

جامعة الشام الخاصة كلية الهندسة المعلوماتية

**إظهار نتائج اختبار الأداء للنظم المؤسساتية الداعمة للويب وخاصة الاكسير**

**Show the performance test results of the institutional systems supporting the web, especially Elixir**

إشراف

د. عمار جوخدار

تقديم الطالبتين

إيناس محمد سمير الصباغ رنيم مظهر دغوظ

14452021-

كلمة شكر

نتقدم بجزيل الشكر والامتنان إلى الدكتور عمار جوخدار لإشرافه على إنجاز هذا العمل ولمعلوماته القيّمة والدقيقة التي أفادنا بها. كما نتقدم بالشكر إلى الدكتور عميد كلية الهندسة المعلوماتية جورج كراز للتسهيلات التي قدمها لطلاب السنة الخامسة عموماً. ونشكر أيضاً كل من ساهم في إنجاز هذا المشروع.

# الخلاصة

تصميم وبناء تقارير باستخدام الرسومات البيانية لدعم القرار في النظم المؤسساتية الداعمة للويب وخاصة الاكسير بحيث نقوم بعرض نتائج مقاييس اختبار الأداء.

**Abstract**

Design and build reports using visualization to support decisions in corporate web-supporting systems, especially Elixir, so that we display the results of performance test measures.

المحتويات

[الخلاصة ii](#_Toc50056791)

[قائمة الأشكال v](#_Toc50056792)

[قائمة الجداول vi](#_Toc50056793)

[الاختصارات vii](#_Toc50056794)

[المقدّمة viii](#_Toc50056795)

[الفصل الأوّل متطلَّبات المشروع 1](#_Toc50056796)

[1.1-الهدف من المشروع 1](#_Toc50056797)

[2.1-المتطلبات الوظيفية 1](#_Toc50056798)

[3.1-المتطلبات غير الوظيفية 3](#_Toc50056799)

[1.3.1-متطلبات البيئة التقنية 3](#_Toc50056800)

[2.3.1-المتطلبات الأمنية 3](#_Toc50056801)

[3.3.1-متطلبات التخزين وإدارة الذاكرة 3](#_Toc50056802)

[الفصل الثاني الدراسة المرجعية 4](#_Toc50056803)

[2.1-مقدّمة 4](#_Toc50056804)

[2.2-لمحة عن تطبيقات الويب 4](#_Toc50056804)

[2..21-بيئة العمل Elixir 4](#_Toc50056804)

[2.3-لمحة عن الهواتف الذكيةSmart Phone 5](#_Toc50056805)

[23..1-نظام التشغيل (flutter)Android 7](#_Toc50056807)

[2.3.2- بنية نظام تشغيل Android 8](#_Toc50056808)

[4.2-خدمات الويب Web Services 9](#_Toc50056809)

[2.4.2-Application Programming Interface) API 9](#_Toc50056810)

[2.4.1 بنية المعطيات JSON 9](#_Toc50056811)

[الفصل الثالث إدارة المشروع 11](#_Toc50056812)

[1.3-مقدّمة 11](#_Toc50056813)

[2.3-خطة إدارة المشروع 11](#_Toc50056814)

[الفصل الرابع تحليل المتطلبات 12](#_Toc50056815)

[1.4-مقدّمة 12](#_Toc50056816)

[2.4-التطبيقات المشابهة 12](#_Toc50056817)

[3.4-مخطط حالات الاستخدام"Usecase Digram" 14](#_Toc50056818)

[4.4-توصيف حالات الاستخدام"Narrative Use Case" 13](#_Toc50056819)

[5.4-مخطط النشاط"Activity Digram" 24](#_Toc50056818)

[الفصل الخامس تصميم النظام 27](#_Toc50056820)

[1.5-مقدّمة 27](#_Toc50056821)

[2.5-التّصميم الأولي للنظام 27](#_Toc50056822)

[3.5-مكونات النظام 27](#_Toc50056824)

4.5-مخطط الصفوف"Class Diagram"…………………………………………. 28

[5.5- مخطط العلاقات والارتباطات"ERD" 29](#_Toc50056825)

[6.5- واجهات النظام 30](#_Toc50056825)

[الفصل السادس التنفيذ والاختبارات 31](#_Toc50056827)

[1.6- مقدمة 31](#_Toc50056828)

[2.6-أدوات التنفيذ 31](#_Toc50056829)

[3.6-الاختبارات 31](#_Toc50056830)

[4.6-دليل التنصيب 31](#_Toc50056831)

[5.6-دليل المستخدم 31](#_Toc50056832)

[6.6- مشاكل وصعوبات 47](#_Toc50056833)

[الخاتمة والآفاق المستقبلية 49](#_Toc50056834)

[المراجع 50](#_Toc50056835)

# قائمة الأشكال

[**الشكل 1: مقارنة بين بيئات التطوير لأظمة تشغيل الهواتف الذكية 3**](#_Toc493134758)

[**الشكل 2: مقارنة بين انتشار أنظمة التشغيل للهواتف الذكية خلال السنتين الماضيتين [2] 4**](#_Toc493134759)

[**الشكل 3:** واجهة تسجيل الدخول **3**](#_Toc493134758)

[**الشكل 4:** الواجهة الرئيسية للمواقع **3**](#_Toc493134758)

[**الشكل 5:** واجهة مشاهدة الإعلانات للأستاذ **3**](#_Toc493134758)

[**الشكل 6:** واجهة مشاهدة الإعلانات للموظف الإداري **3**](#_Toc493134758)

[**الشكل 7:** واجهة مشاهدة الإعلانات للطلاب **3**](#_Toc493134758)

[**الشكل 8:** واجهة إضافة إعلان **3**](#_Toc493134758)

[**الشكل 9:** واجهة عرض المواد عند الطالب أو الأستاذ **3**](#_Toc493134758)

[**الشكل 10:** واجهة عرض المواد عند الموظف الإداري **3**](#_Toc493134758)

[**الشكل 11:** واجهة عرض العلامات عند الطالب **3**](#_Toc493134758)

[**الشكل 12:** واجهة عرض العلامات عند الموظف الإداري **3**](#_Toc493134758)

[**الشكل 13:** واجهة إضافة العلامات للطلاب **3**](#_Toc493134758)

[**الشكل 14:** عرض جميع الطلاب والموظفين والأساتذة **3**](#_Toc493134758)

[**الشكل 15:** شاشة البدء **3**](#_Toc493134758)

[**الشكل 16:** الشاشة الرئيسية **3**](#_Toc493134758)

[**الشكل 17:** الشاشة الرئيسية للأستاذ **3**](#_Toc493134758)

