

فهم المخططات البيانية

زينب رستم

٣ أغسطس ٢٠٢٥

١ الدراسة مرجعية

ترافق تطور تقنيات التعلم العميق مع توسع كبير في مجال التعلم متعدد الوسائط، مما أسهم في تحقيق تقدم ملحوظ في توليد الوصف النصي للصور وتوفير إجابات دقيقة للأسئلة المتعلقة بها، ولكن تحورت أغلب الدراسات الحديثة حول الصور الطبيعية أو العامة لذلك كان لابد من المزيد من البحث للاستفادة من هذا النتاج في مجالات معرفية أكثر تخصصاً، مثل المجال الطبي أو مجال الاستشعار عن بعد وغيرها، بينما يبقى فهم وتحليل المخططات البيانية أحد المجالات الأكثر حداثة والأقل من حيث عدد الأبحاث حولها. سنستعرض في هذه الدراسة المرجعية بعض النماذج المبنية لتنفيذ مهام متعلقة بفهم وتحليل المخططات البيانية. أهم المعايير لتصنيف تقنيات فهم صور المخططات البيانية:

• المهام المتعلقة بفهم صور المخططات البيانية

١. التعرف على المخطط
٢. تصنيف المخطط (تحديد نوعه)
٣. استخلاص البيانات من المخطط
٤. توليد توصيف نصي للمخطط
٥. توليد إجابات لأسئلة حول المخطط
٦. توليد تلخيص لمحتوى المخطط
٧. تحويل مخطط لجدول بيانات

• أنواع المخططات البيانية

١. مخطط الأعمدة البياني

٢. الخط البياني

٣. مخطط الانتشار

٤. المخطط الدائري

٥. chart candle

٦. chart contour

• **ChartAdapter [6]:** اعتمدت بنية هذا النموذج على بنية encoder-decoder المعتمدة في الكثير من أطر العمل السابقة بمجال فهم الصور والمخططات البيانية، تمت إضافة جسر بين المرمز البصري visual encoder ونموذج اللغة الكبير textual decoder ، يتكون الدخل من صورة واستعلام نصي، يتم ترميز الصورة بعد إجراء المعالجة المسبقة، وهنا تبدأ المرحلة الخاصة بالتحول ChartAdapter التي تنقسم إلى أربع مكونات جزئية:

١. Cross-modal projector: يحاذي التضمينات البصرية والنصية.

٢. Latent textual embeddings: تستخلص الدلالات الضمنية للمخطط البياني.

٣. Cross-modal interaction layer: تدمج السمات البصرية والنصية عن طريق At-tention mechanism.

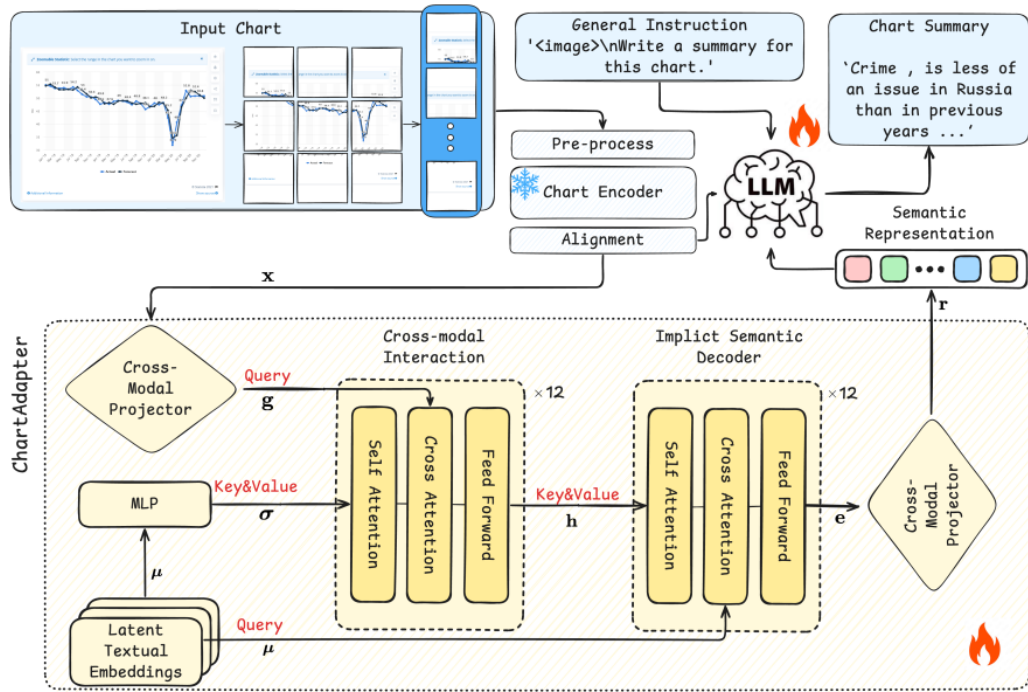
٤. Implicit semantic decoder: يحول الدلالات البصرية إلى صيغة تتوافق مع نموذج اللغة الكبير LLM.

