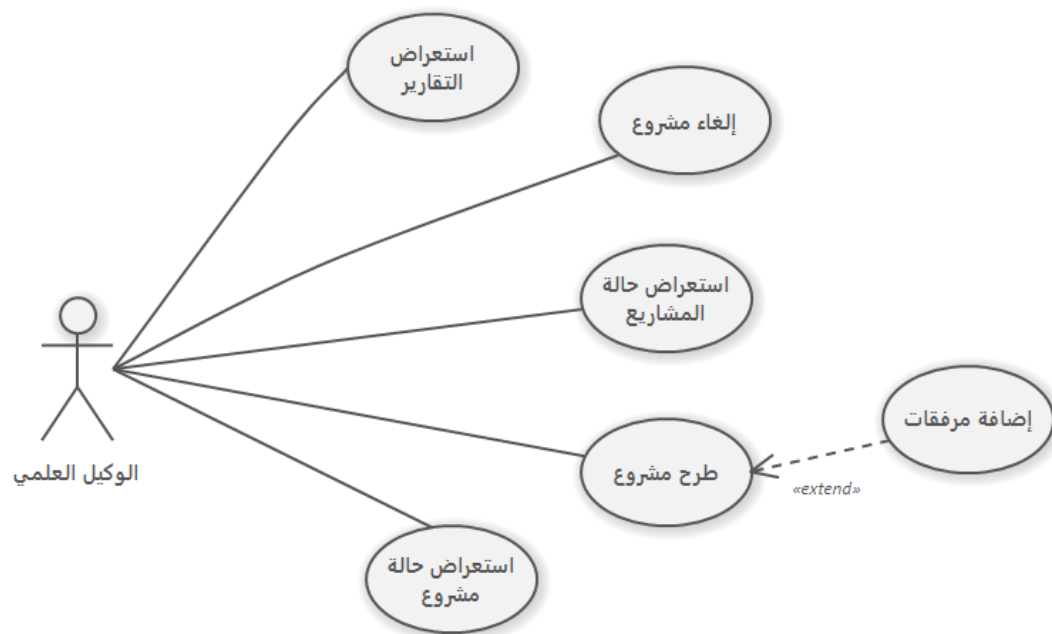
لا يوجد.

المسارات الخاطئة

لا يوجد.

3.4.5 حالات الاستخدام الخاصة بالوكيل العلمي

حالات الاستخدام



الشكل 6: 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 الأحداث

السيناريو الرئيسي الناجح – Main Success scenario

النظام	رئيس فريق العمل
	1. يقوم ببدء عملية طرح مشروع
2. يرسل النظام قائمة بأنواع المشاريع المتوفرة	
	3. يختار نوع المشروع.
4. يرسل قائمة بالموظفين المتاحين	
	5. يختار رئيس فريق العمل ومدير المشروع
	6. يحدد بيانات المشروع المطروح (الجهة الطارحة، معلومات التمويل، ...)
	7. يطلب إتمام عملية الطرح

8. يعلم رئيس فريق لعمل ومدير المشروع بإختيارهم	
9. يسجل معلومات المشروع.	
10. 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

لا يوجد.

ب- حالة الاستخدام استعراض التقارير

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

سير الأحداث

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

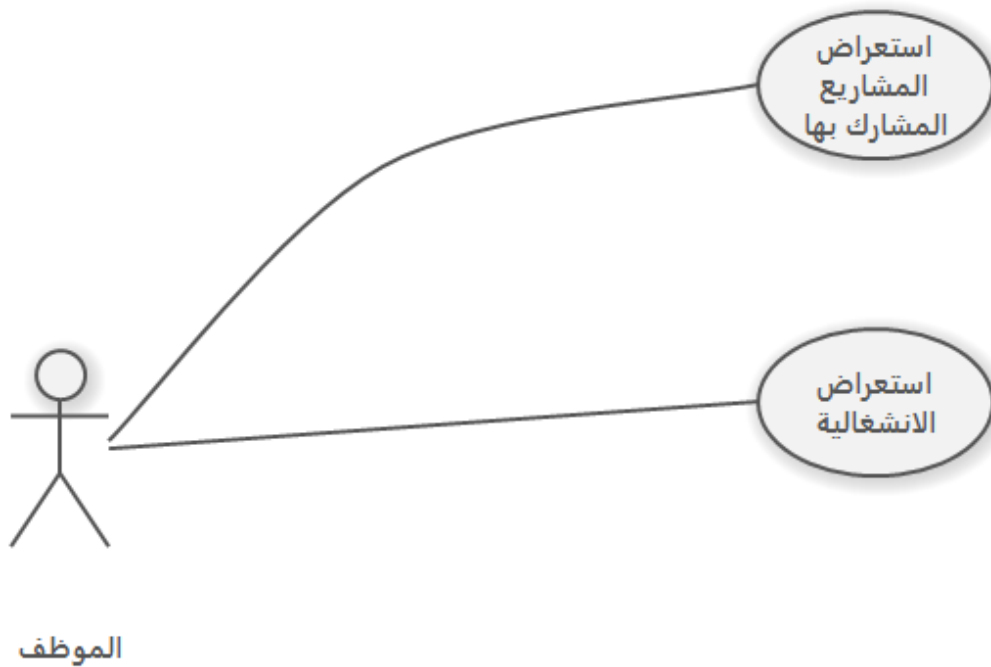
A1: في المرحلة رقم 7 إذا أراد الاطلاع على تقرير آخر، فإنه يتابع من المرحلة رقم 3.

المسارات الخاطئة

لا يوجد.

4.4.5 حالات الخاصة بالموظف

حالات الاستخدام



الشكل 7: مخطط حالات الاستخدام للموظف.

الوصف السردى لحالات الاستخدام

أ- حالة الاستخدام استعراض الانشغالية

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

سير الأحداث

السيناريو الرئيسي الناجح – Main Success scenario

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

المسارات البديلة

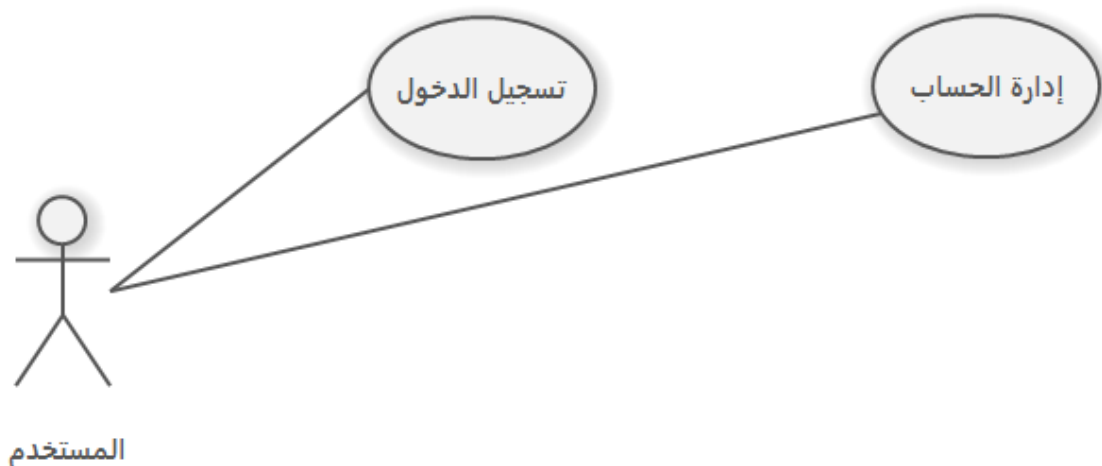
A1: في المرحلة رقم 4 إذا اختار مشروع ليس من المشاريع التي يشارك بها، يعلمه النظام بأن المشروع ليس من المشاريع المشارك بها ويتابع من المرحلة رقم 3.

المسارات الخاطئة

لا يوجد.

5.4.5 حالات الاستخدام الخاصة بالمستخدم

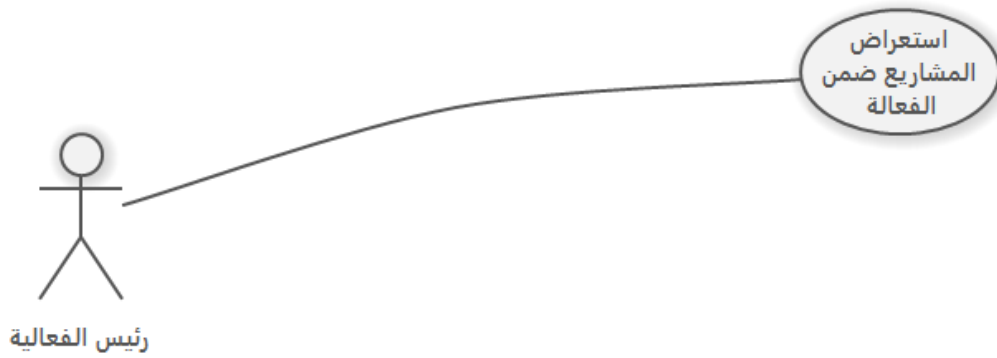
حالات الاستخدام



الشكل 8: مخطط حالات الاستخدام للمستخدم

6.4.5 حالات الاستخدام الخاصة برئيس الفعالية

حالات الاستخدام



الشكل 9: مخطط حالات الاستخدام لرئيس الفعالية

الوصف السردى 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	لا يوجد.
الوصف	يقوم رئيس الفعالية بإصدار تقرير بالمشاريع ضمن الفعالية التي يرأسها.

السيناريو الرئيسي الناجح – Main Success scenario

رئيس فريق العمل	النظام
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.

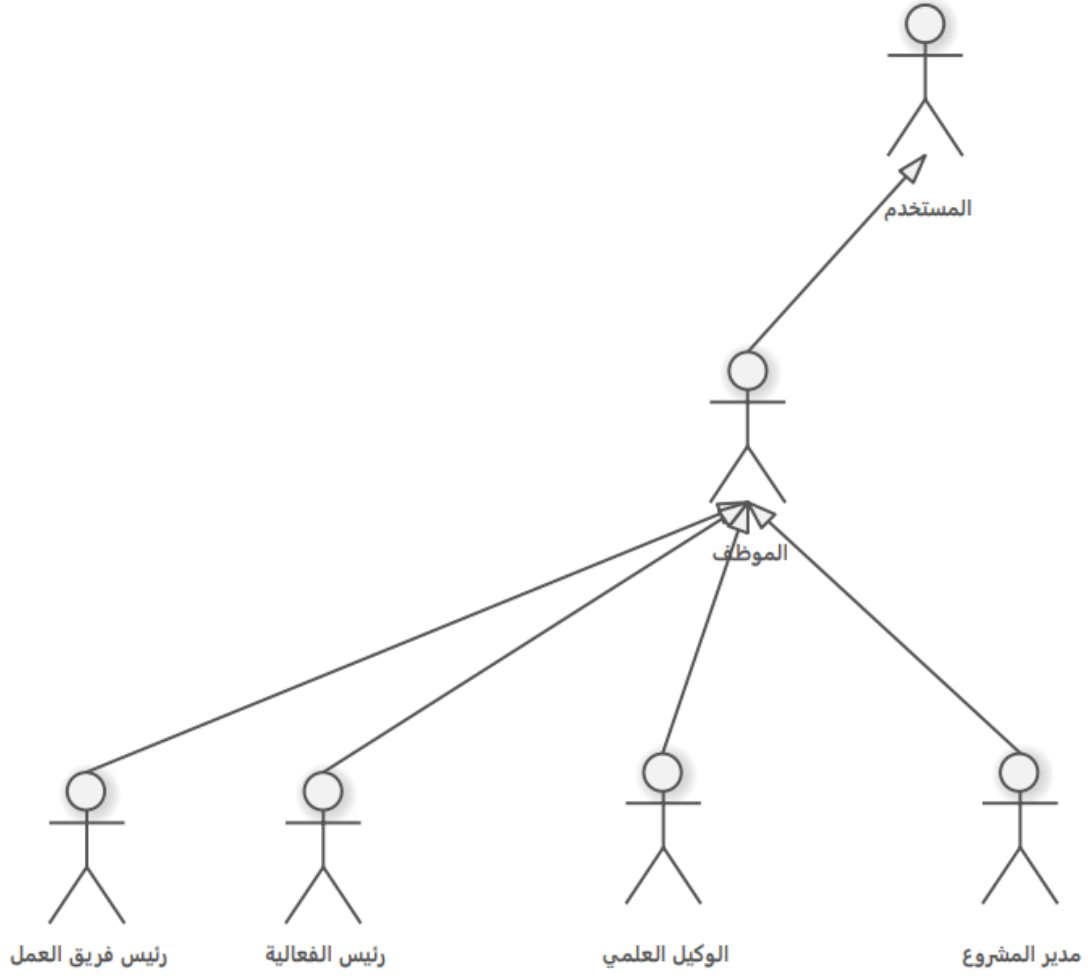
المسارات البديلة

لا يوجد.

المسارات الخاطئة

لا يوجد.