[**الشكل 18:** شاشة عرض الإعلانات **3**](#_Toc493134758)

[**الشكل 19:** شاشة الفرز عن طريق التاريخ **3**](#_Toc493134758)

[**الشكل 20:** شاشة إضافة الإعلانات **3**](#_Toc493134758)

[**الشكل 21:** شاشة تأكيد عملية الإضافة **3**](#_Toc493134758)

[**الشكل 22:** شاشة عرض المواد **3**](#_Toc493134758)

[**الشكل 23:** شاشة تأكيد عملية المتابعة **3**](#_Toc493134758)

[**الشكل 24:** شاشة علامات الطلاب **3**](#_Toc493134758)

[**الشكل 25:** شاشة بحث عن طالب **3**](#_Toc493134758)

[**الشكل 26:** شاشة إعلانات الطالب **3**](#_Toc493134758)

[**الشكل 27:** شاشة بحث وفرز حسب اسم إعلان **3**](#_Toc493134758)

[**الشكل 28:** شاشة عرض العلامات **3**](#_Toc493134758)

[**الشكل 29:** شاشة البحث والفرز حسب اسم مادة **3**](#_Toc493134758)

[**الشكل 30:** شاشة المواد المتوافرة **3**](#_Toc493134758)

[**الشكل 31:** شاشة إضافة متابعة **3**](#_Toc493134758)

# قائمة الجداول

[**الجدول 1: بعض شركات الهواتف الذكيّة والأتظمة التي تبنّتها 6**](#_Toc493039866)

[**الجدول 2: المخطط الزمني لمراحل العمل 11**](#_Toc493039867)

[**الجدول 3: توصيف حالة تسجيل دخول 16**](#_Toc493039867)

[**الجدول 4: توصيف حالة عرض إعلانات 17**](#_Toc493039867)

[**الجدول 5: توصيف حالة عرض علامات 18**](#_Toc493039867)

[**الجدول 6: توصيف حالة عرض مواد 19**](#_Toc493039867)

[**الجدول 7: توصيف حالةإضافة إعلان 20**](#_Toc493039867)

[**الجدول 8: توصيف حالة إضافة علامة 21**](#_Toc493039867)

[**الجدول 9: توصيف حالة اهتمام بمادة 22**](#_Toc493039867)

[**الجدول 9: توصيف حالة إلغاء اهتمام بمادة 23**](#_Toc493039867)

# الاختصارات

JSON JavaScript Object Notation

ORM Object Relational Mapping

GC Garbage Collection

CPU Garbage Collection

HTTP Garbage Collection

# المقدّمة

إن إجراء اختبار الأداء يتم لتزويد stakeholders بمعلومات حول تطبيقهم فيما يتعلق بالسرعة والاستقرار وقابلية التوسع. والأهم من ذلك، أن اختبار الأداء يكشف ما يحتاج إلى تحسين قبل أن يذهب المنتج إلى السوق.

لذلك يجب أن نفكر في كيفية التخطيط والتنفيذ والإبلاغ عن جهود اختبار الأداء بطريقة هادفة. اسأل نفسك هذه الأسئلة الثلاثة

ابدأ بالذهاب بشكل أسرع، أبعد، إلى الأبد ... في وقت مبكر؟

لا ينبغي أن يكون اختبار الأداء هو المرحلة الأخيرة في التطوير. يجب أن نقوم بتوقع المشكلات مبكرًا والتخطيط للمستقبل، فإنك بذلك توفر على نفسك صداع إصلاح مشكلات الأداء التي تم الكشف عنها أثناء الاختبار. لهذا السبب سترغب في إشراك مهندس ضمان الجودة في مرحلة التخطيط للبناء أو بحالة إضافة الميزة الجديدة للتطبيق.

"إجراء اختبار الأداء فقط في ختام النظام أو الاختبار الوظيفي هو مثل إجراء فحص دم تشخيصي على مريض مات بالفعل.".

فيجب أن تكون قادرًا على فهم ما تم اختباره وما لم يتم اختباره وتقديم تقرير عن هذا العمل إلى stakeholders.

يحب بعض الناس جداول البيانات الوحشية. ومع ذلك، فإن معظمهم يواجهون مشكلة في مثل هذه الجداول بسبب بذل الجهد والوقت الكبيرين في قراءتهم وتحليلهم وبذلك يفضلون بيانات موجزة أو مرئية تجيب على بعض الأسئلة الرئيسية.

وفي محاولة حل هذه المشكلات، سنقوم في مشروعنا هذا بالاستفادة من الرسومات البيانية والجداول الإحصائية من خلال عرض نتائج اختبار الأداء بطريقة تسهل على المطور تحديد مكان الخطأ الذي قد يلحق النظام بأي خطر.

# الفصل الأوّل متطلَّبات المشروع

نعرض في هذا الفصل المتطلبات الوظيفية للمشروع أي الوظائف الأساسية التي يجب أن يحققها النظام كما نعرض المتطلبات غير الوظيفية من متطلبات الأداء والواجهات وغيرها.

**متطلّبات الأداء**

* الوثوقية: reliability يجب أن ينجز التطبيق الوظائف المطلوبة منه بدقة عالية. ويُفضّل ضمان عدم التعطّل خلال وقت تنفيذ عملية كاملة.
* زمن الاستجابة Response time: يجب أن تكون سرعة الأداء عالية وزمن الاستجابة قليل.
* سهولة الاستخدام usability: يجب أن يوفّر التطبيق واجهات واضحة بالنسبة للمستخدم.

## 1.1-الهدف من المشروع

تصميم وبناء تقارير باستخدام الرسومات البيانية لدعم القرار في النظم المؤسساتية الداعمة للويب وخاصة الاكسير بحيث نقوم بعرض نتائج مقاييس اختبار الأداء، ليتبين لدينا هل تحقق أن النظام قد نفذ مهام محددة بزمن محدد على عتاديات محددة بوجود عدد محدد من المستخدمين المتزامنين، يقومون بعدد محدد من العمليات.

## 2.1-المتطلبات الوظيفية

يجب أن يحقق النظام الوظائف التالية:

1. **اظهار الموارد:**

إظهار متوسطات الموارد بالنسبة لزمن تنفيذ سيناريو محدد

* 1. **إظهار الموارد بشكل تفصيلي:**

إظهار الموارد بشكل تفصيلي لكل متوسط بالنسبة إلى خدمة محددة.

## 3.1-المتطلبات غير الوظيفية

### 1.3.1-متطلبات **البيئة التقنية**

يعمل على جهاز كمبيوتر، لا يوجد ضرورة توافر الاتصال بشبكة الانترنت عند استخدام للتطبيق.