ثم يتم تمرير كل السمات الناتجة إلى نموذج اللغة الكبير لتوليد ملخصات عالية الجودة. كان الهدف الأساسي من التصميم هو الدمج بين المرمز البصري ونموذج اللغة الكبير من خلال وحدة التحويل ChartAdapter، التي تعزز التعاون بين فهم المخططات وتوليد تلخيصات أو الإجابة على الأسئلة، انعكس هذا التكامل على نتائج النموذج التي تجاوز بها الكثير من النماذج المهمة السابقة.

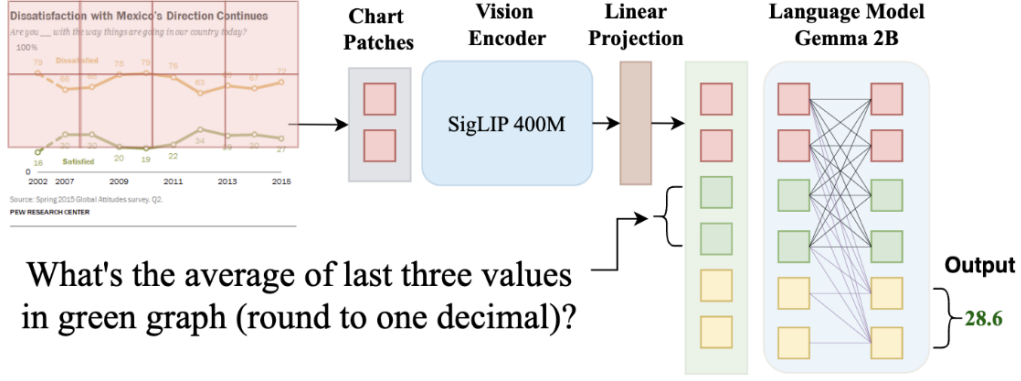
• **ChartGemma [1]:** هو نموذج خفيف الحجم مقارنة بأمثاله في مجال فهم المخططات البيانية (3 مليارات معامل)، له بنية encoder-decoder بحيث يشكل النموذج السابق PaliGemma [2] بنيته الأساسية حيث تم تدريبه على مجموعة بيانات كبيرة (10 مليارات زوج من الصور والنصوص)، تنقسم بنية ChartGemma إلى مكونين أساسيين:

١. المرمز البصري: يستخدم النموذج vision transformer (ViT) encoder.

شكل ١: بنية النموذج ChartAdapter [6]



شكل ٢: بنية النموذج ChartGemma [4]



٢. مفكك الترميز اللغوي (النصي): يُستخدم نموذج اللغة Gemma-2B المدرب مسبقاً على بيانات مكونة من وثائق نصية باللغة الانكليزية بالإضافة إلى نصوص برمجية ورياضياتية، مما يجعله مؤهلاً أكثر للتعامل مع المفاهيم المتعلقة بالخططات البيانية.

في الشكل ٢ لدينا نظرة عالية المستوى للبنية الخاصة بالنموذج ChartGemma، تتم معالجة صورة المخطط البياني وتحويلها على شكل دفعات إلى المرمز الذي يستخلص السمات البصرية من الصورة، بعد ذلك يتم نقل هذه السمات إلى فضاء التضمينات الخاص بنموذج اللغة الكبير، وتُلاحق التضمينات النصية للاستعلام المدخل بالتضمينات البصرية، ثم يقوم نموذج اللغة بتطبيق تقنية الانتباه الكلي على كل رموز tokens الدخل وبالنهاية يتم تطبيق mask على رموز الخرج.