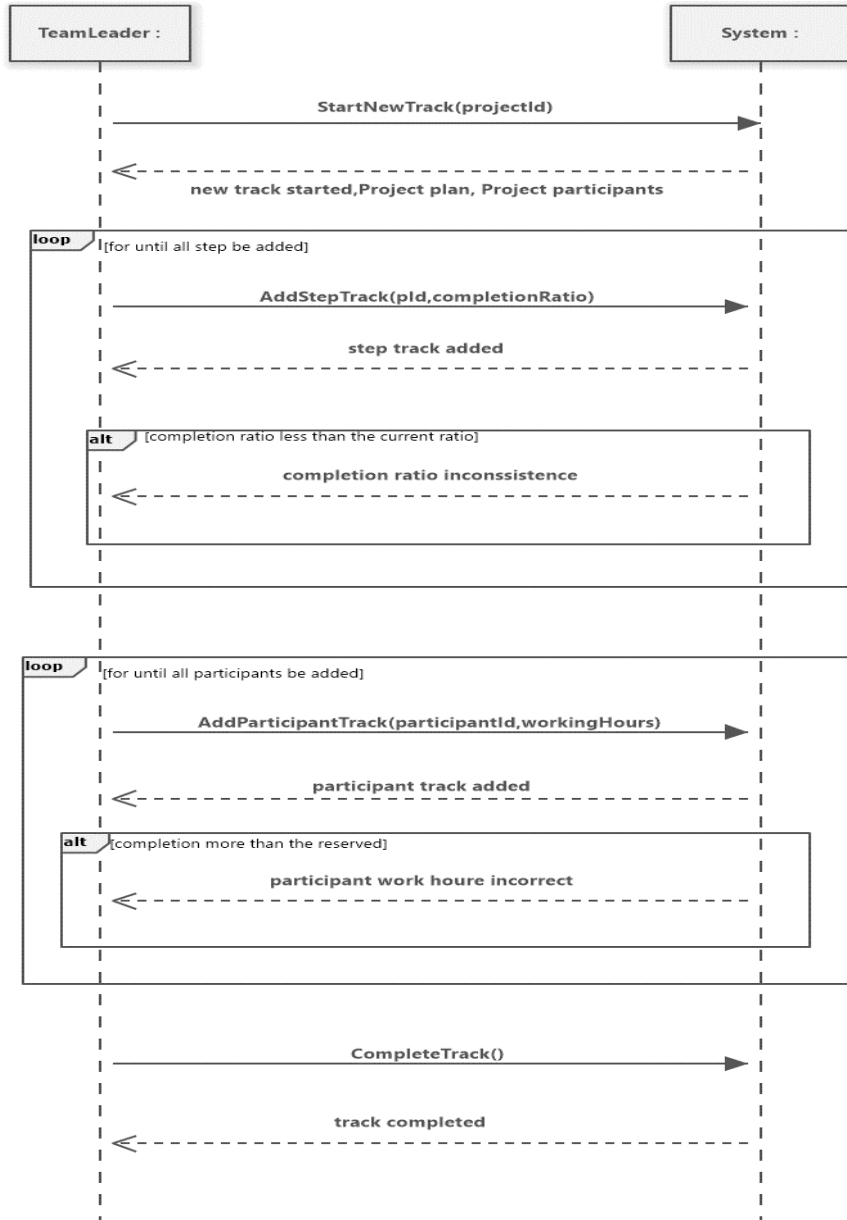
8.4.5 العلاقات بين الفاعلين



الشكل 10: العلاقات بين الفاعلين

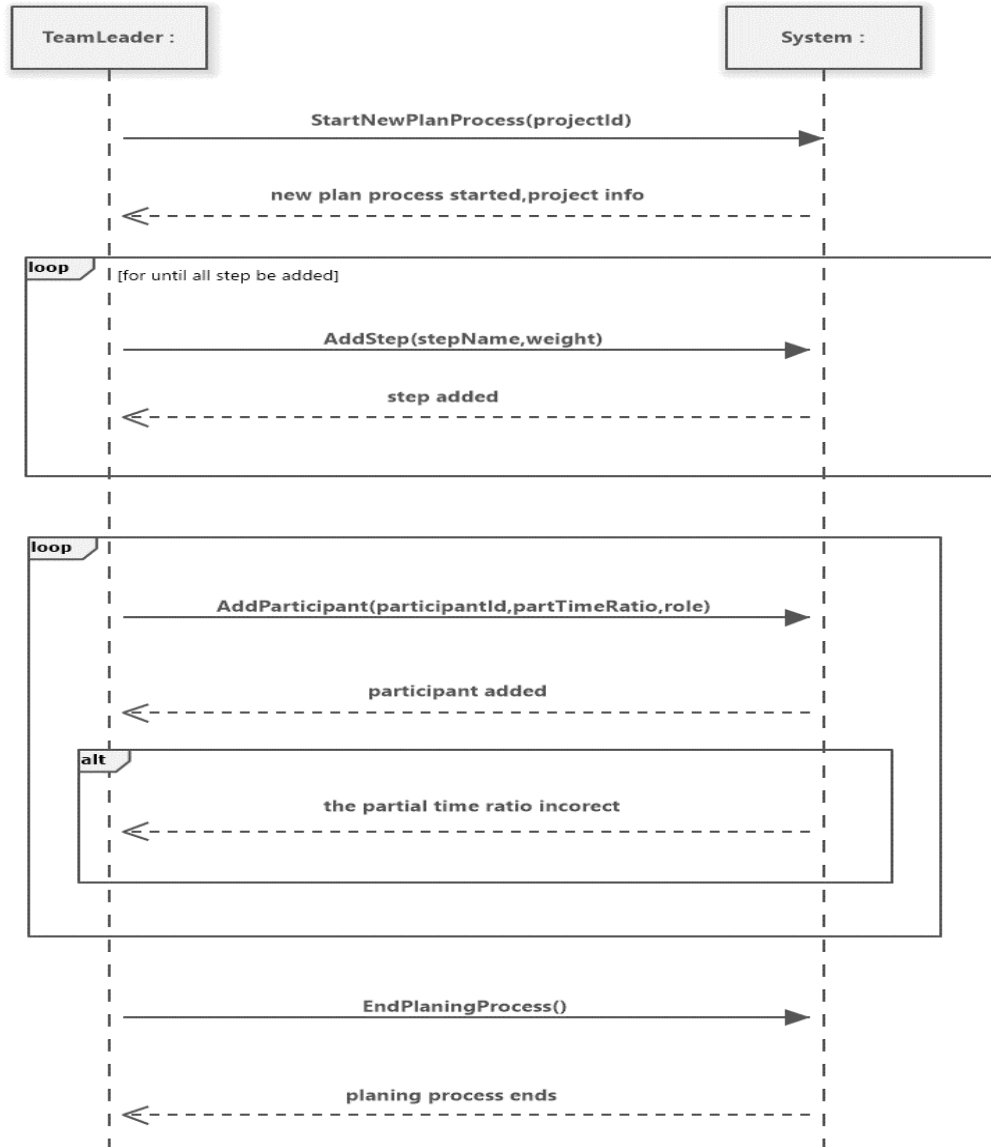
6.5 مخططات التالي لبعض حالات الاستخدام

1.6.5 مخطط التالي لحالة الاستخدام متابعة مشروع



الشكل 11: مخطط التالي لحالة الاستخدام متابعة مشروع

2.6.5 مخطط التالي لحالة الاستخدام تخطيط مشروع



الشكل 12: مخطط التالي لحالة الاستخدام تخطيط مشروع

7.5 عقود عمليات حالة الاستخدام متابعة مشروع

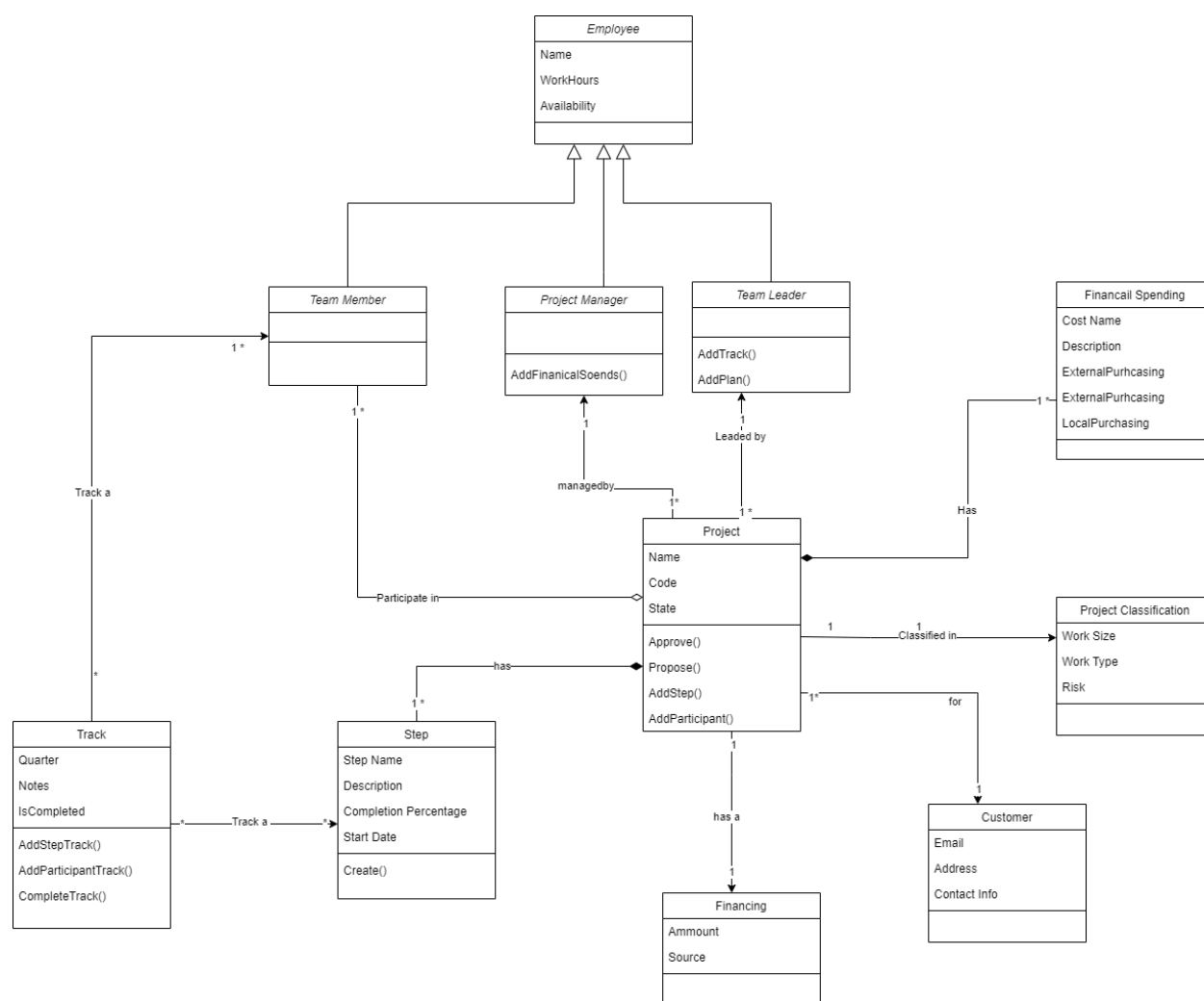
اسم العملية: StartNewTrack(projectId)	
بدء عملية متابعة جديدة	Responsibilities
لا يوجد.	Exception
الستخدام (رئيس فريق العمل) مسجل لدى النظام	Precondition
عملية متابعة جديدة T أنشئت. نسخة من المشروع P أنشئت. عملية المتابعة T أضيفت إلى المشروع P.	Postcondition

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

اسم العملية: AddParticipantTrack(participantId,workHours)	
Responsibilities	إضافة عملية متابعة مشارك.
Exception	لا يوجد.
Precondition	<p>المستخدم (رئيس فريق العمل) مسجل لدى النظام</p> <p>المشارك Participant معرف لدى النظام.</p> <p>عملية المتابعة Track معروفة لدى النظام.</p>
Postcondition	<p>متابعة مشارك ParticipantTrack أنشئت.</p> <p>نسبة ساعات العمل أسندت إلى متابعة المشارك ParticipantTrack.</p> <p>متابعة المشارك ParticipantTrack أضيفت إلى عملية المتابعة Track.</p>

اسم العملية: CompleteTrack()	
Responsibilities	إنهاء عملية المتابعة.
Exception	لا يوجد.
Precondition	<p>المستخدم (رئيس فريق العمل) مسجل لدى النظام</p> <p>عملية المتابعة Track معروفة لدى النظام.</p>
Postcondition	لا يوجد.

8.5 مخطط الصفوف



الشكل 13: مخطط الصفوف

الفصل السادس

الدراسة التصميمية

نبين في هذا الفصل شرحا عن الأساليب والمنهجيات والأنماط التصميمية المستخدمة في تنفيذ النظام.

1.6- 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.