### 2.3.1-المتطلبات الأمنية والأداء

* مرن وسهل الاستخدام
* سرعة الأداء.
* سيكون الموقع متاحاً 24 ساعة.

### 3.3.1-متطلبات التخزين وإدارة الذاكرة

تطبيق يقوم بتخزين المعلومات على ذاكرة الهاتف المحمول وبالتالي يعرض المعلومات المخزنة في حالة عدم الاتصال بالإنترنت.

# الفصل الثاني الدراسة المرجعية

## **1-مقدّمة**

في هذا الفصل سنقوم بتعريف بيئة العمل Elixir، وسنعرض لمحة عن API وعن بنية المعطيات JSON. وبما أنَّ المشروع يتضمن اختبار أداء النظم المؤسساتية، سنقوم باستعراض بعض أهم مقاييس الأداء.

**2.2- لمحة عن تطبيقات الويب**

تخدم تطبيقات الويب عددًا كبيرًا من المستخدمين مع قدرة الأجهزة على خادم واحد مما يزيد من المشكلات التي يواجهها مطورو الويب كل يوم مثل إدارة الذاكرة، واستخدام قواعد بيانات مختلفة، وإدارة الوصول المتزامن، وأمن البيانات والأعمال، وقابلية الصيانة، وامتلاك شبكة ويب واجهة المستخدم التي تعمل عبر متصفحات الويب المتعددة (على سبيل المثال لا الحصر)، لهذا السبب يحتاج المطورون إلى بيئات عمل لتغطية كل هذا التعقيد بالنسبة لهم.

**2.2.1- بيئة العمل Elixir**

تقوم بيئات عمل مثل Spring و Laravel بعمل جيد في تغطية بعض هذه المشكلات ولكنك ستقاتل دائمًا على ثلاث جبهات (العرض والتحكم والتخزين)، بينما يستخدم Elixir من ناحية أخرى نموذج الهندسة المعمارية (MDA) مما يعني أنك تحتاج فقط إلى التركيز وفقًا لمنطق ونموذج سير العمل التجاري وسيهتم Elixir بالباقي، فأنت لست بحاجة إلى تعلم HTML أو كتابة أي بناء جملة لقاعدة البيانات أو حتى القلق بشأن معاملات قاعدة البيانات لأن Elixir يتيح لك التحكم في جميع المستويات الثلاثة من مكان واحد، ويولد المستخدم واجهة لك ويشكل استفساراتك.

الميزات الرئيسية للإكسير:

1. سهولة في البناء والتطوير من خلال مبدأ run then specify.

.2 إنشاء تلقائي لواجهة المستخدم وقاعدة البيانات وإجراءات العمل.

3. إدارة آلية للموارد من ذاكرة وCBU.

4. نموذج أمان متين وسهل للمستخدم.

## 4.2**-خدمات الويب Web Services**

خدمة الويب بشكل أساسي هي برامج مكتوبة بلغات مثل java، asp، php تطلُب برامج تعمل على مخدّم آخر (خدمة الويب) واستخدام استجابة ذلك البرنامج في تطبيق وِب أو أي تطبيق آخر.

تتميز خدمة الويب عن تطبيق الويب بشكل أساسي بطريقة طلب الخدمة، حيث يتم طلب الخدمة عبر واجهتها البرمجية   
API وليس الرسومية GUI (Graphical User Interface)، ويتم إعادة نتيجة الطلب على هيئة XML أو JSON.

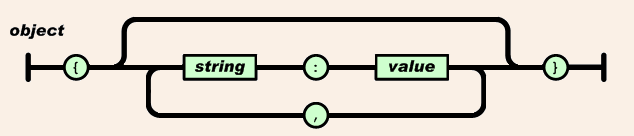
### 1.4.2**-Application Programming Interface) API)**

API هي اختصار ل Application Programming Interface، وهي مجموعة **صفوف** **وتوابع** مكتوبة بلغة برمجة معينة، توفر ال API مجموعة من **الواجهات** المجرّدة التي تخفي خلفها تعقيدات النص البرمجي بحيث يمكن للبرامج الأخرى (التي من الممكن أن تكون مكتوبة بلغات برمجة مختلفة) أن تتفاعل مع هذه الواجهات بكل سهولة، يتم تنظيم هذه الصّفوف والتوابع في مكتبة برمجيّات أو خدمة وِب، تصاحبها شروحات تمكن المبرمج من استيعاب طريقة استعمال تِلك الخدمات أو محتويات المكتبة.

### 2.4.2- بنية المعطيات JSON

(JSON) Java Script Object Notation هي بنية معطيات سلسة لنقل البيانات، سهلة القراءة والاستيعاب من قبل المبرمجين بشكل عام، وسهلة التوليد والتحليل للأجهزة والحواسيب، وهي ظهرت بدايةً للاستخدام مع لغة جافا سكريبت (Java Script)، ولكنها مستقلة تماماً عن اللغة، حيث يُمكن استخدامها من قِبل عائلة لغات ال C والتي تشمل ++C، C#، Java، والعديد من اللغات الأُخرى لاحتواء اللغات آنفة الذكر على مكتبات خاصّة بهذه البنية، تلك المميّزات تجعل من JSON الأفضل في نقل البيانات data)) من لغة لأُخرى.

تتكوّن بنية JSON من مجموعة من الأزواج: الأسماء ((Names والقيم Values)) حيث يمكن للقيمة أن تكون غرض (Object) من النمط JSON أيضاً، ويمكنها أن تكون مصفوفة من أغراض JSON.



الشكل 5: بنية JSON

مثال على بنية JSON:

{"**menu**": {

"**header**": "SVG Viewer",

"**items**": [

{"**id":** "Open"},

{"**id**": "Open New", "label": "Open New"},

null,

{"**id**": "Zooming", "label": "Zoom In"},

{"**id**": "Zoom Out", "label": "Zoom Out"},

{"**id**": "Original View", "label": "Original View"},

null,

{"**id**": "Quality"},

{"**id**": "Pause"}

]

}}

**اختبار أداء:**

اختبار الأداء هو عملية اختبار برمجية تُستخدم لاختبار السرعة ووقت الاستجابة والاستقرار والموثوقية وقابلية التوسع واستخدام الموارد لتطبيق برمجي تحت عبء عمل معين. الغرض الرئيسي من اختبار الأداء هو تحديد وإزالة الاختناقات في الأداء في تطبيق البرنامج.