مميزاته

أهم ما يميز هذا النموذج أن بيانات التدريب هي تعليمات مولدة بشكل مباشر من صور المخططات البيانية (visual instruction-tuning)، وبالتالي لا يعتمد على جداول البيانات أو أية تقنيات لتوليد هذه البيانات من المخطط، مما يمنح النموذج قابلية تعميم أفضل على أنواع الرسوم البيانية الواقعية، ويسمح له بالتقاط السمات المرئية العامة والتفاصيل الدقيقة بشكل أفضل.

أدأؤه

حقق النموذج ChartGemma نتائج متقدمة في 5 معايير تقييم شملت عدة مهام كتلخيص الرسوم البيانية، الإجابة على الأسئلة، وتقصي الحقائق، وقد تفوق على النماذج الأخرى في إنشاء ملخصات دقيقة وفهم المخططات المعقدة.

• **ChartAssistant [5]**: نموذج تم تخصيصه لفهم المخططات البيانية. يهدف النموذج للتعامل مع عدة مهام تتعلق بالمخططات البيانية، مثل التحويل من مخطط إلى جدول بيانات أو الإجابة على أشكال متعددة من الأسئلة (numerical QA – referring QA – open-ended QA) بالإضافة إلى توليد ملخص نصي عن المخططات، شاملاً بتلك المهام أنواع مختلفة من المخططات والرسوم البيانية.

أهم ما يميز هذا النموذج هو التدريب المسبق على تحويل المخططات إلى جداول بيانات نصية، وقد تم تدريبه على مجموعة البيانات ChartSFT الكبيرة والتي تضم أشكال متنوعة من المخططات. تفوق هذا النموذج على نماذج مهمة وحديثة مثل UniChart و ChartLlama لا سيما في اختبارات numerical QA، وذلك على بيانات حقيقية real-world charts بدون fine-tuning فقط اعتماداً على تدريبه المسبق.

- البنية: تم تطوير شكلين مختلفين من هذا النموذج:

١. ChartAst-D: نموذج رؤية-لغة صغير نسبياً مبني على أساس النموذج Donut (260M) (موسط)، يستخدم المرمز البصري Swin-Base بينما مفكك الترميز النصي BART، يعالج الصورة بدقة 224x224.

٢. ChartAst-S: نموذج رؤية-لغة كبير تم بناؤه على النموذج SPHINX (13B موسط)، يتم توظيف عدة رموز بصرية لاستخراج السمات من صورة الدخل، وخلاف ChartAst-D، يتولى نموذج اللغة الكبير عملية تحويل التضمينات النصية بعد دمجها مع البصرية إلى نص لغة طبيعية، وبفضل التصميم المركب للمرمز البصري بالإضافة إلى قوة نموذج اللغة الكبير بالاستنتاج فقد أظهر هذا النموذج نتائج جيدة جداً.

• **ChartLlama [2]**: بني على النموذج LLaVA-1.5 (مع بعض التعديلات) الذي يشكل بحد ذاته أساساً قوياً لمهمة فهم المخططات البيانية، وهو موجه لعدة مهام مثل فهم المخططات البيانية (Chart QA, Chart Summarization/Chart2Text, Data extraction) وتوليدها (Text2Chart, Chart-editing)، وبعض المهام الجديدة التي تشمل (Detailed chart descriptions, multi-chart reasoning, Chart2Chart) كما يمكنه التعامل مع عدد كبير من أنواع المخططات البيانية المتخصصة، وقد تفوق على عدة نماذج سابقة في المخططات البيانية المتخصصة (funnel, Gantt, heatmap).

• **TinyChart [7]**: هو نموذج لغة كبير متعدد الوسائط يملك 3 مليارات معامل، تم تطويره لفهم المخططات البيانية، يتبع طريقة التعلم Program-of-Thoughts بحيث يتدرب على توليد نصوص برمجية بلغة Python لتنفيذ العمليات الحسابية وبالتالي لتقليل الأخطاء في الحساب وتخفيف عبء الحسابات المعقدة، كما يطبق Visual Token Merging لسلاسل السمات الطويلة الناتجة عن الصور عالية الدقة مما يحسن من الفعالية الحسابية للنموذج. تتكون بنيته بشكل أساسي من Vision Transformer Encoder, Vision-Language Connector, Large Language Model (Phi-2). تفوق هذا النموذج على نماذج أخرى أكبر حجماً منه (up to 13B parameters) ونماذج أخرى غير مفتوحة المصدر مثل GPT-4V.