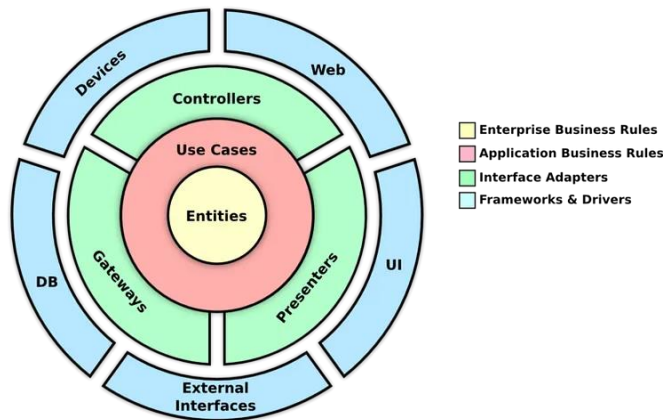
2.6 البنية المعمارية النظيفة – Clean Architecture

1.2.6 مفهوم البنية المعمارية 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 يساهم 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 استخدام البنية المعمارية النظيفة نظام له الخصائص التالية:

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



الشكل 14: مكونات البنية المعمارية النظيفة

2.2.6 الاعتمادية في البنية المعمارية النظيفة

التبعية في البنية المعمارية النظيفة هي علاقة اعتماد بين مكونات النظام المختلفة. تحدّد هذه العلاقة كيفية تفاعل هذه المكونات مع بعضها البعض وكيف تؤثر التغييرات في مكون واحد على المكونات الأخرى. تعتبر قاعدة الاعتمادية Dependency، التي تنص على أن الاعتماديات يجب أن تتجه من الخارج إلى الداخل، مبدأً أساسياً 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 صيانتها وتطويره. من خلال إدارة التبعية بشكل صحيح، يمكن للمطورين بناء أنظمة برمجية أكثر استقراراً وقابلية للتوسع.

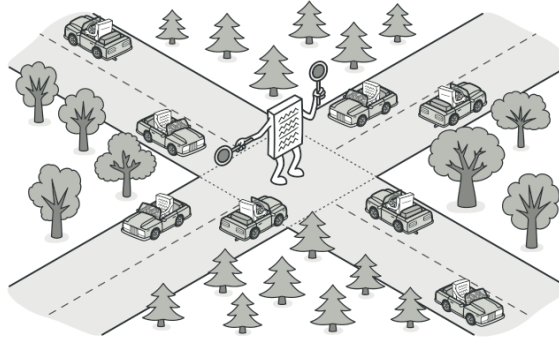
3.2.6 مكونات البنية المعمارية النظيفة

- 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 لتشغيل مثل أطر عمل الويب ومكتبات واجهة المستخدم وقواعد البيانات وغيرها.

3.6 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.

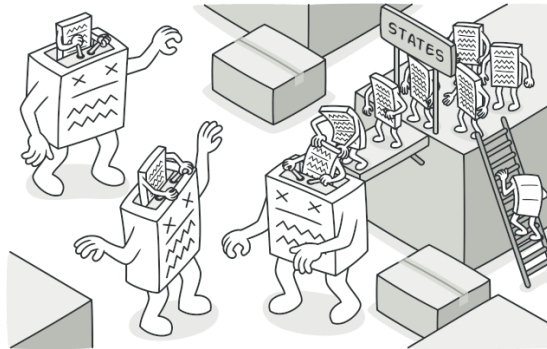
ويأتي دور النمط الوسيط لتبسيط هذا التعقيد عن طريق إدخال كائن وسيط واحد يتحكم في جميع التفاعلات بين الكائنات. بدلاً من أن تتواصل الكائنات مباشرة مع بعضها البعض، يتم توجيه جميع التفاعلات من خلال الوسيط. هذا يقلل من التبعيات بين الكائنات ويجعل النظام أكثر تنظيماً وأسهل في التعديل والتوسيع.



الشكل 15: مثال توضيحي عن النمط الوسيط

4.6 النمط التصميمي الحالة State Design Pattern

وأحد 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، اعتماداً على السياق الذي يوجد فيه.



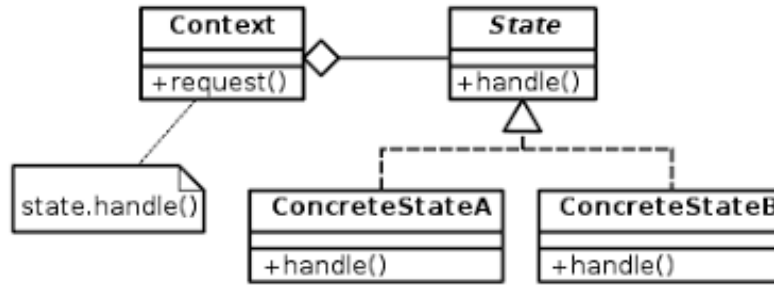
الشكل 16: مثال توضيحي عن نمط الحالة

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 17: مخطط صفوف تجريدي يصف نمط الحالة

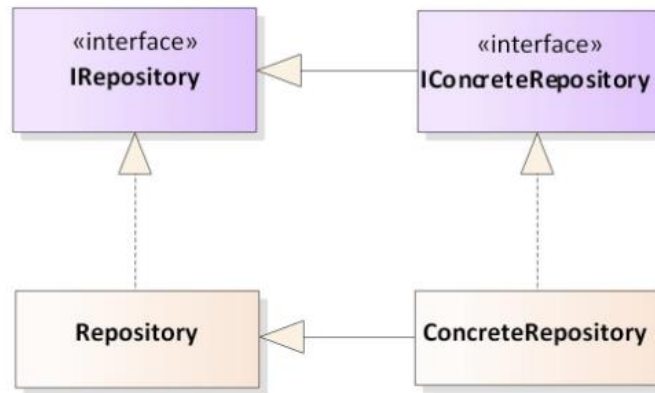
5.6 فصل مسؤوليات الأوامر والاستعلامات Command Query Responsibility Segregation

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.

6.6 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 18: مخطط صفوف تجريبي للنمط المستودع

7.6 غط التصميم Result Pattern

الفصل السابع

تصميم النظام

نعرض في هذا الفصل بعض المخططات للبنية العامة النظام.

1.7 مبادئ التصميم

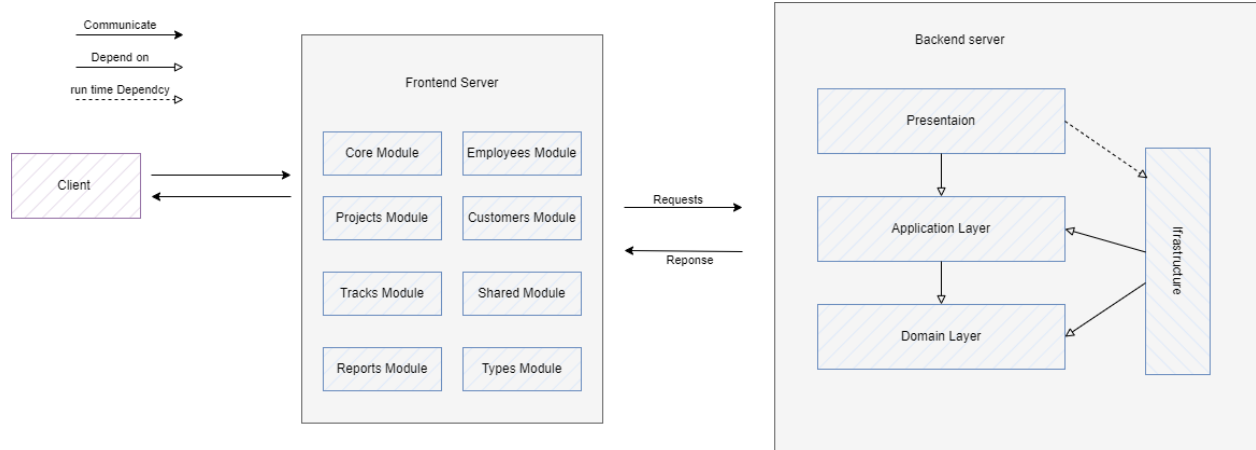
إن بناء تصميم جيد وقابل للتوسع يتحقق عندما نضع أسسا ومبادئ واضحة لهيكل الحل وتقسيمه وطريقة واضحة للشروع في تنجيزه.

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

2.1.7 بنية عامة للتصميم

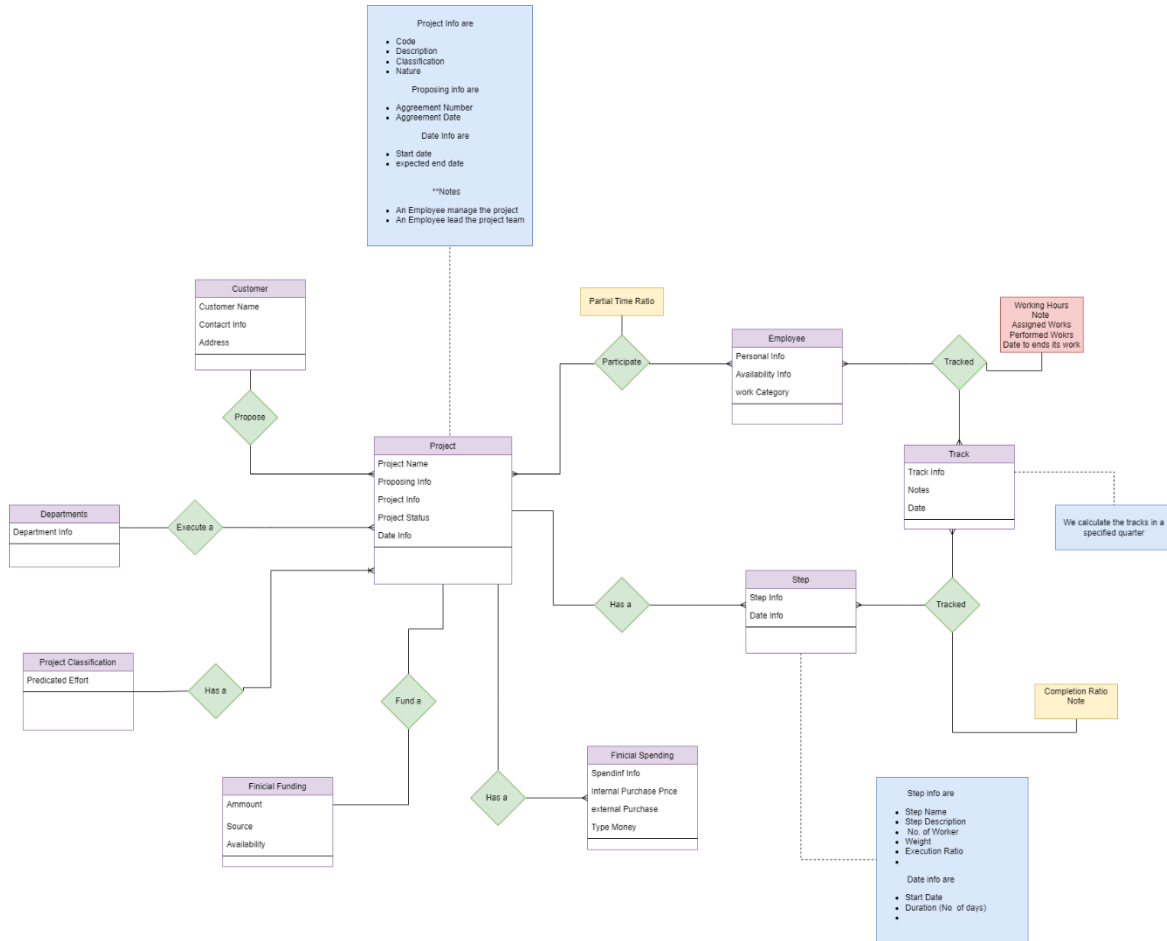
يبين الشكل (X) شكلاً مبسطاً يوضح مكونات النظام، حيث يتكون النظام من جزئين: تطبيق الواجهة الخلفية (Backend) الذي اعتمدنا في تصميمه وتنجيذه على البنية المعمارية النظيفة (Clean Architecture) وبعض مبادئ التصميم المقاد بالمجال (Domain Driven Design)، والجزء الثاني هو طرف الواجهة الأمامية واعتمدنا في تصميمه على مبدأ التقسيم وفق الميزات (Feature Based Decomposition)، حيث قسمنا طرف الواجهة الأمامية إلى مجموعة من الوحدات (Modules) المستقلة بعضها عن بعض.

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



الشكل 19: بنية عامة للتصميم المقترح

2.7 مخطط العلاقات بين الكائنات Entity Relationship Diagram



الشكل 20: مخطط العلاقات بين الكيانات ERD

الفصل الثامن

الأدوات وأطر العمل المستخدمة

نبين في هذا الفصل ما استخدمناه من أدوات وأطر عمل.

1.8 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.

2.8 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 (Single Page Applications - SPAs) وخدمات حقن التبعية Dependency Injection التي تعزز من قابلية الصيانة واختبار التطبيق. بالإضافة إلى ذلك، توفر أداة Angular CLI دعمًا متكاملًا لإعداد المشروع، بناء التطبيق، وإجراء الاختبارات، مما يعزز من إنتاجية المطورين.

3.8 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 و Linux، 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.

4.8 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.

5.8 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.