يركز اختبار الأداء على فحص البرنامج من حيث:

1. السرعة - تحدد ما إذا كان التطبيق يستجيب بسرعة
2. قابلية التوسع - تحدد الحد الأقصى لتحميل المستخدم الذي يمكن لتطبيق البرنامج التعامل معه.
3. الاستقرار - يحدد ما إذا كان التطبيق مستقرًا في ظل أحمال متفاوتة

**أنواع اختبار الأداء:**

1. Load Testing: عبء متوقع (لتحديد الاختناقات)
2. Stress Testing: عبء عمل شاق (لتحديد نقطة انهيار البرنامج)
3. Endurance Testing: التعامل مع الحمل على فترة طويلة (التحمل)
4. Spike Testing: يختبر رد فعل البرنامج على الارتفاعات الكبيرة في الحمل
5. Volume Testing: التحقق من الأداء في احجام قاعدة بيانات مختلفة
6. Scalability Testing: تحديد قابلية التوسع لدعم زيادة حمل المستخدم

يتم تصنيف مقاييس اختبار الأداء بالشكل التالي:

1. المقاييس التي تؤثر على السرعة:
2. Response time
3. Hits per second
4. Throughput
5. Garbage collection
6. المقاييس التي تؤثر على قابية التوسيع:  
   1- Free physical memory  
   2- Total Size Hard Disk  
   3- CPU Time  
   4- Total swap space
7. المقاييس التي تؤثر على الاستقرار:  
   1- Committed memory   
   2- Thread counts3- Error rate

**Response time**

**تعريفه:**

الوقت الذي يستغرقه النظام للوصول الى مدخلات محددة حتى تنتهي العملية. على سبيل المثال لديك واجهة API وتريد ان تعرف بالضبط مقدار الوقت المستغرق لتنفيذه وارجاع البيانات في JSON.

**أهميته ودلالته بالنسبة إلى:**

1. المستخدم:

|  |  |
| --- | --- |
| Significance | Response time |
| أفضل وقت استجابة. إذا كان وقت الاستجابة 0.1، يشعر المستخدمون دائمًا أن التطبيق أو النظام يستجيب على الفور، ولا يشعرون بأي انقطاع. | **0.1 ثانية** |
| هذا هو الحد الأقصى لوقت الاستجابة المقبول. من غير المحتمل أن يشعر المستخدمون بأي مقاطعة، على الرغم من أنهم قد يواجهون بعض التأخير. قد يؤدي وقت الاستجابة الذي يزيد عن ثانية واحدة إلى مقاطعة تجربة المستخدم | **1.0 ثانية** |
| إنه حد أقصى يتجاوز وقت الاستجابة بعده الحد المقبول. ومع ذلك، في وقت اليوم، إذا تجاوز وقت الاستجابة 6 ثوانٍ، فسيترك المستخدم هذا الموقع أو يغادر التطبيق. | **0 1ثانية** |

بشكل عام، يجب أن يكون وقت الاستجابة سريعًا قدر الإمكان في فترة 0.1 - 1 ثانية. ومع ذلك، يمكن للأشخاص التكيف مع أوقات استجابة أبطأ، لكنهم لن يكونوا سعداء أبدًا بوقت استجابة أكبر من ثانيتين. كلما قلت أوقات الاستجابة، كان رضا المستخدم أفضل، وتكاليف أقل، ورضا أكبر للمستخدمين.

1. المطور:

يساعدنا وقت الاستجابة القصوى في معرفة المكونات التي من المحتمل أن تكون إشكالية.

**طرائق معالجة البطء في وقت الاستجابة:**

* اختر مزود استضافة ويب موثوق به

إذا كنت تعاني من بطء الأداء، فقد حان الوقت لترقية حزمة الاستضافة أو تغيير مقدمي الخدمة. حاول الابتعاد عن أرخص مضيفي الويب لأن الأداء الذي ستختبره سيعكس السعر الذي تدفعه. في حالة استخدامك للاستضافة المشتركة (بمعنى أنك تشارك موارد الخادم مع مواقع الويب الأخرى)، فقد يكون الوقت قد حان للترقية إلى خادم VPS أو خادم مخصص وفقًا لمتطلباتك

* تحسين خادم الويب الخاص بك

قد لا يشعر الجميع بالراحة عند القيام بتحسين خادم الويب الخاص بهم، ومع ذلك، فإن ضمان تحسين خادمك بالكامل سيساعد في تقليل وقت استجابة الخادم لديك. عندما يتعلق الأمر باختيار برنامج الخادم، هناك عدد قليل من الخيارات المتاحة؛ على الرغم من أن معظم المستخدمين يختارون استخدام Apache أو Nginx. هناك إيجابيات وسلبيات لكل خادم ويب على الرغم من أن Nginx معروف بتقديم أداء أفضل في العديد من الظروف. تعرف على المزيد حول الفرق بين Nginx وApache..

* التخزين المؤقت

يساعد التخزين المؤقت بشكل كبير في تحسين وقت استجابة الخادم. **طريقة حسابه:**

(Request end time – Request start time)

**Hits per second**

**تعريفه :**

عدد الزيارات في الثانية يمثل عدد الطلبات المرسلة إلى الخادم (الحمل الذي يتم ضرب الخادم).

عدد الزيارات في الثانية هو إجمالي الحمل الذي تم تعيينه بواسطة المستخدمين الظاهريين المتزامنين على الخادم، بغض النظر عما إذا تم تنفيذها بنجاح أم لا على جانب الخادم.

**Throughput**

**تعريفه:**

عدد المعاملات بالثانية التي يمكن للتطبيق معالجتها وكمية لمعاملات التي يتم انتاجها بمرور الوقت اثناء الاختبار. الإنتاجية هي أحد المقاييس الرئيسية في اختبار الأداء، يتم استخدامه للتحقق من عدد الطلبات التي يمكن للبرنامج معالجتها في الثانية او الدقيقة او الساعة.

**أهميته بالنسبة إلى:**

1. المستخدم:

كلما انخفض معدل النقل، كان أداء الشبكة أسوأ. تعتمد الأجهزة على تسليم حزم ناجح للتواصل مع بعضها البعض، لذلك إذا لم تصل الحزم إلى وجهتها، فستكون النتيجة النهائية هي جودة الخدمة الرديئة. في سياق مكالمة VoIP، قد يتسبب معدل النقل المنخفض في حصول المتصلين على مكالمة ذات جودة رديئة مع تخطي الصوت.