٢ مقارنة الأداء

فيما يلي جدول مقارنة لأداء النماذج في مهام فهم المخططات البيانية:

Table :١ Chart Understanding Model Performance Comparison

Model	Params	ChartQA			Chart-to-Text				OpenCQA	ChartFC
		Aug.	Hum.	Avg.	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-4	Accuracy
TinyChart@768 [7]	3B	93.86	73.34	83.60	17.18	--	--	--	20.39	--
TinyChart@512 [7]	3B	93.60	72.16	82.88	17.93	--	--	--	19.62	--
ChartAdapter [6]	3B	--	--	--	35.55	41.49	15.75	25.79	--	--
ChartLlama [2]	13B	90.36	48.96	69.66	27.47	--	--	--	--	--
ChartInstruct-Llama [3]	7B	82.40	40.64	61.52	26.1	--	--	--	14.78	64.99
ChartInstruct-Flan-T5-XL [3]	3B	93.84	50.16	72.00	25.01	--	--	--	14.81	72.65
ChartAst-D [5]	260M	91.3	45.3	68.3	27.1	--	--	--	14.9	--
ChartAst-S [5]	13B	93.9	65.9	79.9	28.1	--	--	--	--	--
ChartGemma [4]	3B	90.80	69.52	80.16	--	--	--	--	--	70.33

• الخط العريض يشير إلى أفضل أداء في كل عمود.

• -- " تشير إلى بيانات غير متوفرة في الجداول الأصلية.

الملخص

في عصر البيانات الذي نعيشه، تواجه مختلف الفئات تحديات كبيرة في تحليل الكم الهائل من المخططات والرسوم البيانية، حيث تعتمد الأساليب التقليدية على الخبرة الميدانية والجهد اليدوي المكلف، مما يؤدي غالباً إلى ضياع الرؤى أو سوء الفهم. هذه التحديات تؤثر بشكل مباشر على المحترفين الذين يحتاجون

لتحليلات سريعة، والباحثين الذين يتعاملون مع بيانات معقدة، والمعلمين والطلاب في رحلتهم التعليمية، بالإضافة إلى الصحفيين الذين يبنون قصصهم على البيانات، وصناع القرار الذين يعتمدون على دقة هذه التحليلات في تحديد مساراتهم الاستراتيجية.

ChartAnalyzer هي منصة متكاملة مدعومة بالذكاء الاصطناعي، تهدف لتطوير تفاعل المستخدمين مع البيانات المرئية وفهمها. من خلال الاستفادة من أحدث نماذج الذكاء الاصطناعي متعددة الوسائط، يحول النظام صور المخططات الثابتة إلى تجارب تحليل بيانات تفاعلية وقابلة للمحادثة، مما يجعل تفسير الرؤى المعقدة للبيانات في متناول المستخدمين من جميع المستويات.

يُجسد ChartAnalyzer الإمكانيات التحويلية للذكاء الاصطناعي في تحليل البيانات، حيث يجمع بين قدرات الذكاء الاصطناعي متعدد الوسائط المتطورة وهندسة البرمجيات المدروسة لإنشاء منصة لا تحل التحديات التحليلية الفورية فحسب، بل تفتح أيضاً إمكانيات جديدة لكيفية تفاعلنا مع تصورات البيانات والتعلم منها. مع استمرار تقدم الذكاء الاصطناعي في الفهم المرئي ومعالجة اللغة الطبيعية، ستصبح منصات مثل ChartAnalyzer أدوات أساسية بشكل متزايد لاتخاذ القرارات القائمة على البيانات في جميع قطاعات المجتمع.

أهداف المشروع

توفير منصة ويب تمكن المستخدمين من:

١. رفع صور المخططات والحصول على تحليل مدعوم بالذكاء الاصطناعي وإجابات على الأسئلة المتعلقة بتلك المخططات.
 ٢. المحادثة التفاعلية مع نماذج الذكاء الاصطناعي حول المخططات المرفوعة، مع مراعاة السياق.
 ٣. تتبع وحفظ سجل التحليلات والملاحظات لكل مستخدم.
- تقديم المنصة كتطبيق ويب حديث وسهل الاستخدام لتحليل تصورات البيانات.