5.8 المكتبة MediatR

هي أداة فعّالة تمكّن من تطبيق نمط التصميم الوسيط (Mediator Pattern) بشكل سهل وبسيط في بيئة .NET. فهي تعتمد 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 التاسع

تنجيز النظام

نقدم في هذا الفصل شرحا مفصلا لمنهجية تنجيز النظام.

1.9 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 منهجية التقسيم وفق الميزات مما يساعد على جعل التطبيق قابلاً للفهم والصيانة ومرناً.

2.9 البنية المعمارية لطرف المخدم

1.2.9 طبقات الحل المقترح

لقد اعتمدنا في تصميم طرف المخدم على البنية المعمارية النظيفة، لذلك نجد أن الحل المقترح هو حل مركز حول المجال ومنطق العمل ومستقل عن طرق العرض وطرق 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 لا تعتمد على 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.

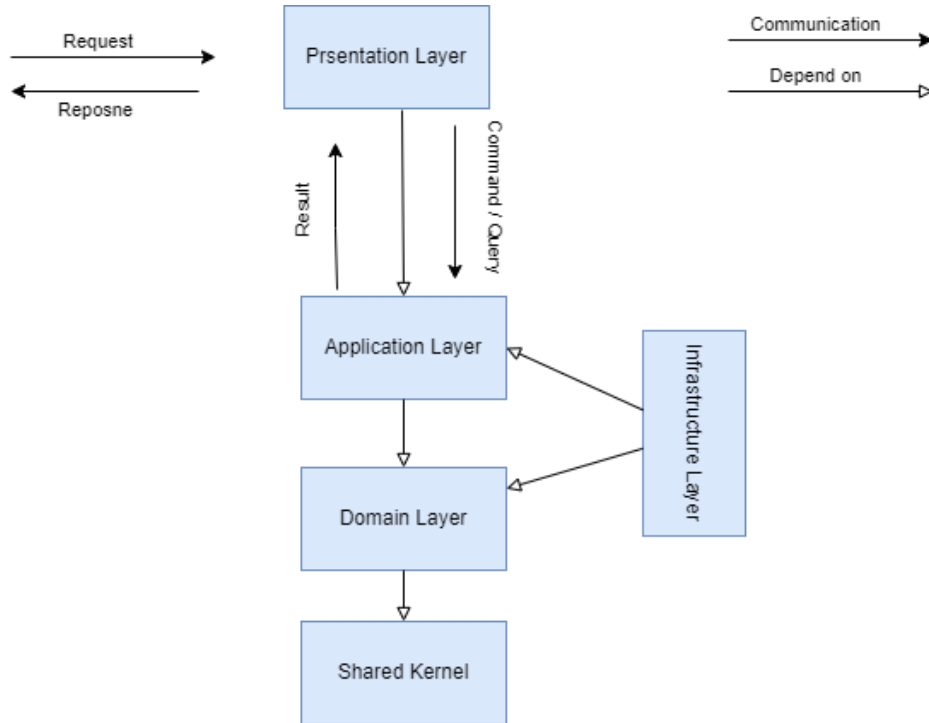
4. طبقة العرض

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 تخزين 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



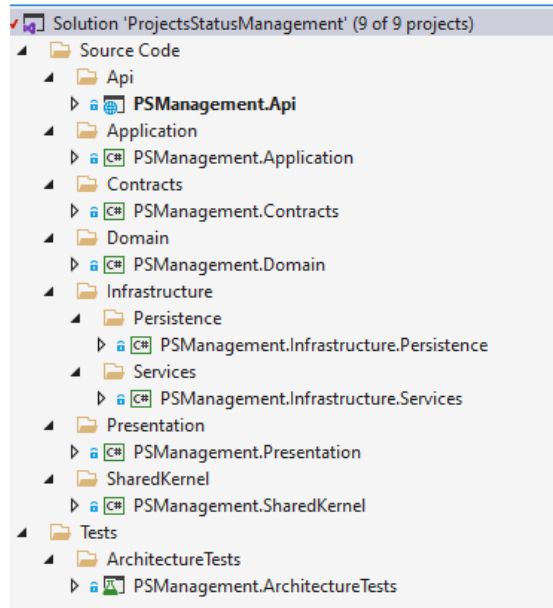
الشكل 21 البنية المعمارية لطرف الواجهة الخلفية

2.2.9 هيكلية مجلدات طرف المخدم

يتكون الحل لدينا من مجلد الرماز المصدري ومجلد الاختبارات.

ويتكون الرماز المصدري للحل من ثمانية مشاريع تحقق المطلوب ويتكون مجلد الاختبارات من مشروع اختبار البنية المعمارية.

وهي كما هو مبين في الشكل أدناه من خلالها تم تحقيق الحل المطلوب.



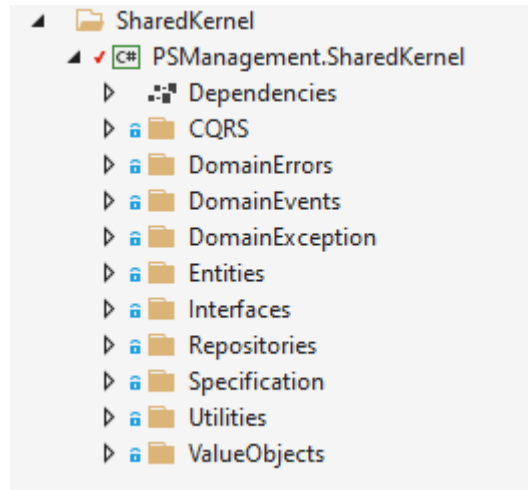
الشكل 22 هيكلية الحل المنجز

3.2.9 طبقة النواة المشتركة Shared Kernel

تحتوي هذه الطبقة على مفاهيم تجريدية صالحة للاستخدام في أي مشروع يتبع البنية المعمارية النظيفة ومبادئ التصميم المقاد بالجمال. فهي تحتوي على مفاهيم مجردة ولا تحتوي على منطق عمل وإنما تجريد للمفاهيم فقط.

ويوضح الشكل أدناه مكونات الطبقة حيث تحتوي على:

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



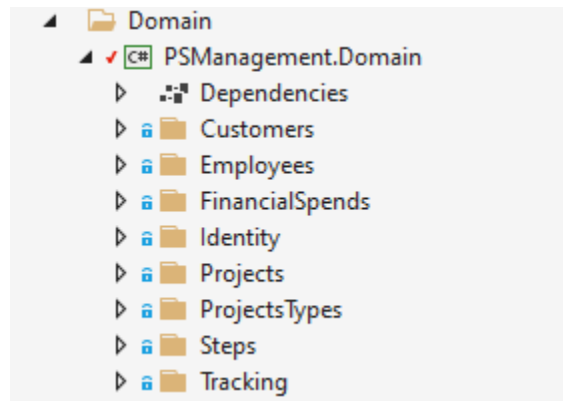
الشكل 23: بنية طبقة النواة المشتركة

3.2.9 طبقة المجال Domain Layer

1.3.2.9 الهيكلية العامة للطبقة

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

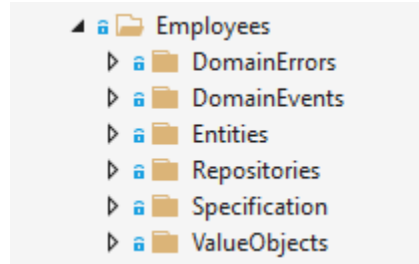
- مجال المشاريع نهتم فيه بتعريف المشروع وإجرائته وإدارة مرفقاته وإدارة أطواره.
 - مجال أنواع المشاريع نعرف فيه أنواع المشاريع.
 - مجال الجهات الطارحة ندير فيه الجهات الطارحة ومعلومات التواصل معها.
 - مجال الموظفين نعرف فيه البيانات التي تهمنا ضمن مسألتنا عن الموظف والتي نكاملها ونأخذها من نظام الذاتية.
 - مجال المتابعة نهتم فيه بمتابعة المشروع والمراحل وعمل الموظفين ونغلف فيه إجراءات المتابعة.
 - مجال المراحل نهتم فيه بتعريف المراحل وإدارتها.
 - مجال خطة الإنفاق ندير فيه خطة الإنفاق ومعلوماتها.
- كما يوجد مجلد الهوية Identity نعرف فيه المستخدمين وصلاحياتهم وأدوارهم.
- ويبين الشكل أدناه هذه المجالات.



الشكل 24: بنية طبقة التطبيق.

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

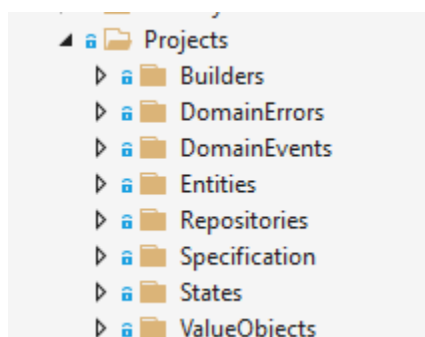
- الكائنات Entities : نعرف فيه الكائنات ضمن المجال وهي تغلف قواعد العمل وإصدار الأحداث.
- الأحداث Events: حيث نعرف الأحداث ضمن هذا المجال ونُدع للكائنات إصدارها وفق لقواعد العمل وتساعد هذه الأحداث باستخراج عمليات النظام التي لا تنتمي لسيناريو العمل الأساسي ومعالجتها في مرحلة لاحقة كما في حالة تسجيل تبدلات العامل عند تغيير مشاركته.
- الأخطاء Errors: نعرف فيه الأخطاء ضمن المجال وهي أخطاء تحقق Validation Errors تستخدمها طبقة التطبيق.
- المستودعات Repositories: وهو مجرد عقود عمليات تخزين البيانات واسترجاعها، فهو يَمَكِّننا 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 التصفية عليها (pagination).



الشكل 25: البنية الضمنية في طبقة التطبيق

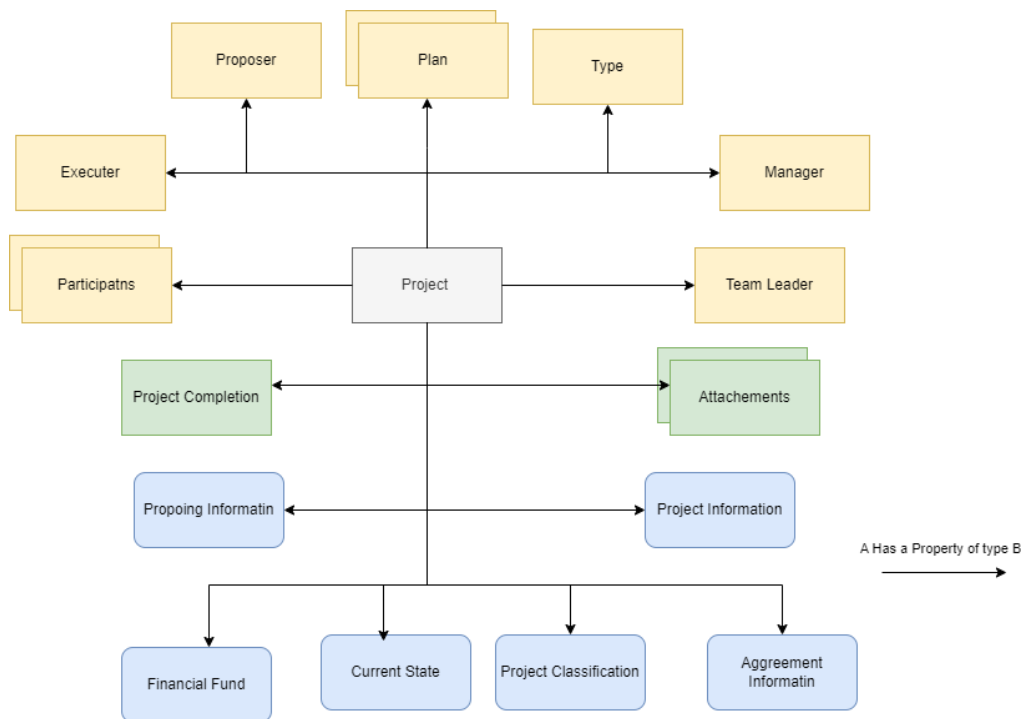
2.3.2.9 مجال المشاريع

إن هذا المجال يمثل أساس النظام أي هو المحور الأكبر في هذا النظام. وعلى عكس بقية المجالات فهو يحوي عددا من المجلدات غير الموجودة في البقية سنشرحها هنا.



الشكل 26: بنية مجال المشاريع

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



الشكل 27: 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 الطرح وتاريخه . لماذا هذا غرض قيمة؟ لأنه على الرغم من أن كتاب الطرح وتاريخه معلوماتان مختلفتان، لكن لهما دلالة وهي معلومة الطرح التي تتغير كلياً بتغير أحدهما أي ليس غرضاً ذا هوية على عكس كائن المرفق الذي يعرف بمعرف (Id) وتغيير عنوان المرفق لا يغير المرفق بذاته. وهكذا نميز بين أغراض القيمة والكائنات.