1. مدير الشبكة:

يزداد وقت الاستجابة سوءًا إذا تم تنزيل عدة أشخاص في وقت واحد. إذا كنت مديرًا لتكنولوجيا المعلومات، فإن النظر إلى استخدام نقطة النهاية يمكن أن يخبرك بالموظفين الذين يتسببون في زمن انتقال للتطبيقات غير المتعلقة بالعمل. حتى إذا لم تكن مشرفًا وتنظر إلى هذا من وجهة نظر الإنتاجية، فمن المفيد أيضًا معرفة التطبيقات التي تعمل على حل المشكلة. في كلتا الحالتين، تؤدي المعلومات إلى العمل. يعد استخدام معدل النقل لقياس أداء الشبكة مفيدًا عند استكشاف الأخطاء وإصلاحها لأنه يساعد المسؤولين على تحديد السبب الجذري للشبكة البطيئة. ومع ذلك، فهو أحد العوامل الثلاثة التي تحدد أداء الشبكة. والاثنان الآخران هما زمن الوصول وفقدان الحزمة.

**طرق تحسين الإنتاجية:**

إلى حد بعيد، فإن أهم شيء يجب القيام به عند تحسين الإنتاجية هو تقليل زمن انتقال الشبكة. يؤدي وقت الاستجابة إلى إبطاء سرعة النقل مما يؤدي بدوره إلى تقليل الإنتاجية وتوفير أداء شبكة ضعيف للمستخدمين. بشكل عام، تريد تقليل التأخير عن طريق مراقبة استخدام نقطة النهاية ومعالجة اختناقات الشبكة.

* استخدام اتصال سلكي - يمكن أن "تفقد" الاتصالات اللاسلكية لأنها تُرسل عبر الهواء. عندما يحدث هذا، يجب على الخوادم إرسال المعلومات من جديد، مما يتسبب في تأخير. يمكنك تقوية الإشارة اللاسلكية الخاصة بك، ولكن لا يزال من المحتم حدوث زمن الانتقال لأن جميع الإشارات اللاسلكية بها هذه المشكلة بدرجات متفاوتة. يعد استخدام كبل Ethernet طريقة رخيصة وسهلة لتحسين اتصالك. يعد التحديث إلى شبكة الألياف الضوئية أحد الخيارات أيضًا.
* إعادة تشغيل شبكتك - ألم تقم بإيقاف تشغيل أجهزة الشبكة منذ فترة؟ قد يتسبب في تأخر. افصل جهاز التوجيه والمودم وانتظر بضع لحظات وأعد التشغيل.
* أغلق التطبيقات التي تستخدم قدرًا كبيرًا من النطاق الترددي - تتمتع جميع اتصالات الشبكة بنطاق ترددي محدود، وإذا كنت تستخدم أكثر من مشاركتك العادلة، فسيزيد وقت الاستجابة. خفف من استخدام تلك التطبيقات كثيفة النطاق الترددي.
* تعطيل جدران الحماية الخاصة بك - قد يبدو هذا اقتراحًا مجنونًا في البداية، ولكن دعني أوضح… تقوم جدران الحماية بتصفية جميع حركة مرور الشبكة الواردة والصادرة، ويمكن لجدار الحماية التالف أن يبطئ الشبكة. سيحدث نفس الشيء إذا كنت تقوم بتشغيل عدة برامج جدار حماية في نفس الوقت لأن كلاهما يضع ضغطًا هائلاً على شبكتك. يمكن أن يساعدك تعطيل جدار الحماية في معرفة ما إذا كان هذا عاملًا مهمًا في تباطؤك الحالي.
* تجاوز أجهزة الشبكة المعيبة - يحدث التأخر أحيانًا بسبب الأجهزة المعيبة. يمكنك محاولة العمل بدون معدات معينة لترى كيف يؤثر ذلك على سرعة الشبكة. ستكشف عملية الحذف ما هي أجهزة الشبكة، إن وجدت، المسؤولة عن زمن الوصول.
* استدعاء التعزيزات - بعد إجراء اختبار السرعة والتحقق من فقدان الحزمة، قد يكون الوقت قد حان للاتصال بمزود خدمة الإنترنت. يمكن أن تكون المشكلة في نهايتهم.

**طريقة حسابه:**

(Number of transaction / Number of seconds)

**Garbage collection**

**تعريفه:**

يساعد مقياس جمع البيانات المهملة في جعلك على دراية بمشاكل الأداء المختلفة.   
يوفر معلومات حول حجم الذاكرة غير المستخدمة التي يتم إرجاعها إلى النظام. تساعد مراقبة جمع القمامة في تحسين الأداء والكفاءة.