متطلبات النظام

المتطلبات الوظيفية

١. المصادقة وإدارة المستخدمين
تسجيل المستخدمين، تسجيل الدخول، تسجيل الخروج، والتحقق من الجلسات (باستخدام JWT). فقط المستخدمون الموثوق بهم يمكنهم إجراء الدردشة.

٢. معالجة صور المخططات
رفع صور المخططات. تخزين واسترجاع صور المخططات المرتبطة بالمستخدمين.
٣. تحليل المخططات
تحليل صور المخططات المرفوعة باستخدام نماذج الذكاء الاصطناعي المتكاملة.
دعم التحليل العام والمتقدم (السياقي، أو باستخدام البرمجة الذهنية - thoughts of Program -)
تمكين المستخدمين من طرح أسئلة محددة حول مخططاتهم.
٤. الدردشة التفاعلية
واجهة محادثة فورية للمستخدمين للتفاعل مع الذكاء الاصطناعي حول مخططاتهم.
الحفاظ على سياق المحادثة لضمان استجابات ذكية وذات صلة.
تتبع وعرض سجل المحادثات والرسائل.
٥. سجل التحليلات والمحادثات
حفظ واسترجاع نتائج التحليل وسجل المحادثات لكل مستخدم.
توفير نقاط نهاية (endpoints) لعرض المحادثات وحذفها.
٦. إدارة النظام
التحقق من حالة الخادم والاتصال بقاعدة البيانات.
عرض نماذج الذكاء الاصطناعي المتاحة واختبار اتصالاتها.
٧. واجهة المستخدم الأمامية
واجهة مستجابة وحديثة تدعم جميع الميزات الرئيسية.
مؤشرات الكتابة، تواريخ الرسائل، ودعم الأجهزة المحمولة.
٨. واجهة البرمجة (API) والنظام الخلفي
نظام خلفي باستخدام FastAPI مع مسارات محددة بوضوح لجميع العمليات الرئيسية.
معالجة الأخطاء وتسجيلها لجميع نقاط النهاية.
المتطلبات غير الوظيفية

١. الأداء

أوقات استجابة سريعة للردشة والتحليل.
إدارة فعالة للحالة وتحسين عرض المكونات.
مؤشرات التحميل لإعلام المستخدم.

٢. القابلية للتوسع

بنية نظامية وواضحة (طبقات المجال، التطبيق، البنية التحتية).
قابلية التوسع لدعم نماذج وميزات جديدة وزيادة أحمال المستخدمين.
تحسين قاعدة البيانات وتصميم مخطط فعال.

٣. الأمان

مصادقة آمنة وحماية المسارات.
التحقق من المدخلات واتصالات API آمنة.
معالجة الأخطاء بشكل صحيح وتسجيل الأنشطة.

٤. سهولة الاستخدام

واجهة مستخدم حديثة ونظيفة وسهلة الوصول.
تنقل بديهي وأنماط تفاعل واضحة.
مراعاة إمكانية الوصول لجميع المستخدمين.

٥. القابلية للصيانة

هيكل تعليمات برمجية نظيف وفصل واضح للمسؤوليات.
وثائق شاملة (مثل مخططات UML، تدفقات النظام، إلخ).
مكونات قابلة للاختبار والتعديل.