يحتوي هذا المجال أيضاً على مجلد الباني،

ففي هذا المجال اضطررنا إلى استخدام النمط التصميم الباني (Builder Design Pattern). يساعدنا هذا النمط في تقديم واجهة موحدة لبناء غرض من هذا الصف فهذا الصف يحتوي على عدد كبير من الواصفات ونحن في طبقة المجلد لا نعلم 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 لا نعلم أي يحتاج الكائن الذي سيتعامل مع الغرض المنشأ. لذلك يكون الحل هو بكتابة بان واحد يأخذ جميع الواصفات وترك عملية إسناد القيم للصف Project Builder ويوضح الشكل أدناه طريقة استخدامنا 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 28: مثال على استخدام الصف الذي يحقق نمط الباني

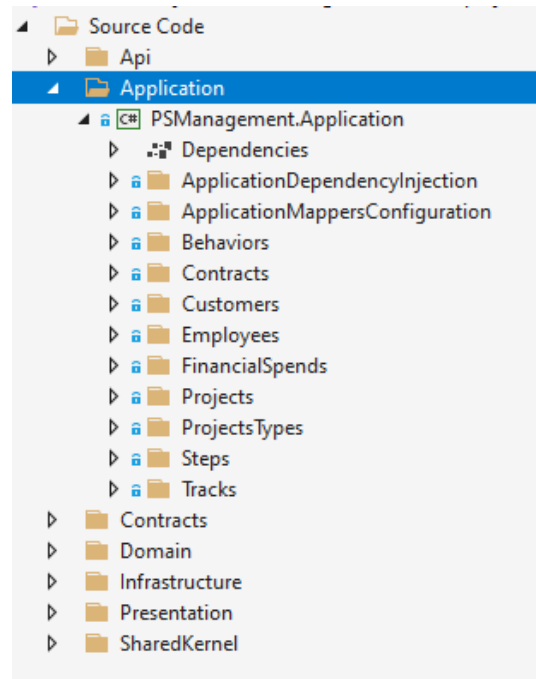
كما في هذا المجال استخدمنا النمط التصميمي الحالة لإدارة أطوار المشروع والانتقالات بينها. حيث بين الشكل (x) مخطط الصفوف المستخدم. حيث عرفنا صف مجردا يحوي الطرائق التي تعرف الانتقال بين الحالات Complete, Plan , Approve, Propose, Cancel, Complete وتترك مسؤولية التنجيز الفعلي إلى الحالة المعنية التي يقع 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 29: مثال على آلية تغليف الحالة لمؤثر الانتقال

4.2.9 طبقة التطبيق Application Layer

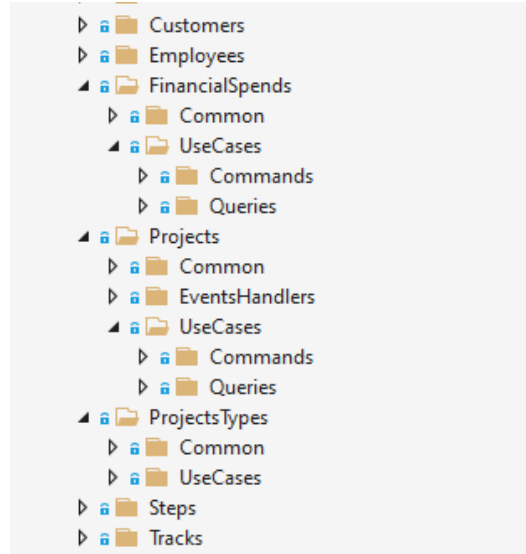
مسؤولية هذه الطبقة هي تغليف طبقة المجال وإضافة منطق العمل كما تعرف حالات الاستخدام التي تتيحها للطبقات الأعلى.



الشكل 30: بنية طبقة التطبيق.

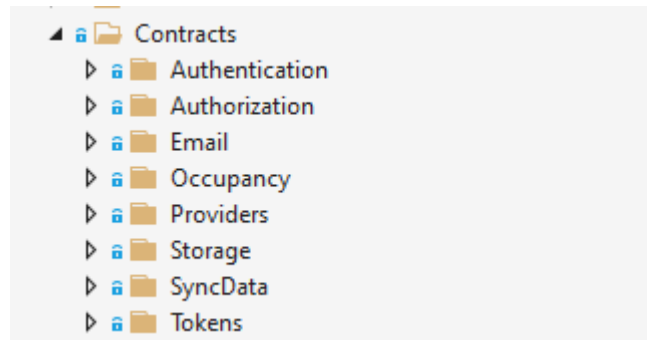
في هذه الطبقة نجد لدينا شقين أساسيين هما:

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



الشكل 31: البنية الضمنية في كل مجال.

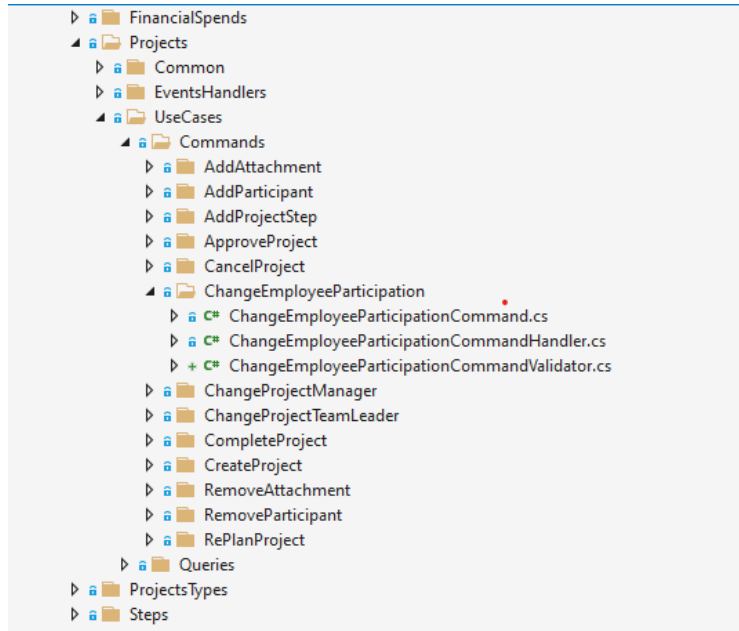
والشق الثاني تعريف لعقود خدمات (Interfaces) يتم التعامل معها دون الاهتمام بالتنجيز الفعلي مما يساعد على جعل منطق العمل في بعض الحالات مستقلا عن طريقة التنجيز الفعلية.



الشكل 32: العقود ضمن طبقة التطبيق.

استخدمنا في هذه الطبقة لتحقيقها النمط التصميمي (CQRS) Command Query Responsibility Segregation 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 33: مثال على تنجيز حالة استخدام بالنمط CQRS