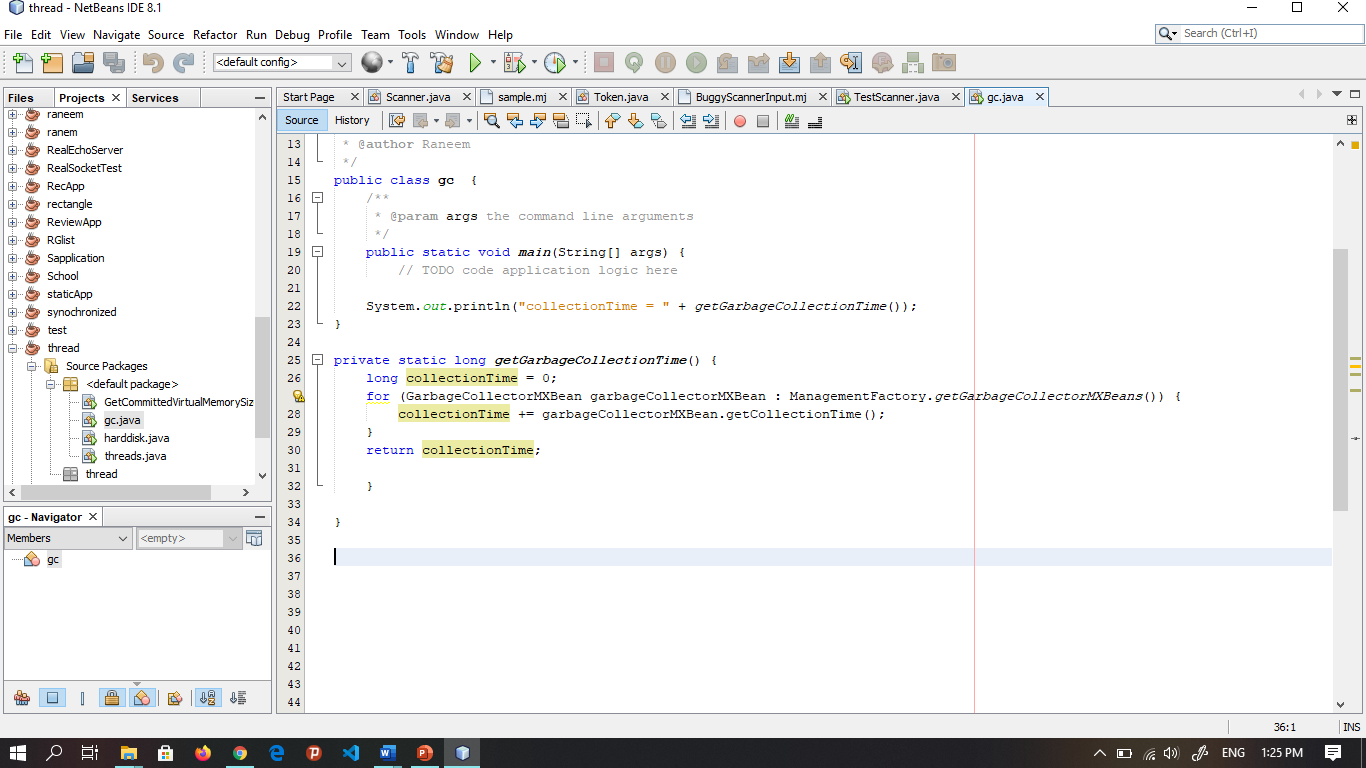
**أهميته:**

يعد جمع البيانات المهملة أمرًا شائعًا لأنه يعزز التطوير السريع للتطبيقات. من ناحية أخرى، كلما أظهر النظام أداء ضعيفًا بشكل غير متوقع، يظهر جامع القمامة دائمًا كمشتبه به محتمل.

بشكل عام، هناك متغيرين يؤثران على النفقات العامة لجهاز جمع القمامة الحديث والدقيق والمتعدد الأجيال: مدة الإيقاف المؤقت وتكرار التجميع.

جمع البيانات المهملة يرتبط بإدارة الذاكرة ويساعد على تحسين أداء التطبيق. يجب ألا يكون جمع البيانات المهملة في النظام سريعًا جدًا أو متأخرًا جدًا. تؤدي العديد من دورات GC إلى تدهور أداء النظام وتسبب ارتفاعًا حادًا في وحدة المعالجة المركزية بينما يؤدي التأخير في دورات GC إلى تسرب الذاكر.

**طريقة حسابه:**



**Free physical memory**

**تعريفه:**

مقدار الذاكرة الفعلية المتاحة للاستخدام.

**أهميته:**

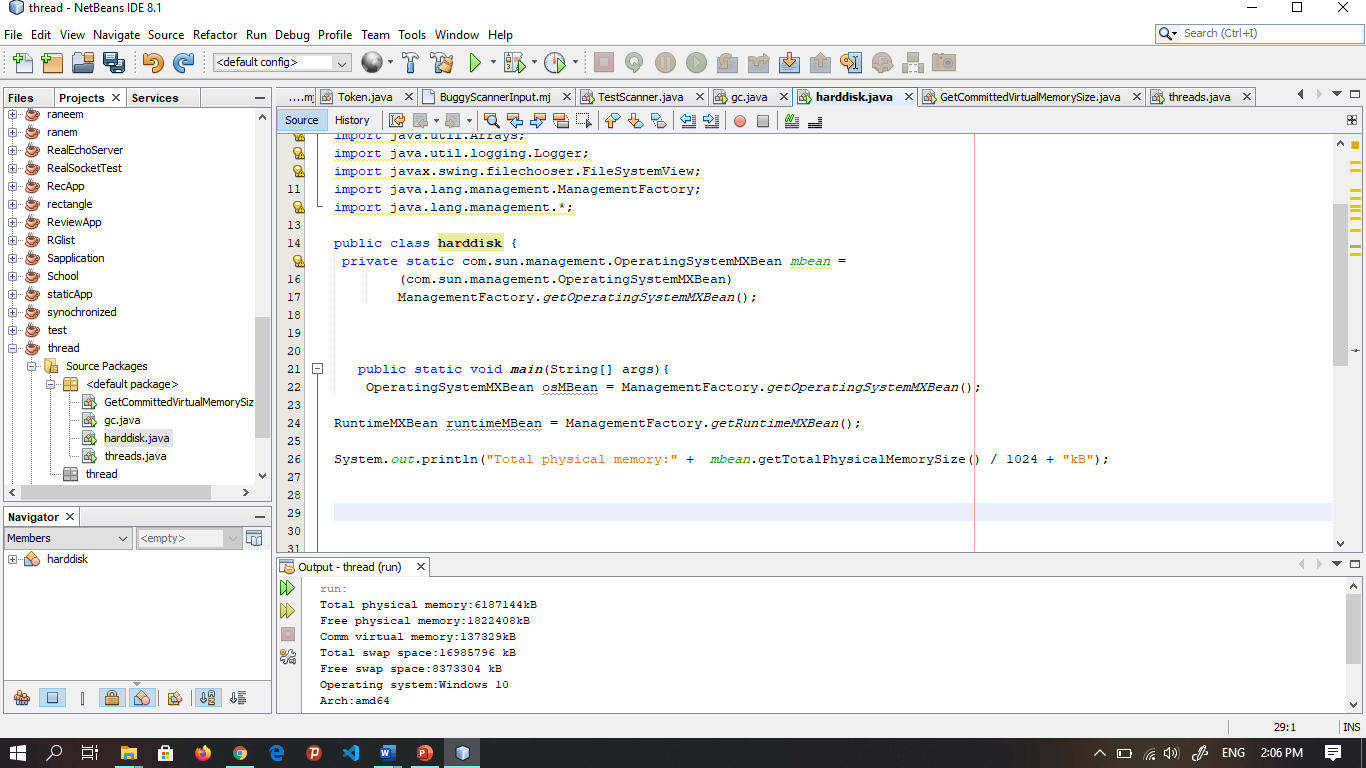
تعد أهمية استخدام الذاكرة في اختبار الأداء أمرًا بالغ الأهمية لفهم مقدار الاستخدام بناءً على حالة التحميل لعدد معين من المستخدمين في النظام. يمنحني هذا القدرة على معرفة ما إذا كان حجم أجهزتي مناسبًا ولا توجد ندرة في الذاكرة على النظام. يساعدني استخدام الذاكرة أيضًا في فهم ما إذا كان هناك أي تسرب بسبب سوء أداء الكود، وأي اتصالات مفتوحة ولم يتم إغلاقها بشكل صحيح، وكائنات مفتوحة، وسطر رموز زائدة عن الحاجة، كل هذه العوامل تحمل الذاكرة ولن يتم مسحها إلا عند إعادة تشغيل الخدمات للإفراج ذاكرة. ومن ثم فإن الذاكرة هي عداد أداء مهم لتحديد صحة وقدرة النظام العام.