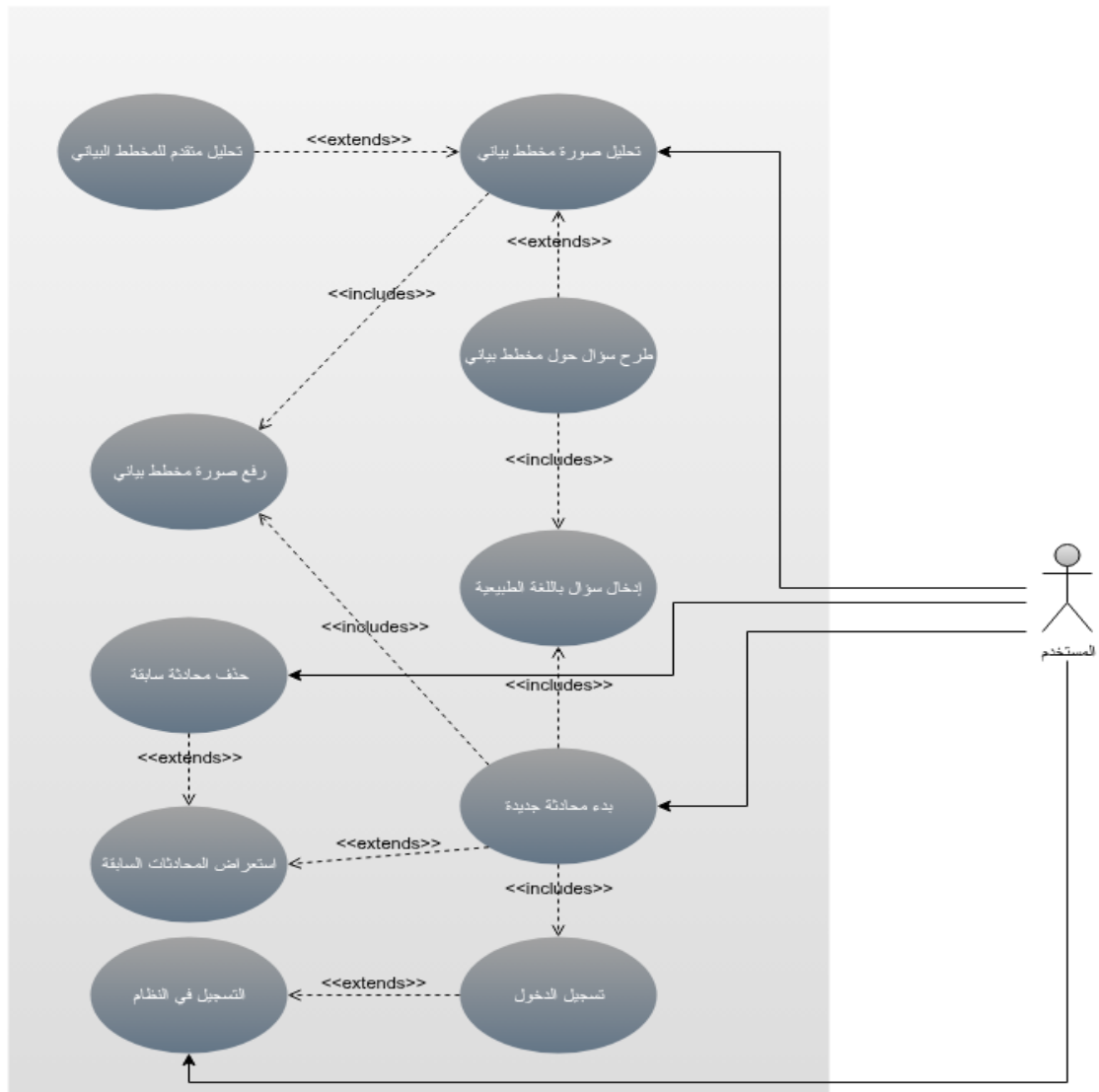
٦. التوافق

يعمل على جميع المتصفحات الحديثة والأجهزة المحمولة.