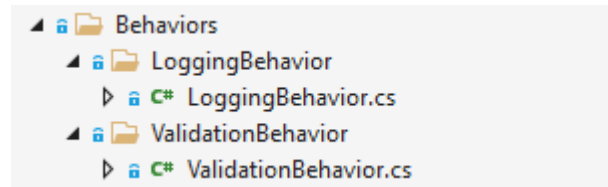
يتم تنجيز حالة استخدام ما بفصلها إلى شقين الأمر ومعالج هذا الأمر. يتم استخدام هذه الحالة من الطبقة الأعلى من خلال إرسال الطب ChangeEmployeeParticipationCommand بعد تحديد قيمه المطلوبة ضمنه (الرقم المعرف للموظف ومعرف الزبون ومعلومات المشاركة الجديدة) ويترتب على عاتق الطبقة من خلال النمط الوسيط (Mediator) 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);
```

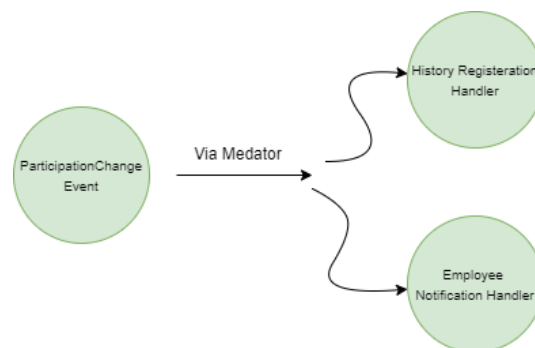
وفي سياق النمط التصميمي الوسيط علينا أن نذكر أيضا أن المكتبة MediatR عندما قامت بتنجيذه أتاحت لنا تعريف خط تسلسل من السلوك Pipeline Behavior يقاطع الطلب ويحصل على سياق كائن الأمر أو الاستعلام قبل تسليمه للمعالج المسؤول عن معالجة الأمر أي باختصار قبل أن يصل الأمر ChangeEmployeeParticipationCommand إلى 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 آخر وهو سلوك التحقق Validation Behavior المسؤول عن التحقق من أهلية البيانات المدخلة أي لكي نمنع في حالة تغير مشاركة موظف من إدخال قيم سالبة لنسبة التفرغ وبعض الشروط الأخر التي عرفناها في الصف ChangeEmployeeParticipationCommandValidator ويوضح الشكل أدناه السلوكين اللذين عرفناهما.



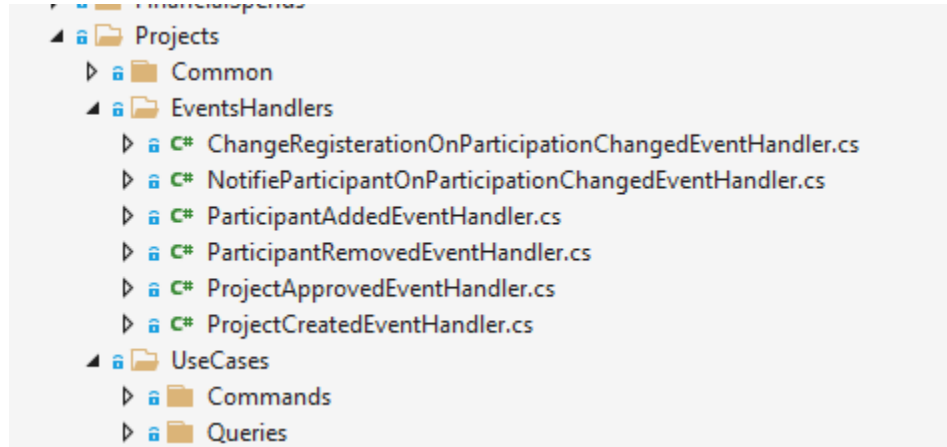
سلوك التسجيل مسؤوليته التحقق من أهلية البيانات الواردة. وسلوك التسجيل الذي يسجل الاستعلامات والأوامر المهمة. ومن أجل عدم تسجيل جميع الاستعلامات والأوامر لأن ذلك يخفف الأداء وليست جميع الأوامر والاستعلامات مهمة قمنا بفصل الاستعلامات إلى نوعين Loggable Commands و Commands وكذلك الأمر بالنسبة للاستعلامات 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 يهمله أمر معالجته يستقبله من خلال أيضا النمط الوسيط الذي بدوره يسلمه للمعالج المناسب الذي يعلمه من خلال تنجيذه للواجهة IEventHandler وفي سياق هذا نوضح ماذا ستفقدنا منه.



الشكل 34: 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 تقريراً 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 لا يصح لأن جمع إجراءات إرسال الإعلام وإجراءات تسجيل تاريخ المشارك ضمن حالة الاستخدام ذاتها يجعلها غير مترابطة فيما بينها وليست ذات مسؤولية واحدة أي يخرق مبدأ أن تصميميان هما High Cohesion و Single Responsibility لذلك قمنا بهذا الفصل الذي يجعل مكونات معالجة حالة الاستخدام مترابطة ومستقلة عن الإجراءات التي تتم عند وقوع الحدث.



الشكل 35: بعض معالجات الأحداث في مجال المشاريع.

وأيضا من الأنماط المستخدمة في هذه الطبقة هو النمط Unit Of Work المعروف 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 BeginTransaction, RoleBack, Commit حيث تكون العملية Begin Transaction عن بدء مناقلة معزولة وعند الانتهاء من حالة الاستخدام ننهي المناقلة أي تنطبق وهذا المفهوم مشابه لمفهوم المناقلات في قواعد المعطيات.

وأيضاً تم استخدام نمط النتيجة Result Pattern المقدم من المكتبة Ardalis.Result للتحكم بسير العمل في هذه الطبقة عوضاً 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 36: مثال عن استخدام نمط النتيجة

5.2.9 طبقة العرض

1.5.2.9 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

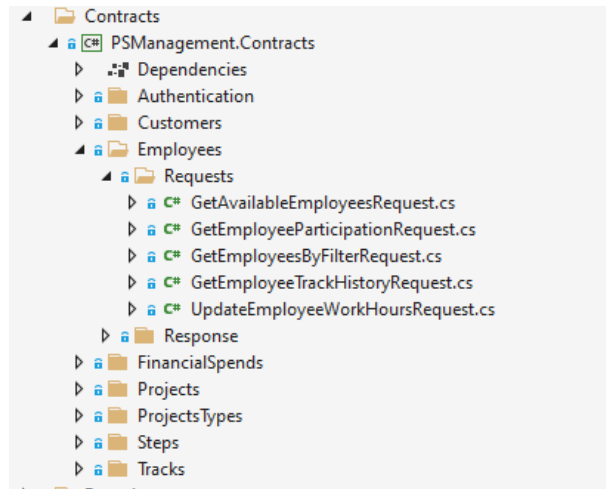
إن لكل من هذه المشاريع دوره الواضح والحاجة إليه والفائدة التي يقدمها، فمشروع التعاقدات يعرف الطلبات التي ترد إلى النظام والردود عليها، بينما يعرف مشروع العرض النقاط الطرفية في كل مجال ومهمته هي فقط أخذ الطلب وتحويله إلى أمر أو استعلام وتسليمه عن طريق النمط الوسيط إلى معالج الحدث ومن ثم إعادة النتيجة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

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

أ. مشروع التعاقدات

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

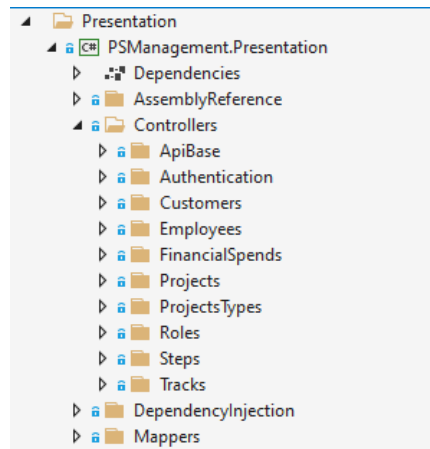
وهذا المشروع يحوي على ستة مجلدات تمثل المجالات الموضحة في مبادئ التصميم وفي كل مجلد يوجد مجلد للطلبات وآخر للردود كما هو مبين في الشكل أدناه. حيث يساعد هذا المشروع على تقديم طريقة واضحة لتوثيق النظام أي نستطيع من خلاله أن نعرف بشكل دقيق دون اطلاع أحد على طبقة التطبيق بجميع الطلبات الممكن إرسالها.



الشكل 37: بنية مشروع التعاقدات

ب. مشروع العرض

يحتوي 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 38: 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 39: مثال على حقن الاعتماديات واختيار البنية التحتية

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

6.2.9 طبقة البنية التحتية

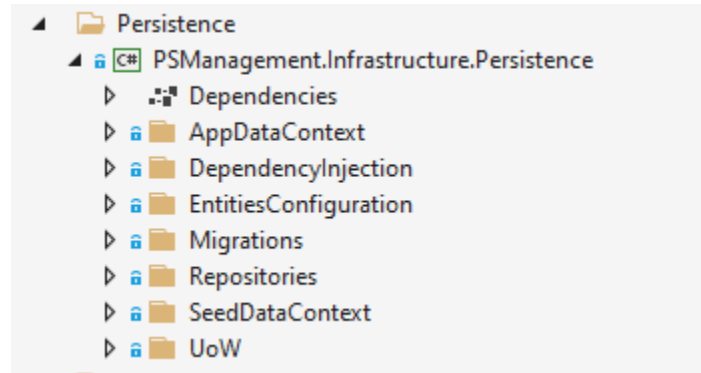
1.6.2.9 الهيكلية العامة

تم تحقيق هذه الطبقة من خلال مشروعين أولاً 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.

2.6.2.9 مداومة البيانات Data Persistence

أ. هيكلية عامة

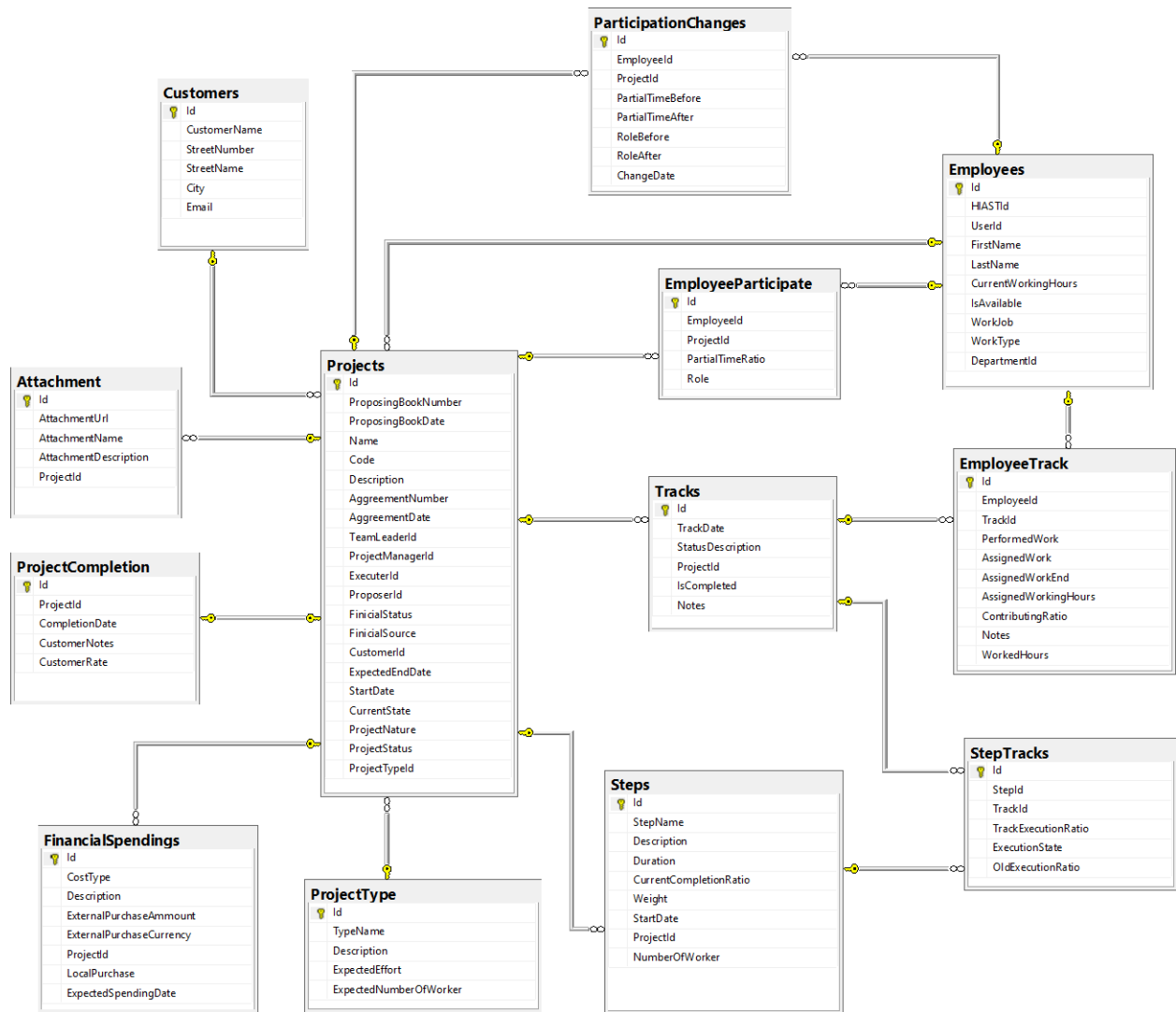
في هذا المشروع تم تعريف طريقة التخزين الفعلية للبيانات كما قمنا فيه بتنجيز المستودعات الموجودة عقودها في طبقة المجال. ويوضح الشكل أدناه هيكلية المشروع.



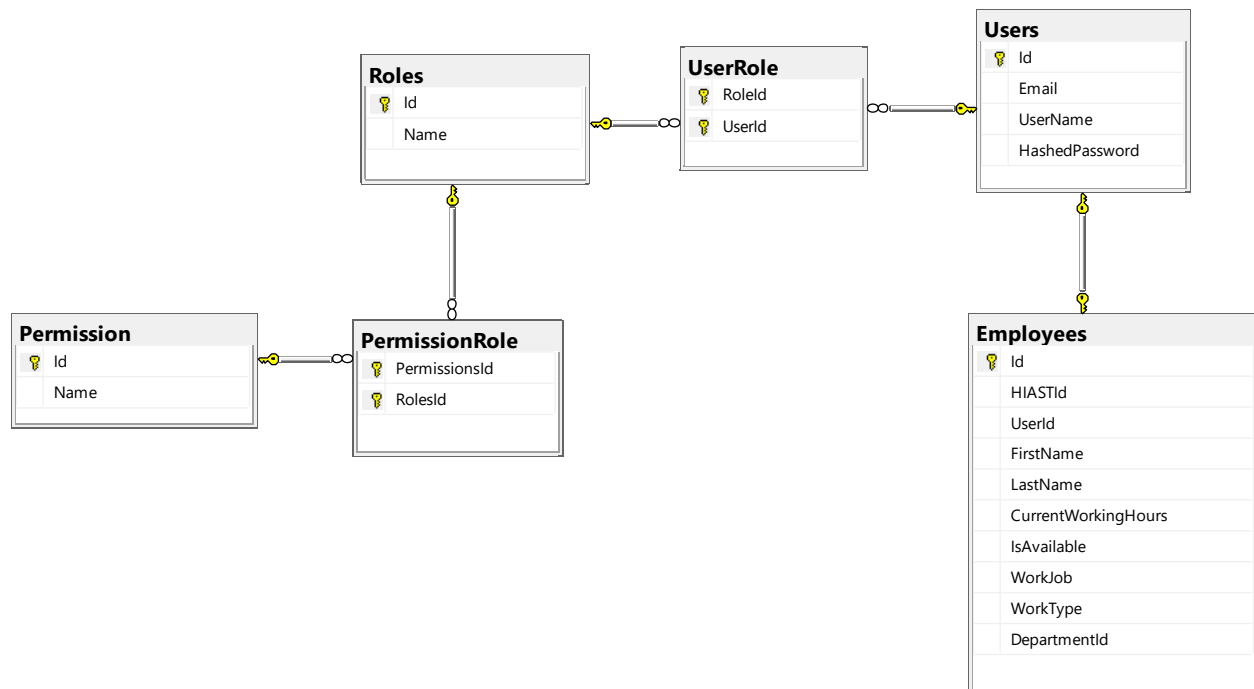
الشكل 40: بنية مشروع مداومة البيانات

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 41: مخطط العلاقات ضمن قاعدة المعطيات



الشكل 42: 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 يتم 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 والتنفيذ

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

حيث عرفنا لدينا الأدوار الآتية:

المخطط وهو مسؤول عن إدارة أنواع المشاريع والجهات الطارحة.

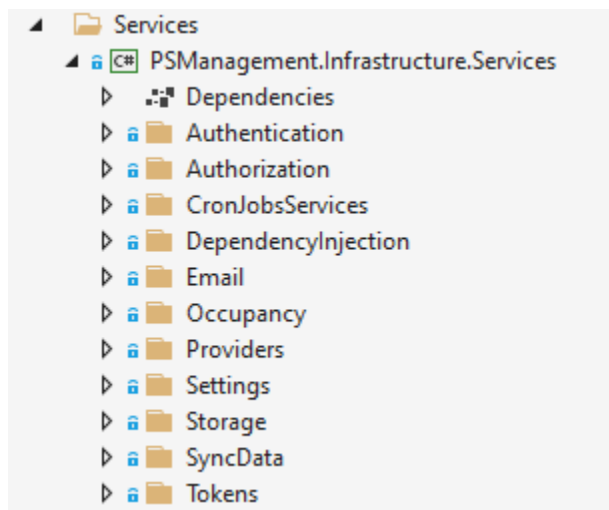
الموظف وهو يتمتع بصلاحيات تخطيط ومتابعة المشاريع.

الوكيل العلمي وهو يتمتع بصلاحيات طرح المشاريع إلغائها واستعراضها جميعا.

مدير النظام وهو مسؤول عن إدارة المستخدمين وأدوارهم.

2.6.2.9 الخدمات Services

في هذا المشروع نم تنجيز العقود المعرفة في طبقة التطبيق.



الشكل 43: بنية مشروع الخدمات

من الخدمات الأساسية 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 لاهتم في كيفية حفظ الملفات وإنما 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 التأخير 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 قاعدة البيانات التي لدينا. إن تنجيز هذه الخدمة وهمي أي نقرأ من مصدر غير حقيقي لأن المصدر الرئيسي عن طريق LDAP لا يتاح 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 ما هو 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 الموظفين لدى نظامنا مع البيانات الموجودة على مخدّم 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.

3.9 بنية الواجهة الامامية Front-end

1.3.9 منهجية التصميم

تم بناء الواجهة الامامية باستخدام إطار العمل Angular. ومن أجل تحقيق مبادئ التصميم التي ذكرناها سابقا ولفصل هذه المجالات في تطبيق الواجهة الامامية 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.

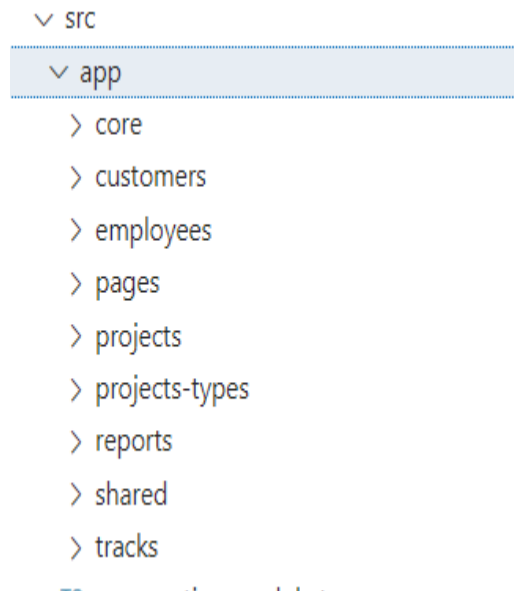
يهدف هذا النوع من التقسيم إلى تعزيز مبدأ التماسك العالي High Cohesion والترابط المنخفض Low Coupling.

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

2.3.9 بنية تطبيق الواجهة الامامية

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

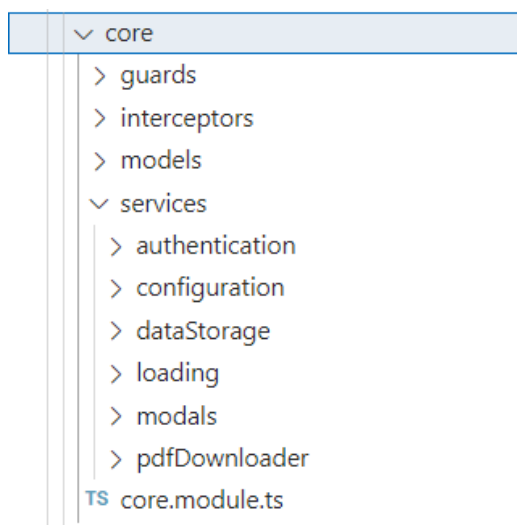


الشكل 44: الوحدات في تطبيق الواجهة الامامية

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

تحتوي الوحدة المشتركة على المكونات المشتركة بين الوحدات الأخرى وقالب الصفحة أي تحتوي فقط مكونات للعرض.

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



الشكل 45: بنية وحدة النواة Core Module

وبالنسبة لبقية الوحدات فهي كما ذكرنا تتكون من المجلدات الآتية:

أ. المكونات Components

ب. النماذج Models : يحتوي طلبات الإرسال والردود

ت. الصفحات Pages: يحتوي الصفحات المستقلة

ث. التوجيه Routing

ج. الخدمات Services: يحتوي على خدمات تحقق الاتصال مع تطبيق المخدم وتطلب البيانات.

ويوضح الشكل أدناه هذه البنية.