**طريقة تحسين الذاكرة الفيزيائية :**

يعد وجود مساحة كافية متاحة أمرًا ضروريًا للغاية لتعمل الذاكرة الظاهرية وذاكرة الوصول العشوائي بشكل صحيح. يمكن تحسين أداء الذاكرة الظاهرية تلقائيًا عن طريق تحرير الموارد في التسجيل

تعد ترقية ذاكرة الوصول العشوائي الخاصة بك هي الطريقة الوحيدة التي يمكنك من خلالها زيادة الذاكرة الافتراضية الخاصة بك، عن طريق زيادة حجم الذاكرة الإجمالية المتاحة للنظام. في ذلك، سوف تخفف من مشكلة الذاكرة الافتراضية أثناء العملية ويمكن أن ترى زيادة في سرعة نظامك أيضًا

**طريقة حسابه:**

****

**Total Size Hard Disk**

**تعريفه:**

الحد الأقصى للبيانات التي يمكن للقرص أو محرك الأقراص الاحتفاظ بها.

**أهميته:**

توافر مساحة القرص في الخوادم أمر بالغ الأهمية. تقوم التطبيقات التي تعمل على هذه الخوادم بتخزين ملفات السجل وكتابة البيانات إلى قاعدة البيانات، والتي يتم تثبيتها أيضًا على نفس الخادم؛ إذا لم تتوفر مساحة كافية على القرص ، فلن يعمل التطبيق بشكل صحيح ، وقد يتعطل. تعد مراقبة مساحة القرص أمرًا ضروريًا لمسؤولي تكنولوجيا المعلومات لضمان الأداء العالي للخوادم، والحفاظ على وقت تشغيل الشبكة بالشكل الأمثل عن طريق تجنب النقص المفاجئ وغير المتوقع في مساحة قرص الخادم

القياس يعطي معلومات عن المساحة الخالية على القرص والذاكرة المستخدمة أثناء تنفيذ السيناريو. توفر مساحة القرص معلومات عن مساحة القرص المستخدمة

**CPU Time**

**تعريفه:**

مقدار الوقت المحدد الذي قضته وحدة المعالجة المركزية لمعالجة البيانات أو العمليات.

**أهميته:**

**Total swap space**

**تعريفه:**

مساحة التبديل هي مساحة على القرص الثابت تكون بديلاً عن الذاكرة الفعلية. يتم استخدامه كذاكرة افتراضية تحتوي على صور ذاكرة العملية. عندما ينقص جهاز الكمبيوتر الخاص بنا من الذاكرة الفعلية،

فإنه يستخدم ذاكرته الافتراضية ويخزن المعلومات في الذاكرة على القرص فإنه يستخدم ذاكرته الافتراضية ويخزن المعلومات في الذاكرة على القرص.

**أهميته:**

تساعد مساحة التبديل نظام تشغيل الكمبيوتر في التظاهر بأن لديه ذاكرة وصول عشوائي (RAM) أكبر مما هو موجود بالفعل. ويسمى أيضًا ملف المبادلة. يُطلق على تبادل البيانات بين الذاكرة الظاهرية والذاكرة الحقيقية اسم التبادل والمساحة على القرص باسم "مساحة التبديل"، يساعد وجود ملفات مبادلة كافية النظام في الحفاظ على مساحة خالية من الذاكرة الفعلية طوال الوقت يمكن استخدام المساحة الموجودة في الذاكرة الفعلية التي تم تحريرها بسبب مساحة التبادل بواسطة نظام التشغيل لبعض المهام المهمة الأخرى

إذا لم يكن لديك ذاكرة كافية، فسيتم استخدام المبادلة كثيرًا وبشكل ملحوظ خلال أي طفرات في متطلبات الذاكرة. إذا لم يكن لديك ذاكرة كافية ولا توجد مساحة للتبديل، فسيؤدي ذلك غالبًا إلى فشل تخصيص الذاكرة للطلبات التي تحتاج إلى المزيد من صفحات الذاكرة.

**طريقة حسابه:**

**Committed memory**

**تعريفه:**

هي مزيج من ذاكرة الوصول العشوائي ومساحة القرص التي يمكن أن تستخدمها العمليات الجارية.

**أهميته:**

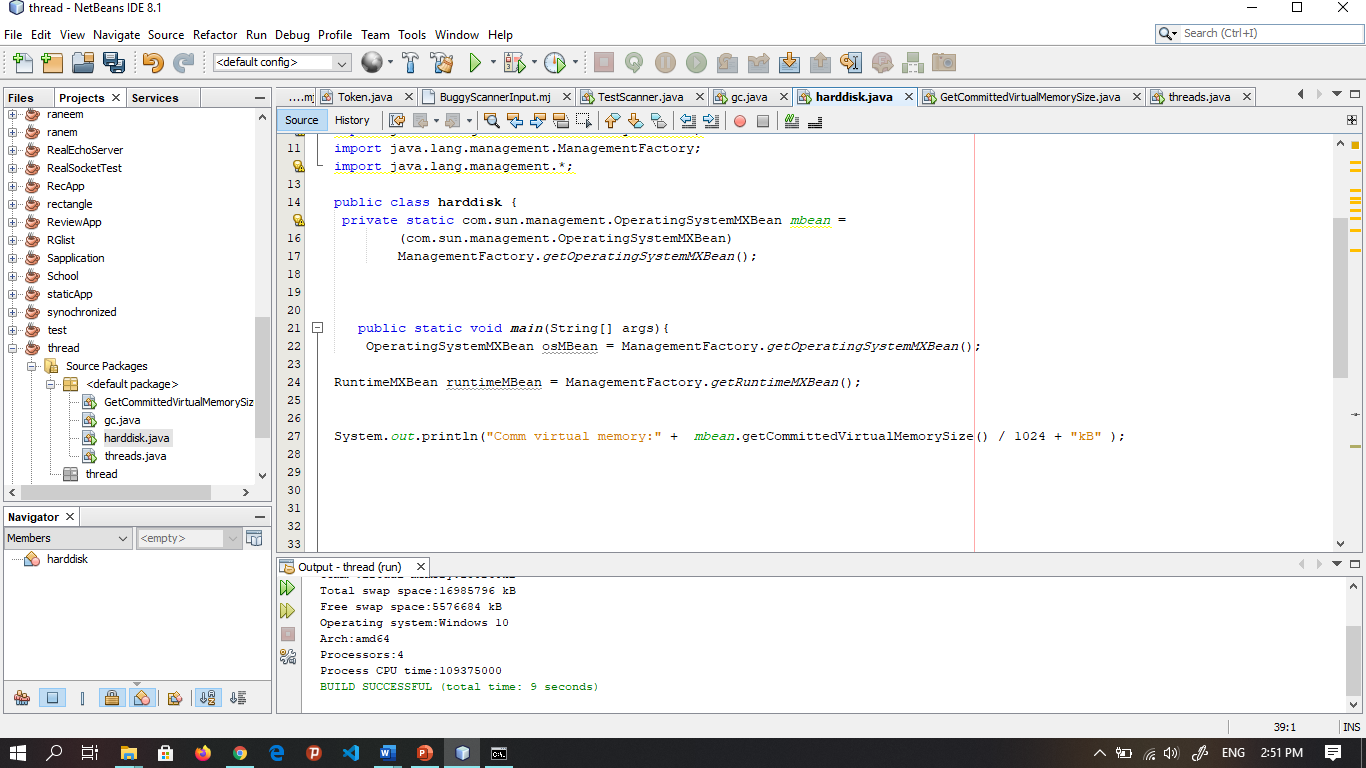
الذاكرة الافتراضية مهمة لتحسين أداء النظام، وتعدد المهام، واستخدام البرامج الكبيرة والمرونة. ومع ذلك، يجب ألا يعتمد المستخدمون على الذاكرة الظاهرية كثيرًا، لأن استخدام البيانات الافتراضية أبطأ بكثير من استخدام ذاكرة الوصول العشوائي. إذا اضطر نظام التشغيل إلى تبديل البيانات بين الذاكرة الظاهرية وذاكرة الوصول العشوائي كثيرًا، فيمكن أن يجعل الكمبيوتر يشعر بالبطء الشديد - وهذا ما يسمى thrashing. تم تطوير الذاكرة الظاهرية في وقت كانت فيه الذاكرة الفعلية - التي يشار إليها أيضًا باسم ذاكرة الوصول العشوائي - باهظة الثمن.