مخططات تحليل النظام

مخطط حالات الاستخدام

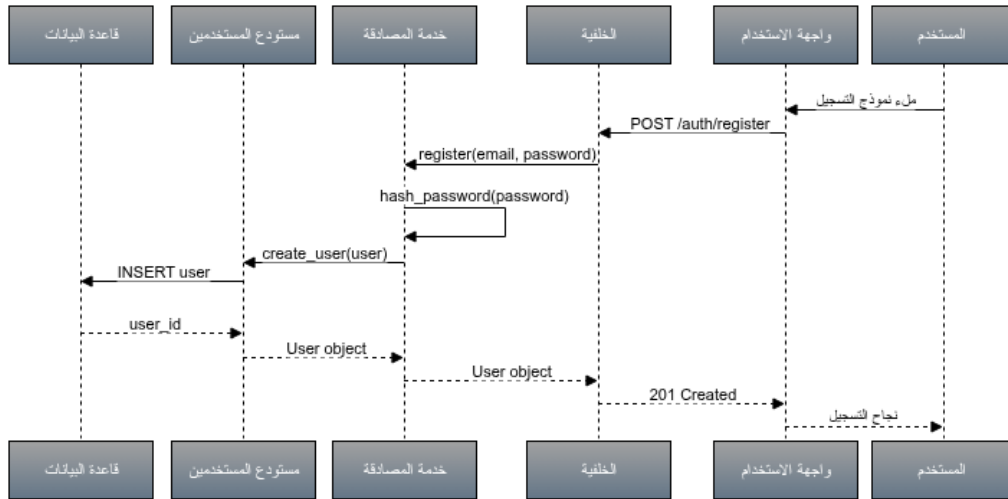
شكل ٣:
UsecaseDiagram



مخطط تسالي النظام

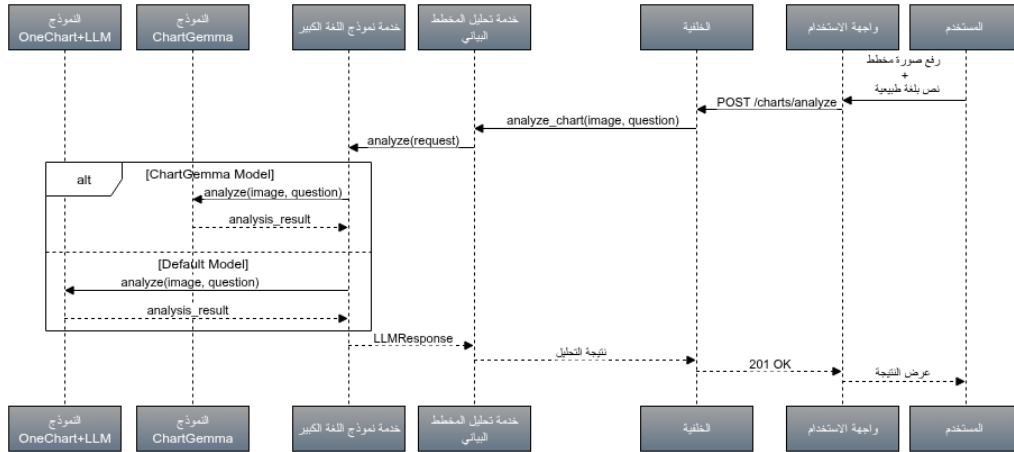
١. حالة تسجيل مستخدم جديد

شكل ٤:
SSD1



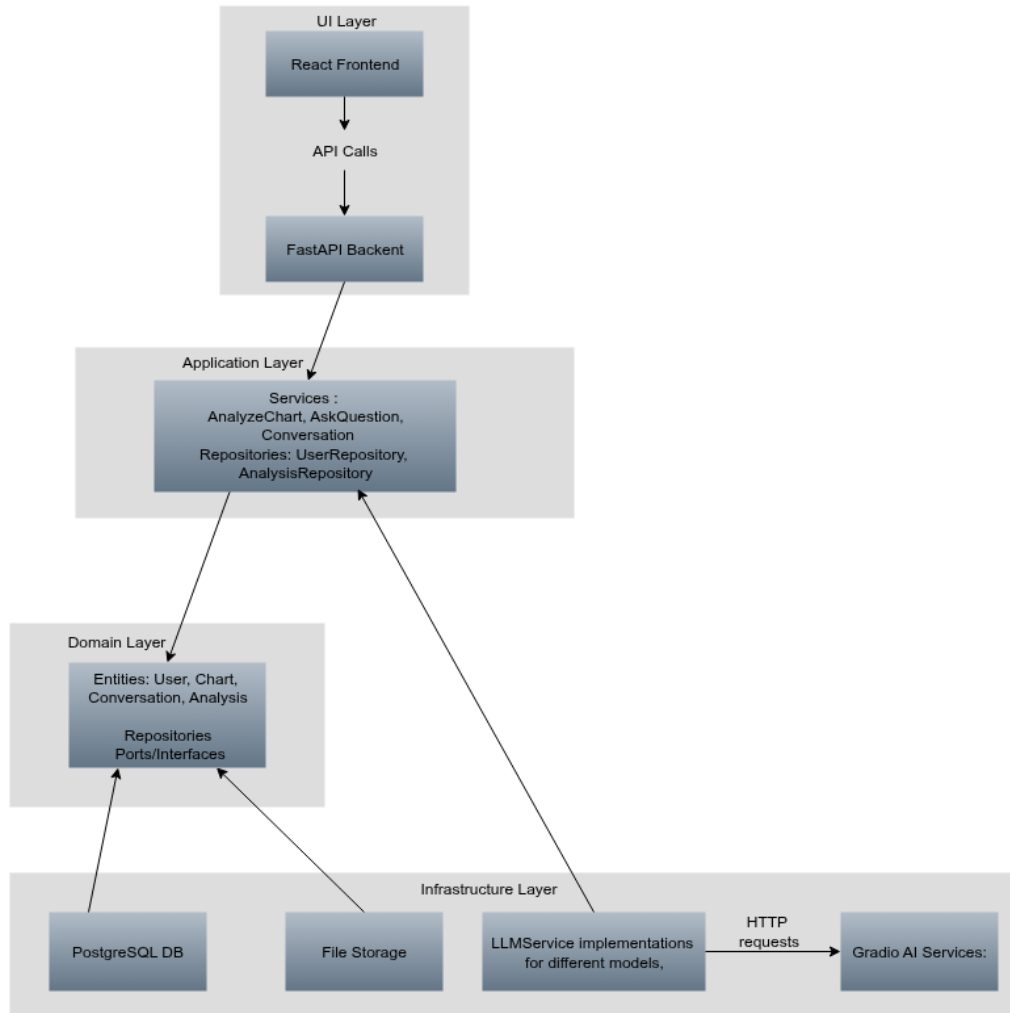
٢. حالة تحليل مخطط بياني

شكل ٥:
SSD2



مخطط طبقات ومكونات النظام

شكل ٦:
Layers



التنفيذ العملي

البنية المعمارية لنظام تطبيق الويب

تم بناء النظام وفق مبادئ الهندسة المعمارية النظيفة (Clean Architecture)، (لما تعطيه من أولوية

لبناء أنظمة مرنة وقابلة للاختبار والصيانة، من خلال فصل المهام وإدارة التبعية بفعالية كبيرة. يتم تحقيق ذلك عن طريق تنظيم الكود البرمجي وفق طبقات متحدة المركز، بحيث يكون منطق العمل المجرد في المركز، وتتوضع المهام الأخرى في طبقات خارجية. أهم مبادئ التصميم الجديرة بالذكر:

١. فصل الاهتمامات separation of concerns

يتم عزل أجزاء النظام التي تختلف في مجالها/اهتماماتها في طبقات منفصلة، مما يحد من أثر التغيير في مكون ضمن جزء واحد على مكونات الأجزاء الأخرى.

٢. قلب الاعتمادية dependency inversion

يتم توجيه التبعية/الاعتمادية نحو التجريد الذي يمثل منطق العمل، تعتمد الطبقات الخارجية على الداخلية التي بدورها مستقلة عن كل ما هو خارجها. تضمن هذه المنهجية عدم تأثر منطق العمل الأساسي بالمكونات الخارجية مثل واجهة الاستخدام أو قواعد البيانات.

٣. قابلية الاختبار Testability

بعد تحقيق الفصل الواضح بين الاهتمامات، يصبح من السهل اختبار وحدات المكونات على انفراد، وبالتالي اختبار كل وظيفة بدقة أكبر وبمعزل عن الوظائف الأخرى.

المراجع