```

  ✓ customers
    > components
    > models
    > pages
    > routing
    > services
    TS customers.module.ts
  ✓ employees
    > components
    > models
    > pages
    > routing
    > services
    TS employees.module.ts

```

الشكل 46: البنية الضمنية في الوحدة

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

```

  ✓ pages
    > home
    > login
    > page-notfound
    > project-faq
    > psm-start

```

الشكل 47: الصفحات الساكنة في التطبيق

الفصل العاشر

الواجهات والاختبارات

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

1.10 مقدمة

تعد الاختبارات من الركائز الأساسية لضمان جودة الأنظمة البرمجية وتعمل على التأكد من أن جميع مكونات النظام تعمل كما هو متوقع منها. تتنوع أنواع الاختبارات بين اختبارات الوحدة (Unit Testing)، واختبارات التكامل (Integration Testing)، واختبارات القبول (Acceptance Testing) وغيرها.

فيما يتعلق باختبارات البنية المعمارية (Architecture Tests)، فإنها تُستخدم لضمان أن بنية النظام تتبع المبادئ التصميمية المحددة وتتحقق من صحة تطبيق الأنماط التصميمية وفصل الطبقات وعدم وجود التبعيات غير المرغوب فيها. تهدف هذه الاختبارات إلى التأكد من أن كل طبقة في النظام مستقلة وقابلة للصيانة والتوسع، مما يساهم في تحسين أداء النظام واستقراره على المدى الطويل.

يُعدّ إجراء اختبارات البنية المعمارية أمرًا حيويًا فهو يساهم في ضمان أن النظام يمثل للمواصفات المعمارية المقررة، مما يساعد في تقليل الأخطاء وتحسين جودة الحل البرمجي بشكل عام.

2.10 اختبار البنية المعمارية

تم إجراء نوعان من الاختبارات الأول هو اختبار الاعتمادية بين الطبقات والثاني هو اختبار احتراك قواعد التسمية.

أ. الاعتمادية بين الطبقات:

تم إجراء خمسة اختبارات للتأكد من أن اعتماديات النظام تحترم مبادئ البنية المعمارية النظيفة، وهي:

- لا يجب أن تعتمد طبقة المجال سوى على طبقة الموارد المشتركة.
- لا يجب أن تعتمد طبقة التطبيق سوى على طبقة المجال.
- لا يجب أن يعتمد مشروع الخدمات في البنية التحتية سوى على طبقة التطبيق.
- لا يجب أن يعتمد مشروع مداومة البيانات سوى على طبقة المجال.
- لا يجب أن تعتمد طبقة العرض سوى على طبقة التطبيق.

ونلاحظ في الشكل أدناه أن هذه الاختبارات جميعها نجحت.

Test	Duration	Traits	Error Message
PSManagement.ArchitectureTests (12)	1.4 sec		
PSManagement.ArchitectureTests (12)	1.4 sec		
DependencyTests (5)	697 ms		
Services_Should_Not_DependOnOtherProjectExceptApplication	11 ms		
Presentation_Should_Not_DependOnOtherProjectExceptApplicationAndContracts	67 ms		
Persistence_Should_Not_DependOnOtherProjectExceptDomain	283 ms		
Domain_Should_Not_DependOnOtherProjectExceptSharedKernel	174 ms		
Application_Should_Not_DependOnOtherProjectExceptDmain	162 ms		
ArchitectureDomainTests (2)	215 ms		
ArchitectureApplicationTests (5)	445 ms		

الشكل 48: اختبار الاعتماديات

ب. احترام قواعد التسمية:

تم إجراء اختبارات للتأكد من قواعد التسمية للأحداث والأوامر والاستعلامات في طبقة التطبيق، وهي:

- يجب أن تنتهي أسماء جميع الأوامر باللاحقة Command.
- يجب أن تنتهي أسماء جميع الاستعلامات باللاحقة Query.
- يجب أن تنتهي أسماء معالجات الأحداث باللاحقة QueryHandler.
- يجب أن تنتهي أسماء جميع معالجات الأوامر باللاحقة CommandHandler.
- يجب أن تنتهي أسماء جميع معالجات الأحداث باللاحقة EventHandler.

ويوضح الشكل أدناه نجاح هذه الاختبارات.

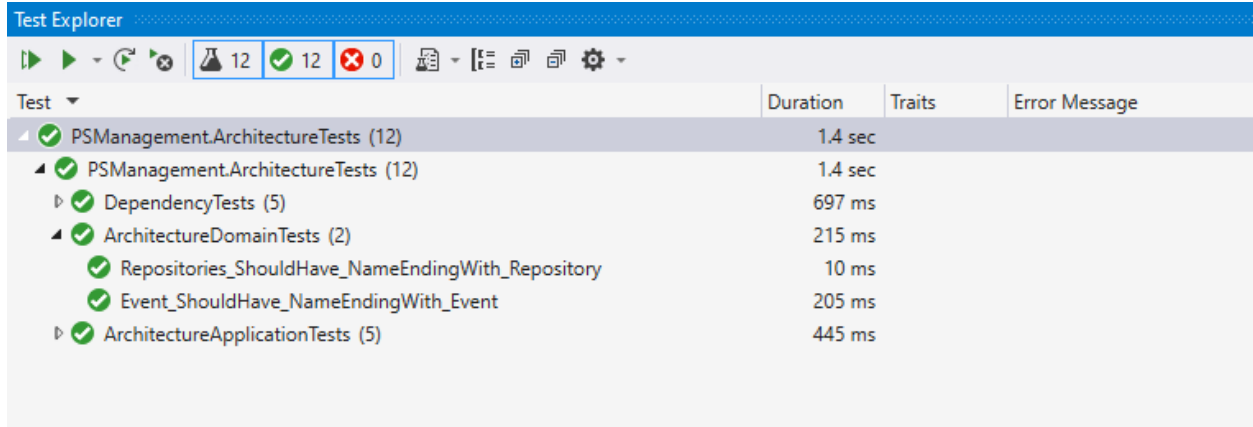
Test Explorer			
<div> <div>12</div> <div>12</div> <div>0</div> </div>			
Test	Duration	Traits	Error Message
PSManagement.ArchitectureTests (12)	1.4 sec		
PSManagement.ArchitectureTests (12)	1.4 sec		
DependencyTests (5)	697 ms		
ArchitectureDomainTests (2)	215 ms		
ArchitectureApplicationTests (5)	445 ms		
QueryHandlers_ShouldHave_NameEndingWith_QueryHandler	20 ms		
Queries_ShouldHave_NameEndingWith_Query	88 ms		
EventHandler_ShouldHave_NameEndingWith_EventHandler	44 ms		
Commands_ShouldHave_NameEndingWith_Command	205 ms		
CommandHandlers_ShouldHave_NameEndingWith_CommandHandler	88 ms		

الشكل 49: اختبارات قواعد التسمية في طبقة التطبيق

تم إجراء اختبارات للتأكد من قواعد التسمية للأحداث والمستودعات في طبقة المجال، وهي:

- يجب أن تنتهي اسماء جميع الأحداث باللاحقة Event.
- يجب أن تنتهي اسماء جميع المستودعات باللاحقة Repository.

ويوضح الشكل أدناه نجاح هذه الاختبارات.



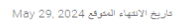
The screenshot shows the Test Explorer window with a blue header. Below the header is a toolbar with icons for running tests, a summary bar showing 12 passed tests (green checkmarks), 12 successful assertions (green checkmarks), and 0 failed tests (red X). The main table lists the test hierarchy and results:

Test	Duration	Traits	Error Message
PSManagement.ArchitectureTests (12)	1.4 sec		
PSManagement.ArchitectureTests (12)	1.4 sec		
DependencyTests (5)	697 ms		
ArchitectureDomainTests (2)	215 ms		
Repositories_ShouldHave_NameEndingWith_Repository	10 ms		
Event_ShouldHave_NameEndingWith_Event	205 ms		
ArchitectureApplicationTests (5)	445 ms		

الشكل 50: اختبار قواعد التسمية في طبقة المجال

أ. بطاقة تعريف مشروع

طباعة | تقرير الجاز | تقرير خطة زمنية



ب. تقرير متابعة مراحل

بطاقة متابعة مراحل

تقرير خطة زمنية

تقرير ايجاز

تقرير متابعة الممالين

طباعة



أتمتة أعمال مديرية التخطيط

بناء نظام معلوماتي لأتمتة أعمال مديرية التخطيط

تاريخ البدء Aug 26, 2024

تاريخ الانتهاء المتوقع May 29, 2024

ذاتية المشروع

رمز المشروع: QWE-2

كتاب طرح : Aug 27, 2024 / 23434

كتاب الموافقة : Aug 29, 2024 / 234

التمويل : ممول / المركز

معلومات حالة المشروع

المرحلة التطوير : مكتمل

نوع المشروع : مشروع تطويري

طبيعة المشروع : عادي

المعلومات التنفيذية

الجهة المنفذة: قسم المعلومات

رئيس فريق العمل: حسن خضور

مدير المشروع : حسن خضور

ينفذ لصالح : شؤون الطلاب

قائمة متابعة المراحل لعملية المتابعة التي حدثت بتاريخ Aug 31, 2024

الوضع الراهن : متأخر

وضع عملية المتابعة : مكتملة

الملاحظات : يوجد تأخر في العمل بس تأخر توريد المواد الأولية

#	اسم المرحلة	النسبة التنفيذية	نسبة التنفيذ السابقة	نسبة تنفيذ المتابعة	نسبة التنفيذ الكلية	طبيعة التنفيذ
1	مرحلة التخطيط	12	25	75	100	وفق الخطة
2	مرحلة التواصل	8	25	75	100	وفق الخطة
3	مرحلة التطوير	20	0	75	75	وفق الخطة
4	مرحلة التسليم	20	0	90	90	وفق الخطة
5	مرحلة الاختبار	40	25	75	100	وفق الخطة



الجهة المنفذة

قسم المعلومات

الجهة الطارحة

شؤون الطلاب

رئيس فريق العمل

حسن خضور

مدير المشروع

حسن خضور



بطاقة معلومات مشروع

تقرير خطة زمنية

تقرير انجاز

تقرير متابعة المراحل

بطاقة التعريف

طباعة



أتمتة أعمال مديرية التخطيط

بناء نظام معلوماتي لأتمتة أعمال مديرية التخطيط

تاريخ البدء Aug 26, 2024

تاريخ الانتهاء المتوقع May 29, 2024

معلومات حالة المشروع

المرحلة التطوير : مكتمل
نوع المشروع : مشروع تطوري
طبيعة المشروع : عادي

المعلومات التنفيذية

الجهة المنفذة: قسم المعلومات
رئيس فريق العمل: حسن خضور
مدير المشروع : حسن خضور
ينفذ لصالح : شؤون الطلاب

ذاتية المشروع

رمز المشروع: QWE-2
كتاب الطرح : 23434 / Aug 27, 2024
كتاب الموافقة : 234 / Aug 29, 2024
التمويل : ممول / المركز

قائمة متابعة العاملين لعملية المتابعة التي حدثت بتاريخ Aug 31, 2024

الوضع الراهن : وفق الخطة

وضع عملية المتابعة : مكتملة

الملاحظات : يسير العمل في المشروع وفق الخطة