- [١] Ali Mazraeh Farahani, Peyman Adibi, Mohammad Saeed Ehsani, Hans-Peter Hutter, and Alireza Darvishy. Automatic Chart Understanding: A Review. *IEEE Access*, 76221--11:76202, 2023.
- [٢] Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. Chartllama: A multimodal llm for chart understanding and generation. In *Proceedings of the Conference on Artificial Intelligence*, Singapore, 2023 Nanyang Technological University.
- [٣] Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. Chartinstruct: Instruction tuning for chart comprehension and reasoning. In *Proceedings of the Conference on Artificial Intelligence*, Canada, 2024 York University, Canada.
- [٤] Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. Chartgemma: Visual instruction-tuning for chart

reasoning in the wild. In *Proceedings of the Conference on Artificial Intelligence*, Canada, .2024 York University, Canada.

- [5] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. Chartassistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In *Proceedings of the Conference on Artificial Intelligence*, Shanghai, China, .2024 OpenGVLab, Shanghai AI Laboratory.
- [6] Peixin Xu, Yajuan Ding, and Wenqi Fan. Chartadapter: Large vision-language model for chart summarization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Hong Kong SAR, .2024 The Hong Kong Polytechnic University.
- [7] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. In *Proceedings of the Conference on Computer Vision*, Beijing, China, .2024 Renmin University of China.