الرقم الذاتي	الاسم والشهرة	ساعات العمل المسندة	ساعات العمل المنفذة	نسبة المساهمة	العمل المسند	العمل الملجز	التاريخ المتوقع لإنهاء الاعمال	الملاحظات
3	احمد الحسن	11	12	25	تصميم النظام وتخطيطه	كل الأعمال	Jan 10, 2024	يعمل وفق ما هو مسند إليه
1	حسن خضور	11	12	25	تطوير النظام	كل الأعمال	Jan 10, 2024	يعمل وفق ما هو مسند إليه
8	محمد احمد	13	19	50	كل الأعمال	كل الأعمال	Jan 1, 2025	يعمل وفق ما هو مسند إليه



الجهة المنفذة
قسم المعلومات

الجهة الطارحة
شؤون الطلاب

رئيس فريق العمل
حسن خضور

مدير المشروع
حسن خضور



ث. بطاقة تبديلات موظف

نظام إدارة حالة المشاريع

قائمة تبديلات

مشروع إدارة حالة المشاريع

قائمة تبديلات أعمال السيد حسن خضور

إن السيد حسن خضور يعمل حالياً في المشروع بسعة مهندس اختيار وينسبة تفرغ قدرها **18** ويبين الجدول التالي تبديلات عمله أثناء عمله في المشروع

#	تاريخ التبديل	سعة العمل قبل	سعة العمل بعد	نسبة التفرغ قبل	نسبة التفرغ بعد
1	Aug 26, 2024	مصمم واجهات	مطور	10	15
2	Aug 30, 2024	مهندس اختيار	مصمم واجهات	15	18

الصفحة الرئيسية

إدارة المتابعة

عمليات المتابعة

أنواع المشاريع

استعراض أنواع المشاريع

إضافة نوع جديد

الجهات الطارحة

استعراض الجهات الطارحة

إضافة جهة طارحة

إدارة المشاريع

طرح مشروع
 Go to Settings to activate Windows.
 استعراض المشاريع

بطاقة متابعات مشروع

تقرير خطة زمنية

طباعة



مشروع إدارة حالة المشاريع

بناء موقع وب لاصحة اجراءات إدارة مشاريع المعهد العالي

تاريخ البدء Aug 19, 2024

تاريخ الانتهاء المتوقع Sep 19, 2024

معلومات حالة المشروع	المعلومات التنفيذية	ذاتية المشروع
المرحلة التطويرية : يخطط	الجهة المنقذة: قسم النظم الإلكترونية	رمز المشروع: QWER
نوع المشروع : مشروع تطوري	رئيس فريق العمل: حسن خضور	كتاب الم طرح : Aug 19, 2024 / 12
طبيعة المشروع : تطوري	مدير المشروع : حسن خضور	كتاب الموافقة : Aug 19, 2024 / 112
	يلفأ لأصالح : شؤون الطلاب	التمويل : غير ممول / لا يوجد

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



الجهة المنقذة
قسم النظم الإلكترونية

الجهة الطارحة
شؤون الطلاب

رئيس فريق العمل
حسن خضور

مدير المشروع
حسن خضور



بطاقة معلومات مشروع

تقرير خطة زمنية

تقرير الجاز

طباعة



أتمتة أعمال مديرية التخطيط

بناء نظام معلوماتي لأتمتة أعمال مديرية التخطيط

تاريخ البدء Aug 26, 2024

تاريخ الانتهاء المتوقع May 29, 2024

معلومات حالة المشروع

المرحلة التطويرية : مكتمل

نوع المشروع : مشروع تطوري

طبيعة المشروع : عادي

المعلومات التنفيذية

الجهة المنفذة: قسم المعلومات

رئيس فريق العمل: حسن خضور

مدير المشروع : حسن خضور

ينفذ لصالح : شؤون الطلاب

ذاتية المشروع

رمز المشروع: QWE-2

كتاب الطرح : Aug 27, 2024 / 23434

كتاب الموافقة : Aug 29, 2024 / 234

التصويل : ممول / المركز

خطة الإنفاق السنوية

#	الحام	الضراء المحلي	الضراء الخارجي	نوع القطع
1	Jan 1, 2024	100000	1000	USA
2	Jan 1, 2025	110000	20100	USA
3	Jan 1, 2026	10000	20330	USA



الجهة المنفذة

قسم المعلومات

الجهة الطارحة

شؤون الطلاب

رئيس فريق العمل

حسن خضور

مدير المشروع

حسن خضور



خ. بطاقة إنجاز مشروع

بطاقة إنجاز مشروع

تقرير خطة زمنية

طباعة



أتمتة أعمال مديرية التخطيط

بناء نظام معلوماتي لأتمتة أعمال مديرية التخطيط

تاريخ البدء: Aug 26, 2024

تاريخ الانتهاء المتوقع: May 29, 2024

ذاتية المشروع

رمز المشروع: QWE-2

كتاب الطرح: Aug 27, 2024 / 23434

كتاب الموافقة: Aug 29, 2024 / 234

التمويل: ممول / المركز

معلومات حالة المشروع

المرحلة التطويرية: مكمل

نوع المشروع: مشروع تطويري

طبيعة المشروع: عادي

المعلومات التنفيذية

الجهة المنفذة: قسم المعلومات

رئيس فريق العمل: حسن خضور

مدير المشروع: حسن خضور

ينفذ لصالح: شؤون الطلاب

تقييم الجهة الطارحة: 96

ملاحظات الجهة الطارحة: تم استلام المشروع وفق المتطلبات المحددة

تاريخ الانتهاء: Feb 20, 2025

المساهمون في المشروع

#	الرقم الذاتي	اسم المشارك	الوظيفة	الهيئة	نسبة المساهمة
1	3	احمد الحسن	مهندس	مهندسين	28
2	1	حسن خضور	مبرمج	فني	42
3	8	محمد احمد	باحث رئيسي	باحث	30



الجهة المنفذة

قسم المعلومات

الجهة الطارحة

شؤون الطلاب

رئيس فريق العمل

حسن خضور

مدير المشروع


حسن خضور



4.10 بعض الواجهات في الموقع

أ. واجهة تسجيل الدخول

localhost:4200/login



نظام إدارة حالة المشاريع

Hasan@mail.hiast

تسجيل الدخول

2024 ©

Activate Windows
Go to Settings to activate Windows.

ب. واجهة متابعة مشروع

localhost:4200/tracks/detail/3

حسن خضور

Search

نظام إدارة حالة المشاريع

ملخص

متابعة المراحل

متابعة الموظفين

التقارير المرتبطة

وصف المشروع

بناء موقع ويب لمتابعة أعمال شؤون الطلاب

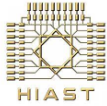
النوع: الراهن وفق الخطة

الملاحظات

يسير العمل في المشروع وفق الخطة

تاريخ عملية المتابعة: Aug 24, 2024

مكتملة



مشروع أتمتة أعمال شؤون الطلاب

رمز المشروع : DTDG

إستراتيجية متابعة مرحلة

إستراتيجية متابعة مشاريع

عمليات المتابعة

أنواع المشاريع

استعراض أنواع المشاريع

إضافة نوع جديد

الجهات الطارحة

استعراض الجهات الطارحة

إضافة جهة طارحة

إدارة المشاريع

طرح مشروع

استعراض المشاريع

المشاريع التي أديرها

Copyright HIAST. All Rights Reserved ©
Developed by Hasan

localhost:4200/byCriterion?listType=managed

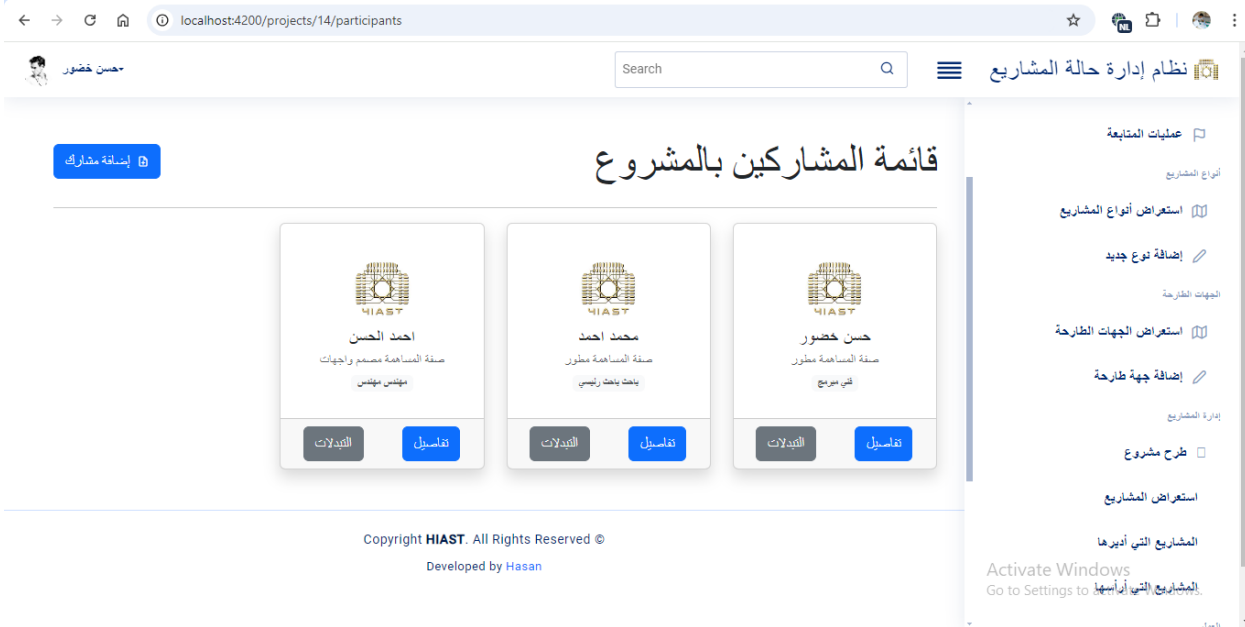
المعمل

Activate Windows
Go to Settings to activate Windows.

ت. الصفحة الرئيسية



ث. صفحة استعراض المشاركين



ج. صفحة إدارة مراحل المشروع

localhost:4200/projects/14/steps

حسن خضور

Search

نظام إدارة حالة المشاريع

إضافة مرحلة

قائمة مراحل المشروع

مرحلة التخطيط

في هذه المرحلة سيتم تخطيط المشروع

نسبة الانجاز 0 التكاليف 12 عدد العمال المطلوب 10

تاريخ البدء Aug 26, 2024 المدة 12

استعراض تاريخ مراحليها تنزيل التكاليف تنزيل المعلومات إزالة

مرحلة التواصل

يتم في هذه المرحلة التواصل مع الزبون لجمع متطلباته

نسبة الانجاز 0 التكاليف 8 عدد العمال المطلوب 10

تاريخ البدء Sep 1, 2024 المدة 2

عمليات المتابعة

أنواع المشاريع

استعراض أنواع المشاريع

إضافة نوع جديد

الجهات الطارحة

استعراض الجهات الطارحة

إضافة جهة طارحة

إدارة المشاريع

عرض مشروع

استعراض المشاريع

المشاريع التي أديرها

Activate Windows
Go to Settings to activate Windows.

المشاريع التي أديرها

الخاتمة

في هذا المشروع تم بناء نظام معلوماتي لتتبع حالة المشاريع في المعهد العالي يساعد مدراء المشاريع ورؤساء فرق العمل في إدارة المشاريع وتخطيطها وتقديم تقارير دورية، كما يمكن الوكيل العلمي من طرح المشاريع وتعريف أنواعها، ويسمح للموظفين بالاطلاع على مساهماتهم وساعات في المشاريع. من خلال هذا النظام أصبح بإمكاننا الاستغناء عن التعامل الورقي و مشاكله، كما أصبح بإمكاننا إصدار أنواع جديدة من التقارير والبيانات الإحصائية التي لم يكن بالإمكان الحصول عليها في ظل العمل الورقي والأنظمة القديمة.

تم تصميم هذا النظام باستخدام البنية المعمارية النظيفة والتصميم القائم على التقسيم لميزات وتم استخدام العديد من الأنماط التصميمية لتنجز النظام، الأمر الذي جعله مرنا وقابلا للتوسع والتعديل.

وفي رحلة تطوير هذا النظام، تمت التغلب على العديد من الصعوبات والتحديات بدءاً من فهم آلية العمل وكيفية سيره ووصولاً إلى التعامل مع أدوات ومفاهيم جديدة.

وفي سبيل تطوير هذا العمل، يمكن مكاملته مع نظام ذاتية الموظفين لتحقيق ربط فعلي لبياناتهم ومزامنة ساعات عملهم، كما يمكن إضافة إمكانية توليد بعض التقارير الإحصائية، كما يمكن بناء كتلة توصية في النظام تساعد في اقتراح العاملين الذين سيشاركون في المشروع الجديد بناءً على تاريخ مشاركتهم.

المراجع

- T. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley, 1991 [1]
[2]