**طريقة تحسين الذاكرة الظاهرية :**

يعد وجود مساحة كافية متاحة أمرًا ضروريًا للغاية لتعمل الذاكرة الظاهرية وذاكرة الوصول العشوائي بشكل صحيح. يمكن تحسين أداء الذاكرة الظاهرية تلقائيًا عن طريق تحرير الموارد في التسجيل

تعد ترقية ذاكرة الوصول العشوائي الخاصة بك هي الطريقة الوحيدة التي يمكنك من خلالها زيادة الذاكرة الافتراضية الخاصة بك، عن طريق زيادة حجم الذاكرة الإجمالية المتاحة للنظام. في ذلك، سوف تخفف من مشكلة الذاكرة الافتراضية أثناء العملية ويمكن أن ترى زيادة في سرعة نظامك أيضًا

**طريقة حسابه:**



**Thread counts**

**تعريفه:**

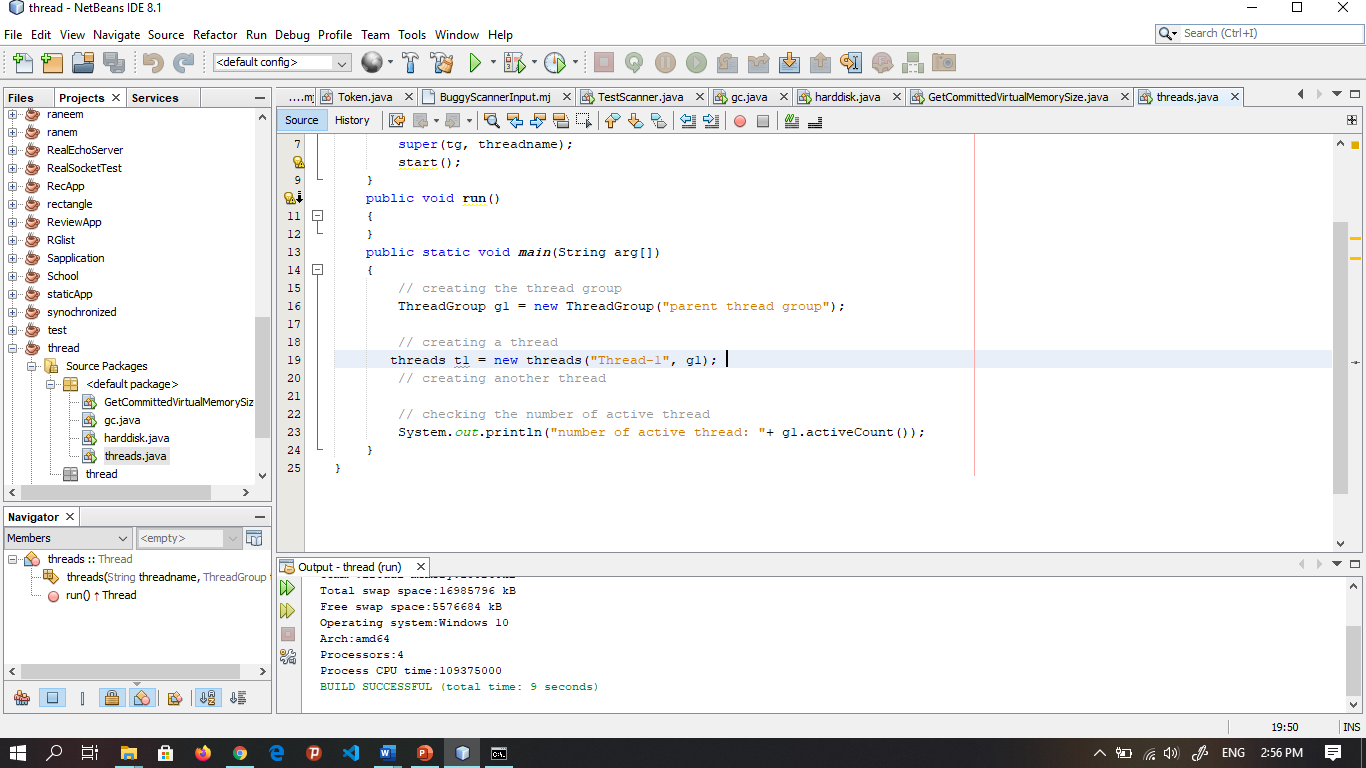
الحد الأقصى لعدد الطلبات المتزامنة التي يمكن للخادم معالجتها. عندما يصل الخادم إلى الحد الأقصى أو يطلب سلاسل الرسائل، فإنه يؤجل معالجة الطلبات الجديدة حتى ينخفض ​​عدد الطلبات النشطة إلى أقل من الحد الأقصى. ستؤدي زيادة هذه القيمة إلى تقليل أوقات استجابة.

**أهميته:**

اضبط قيمة عدد الخيوط بناءً على الحمل وطول الوقت لطلب متوسط. بشكل عام، قم بزيادة هذا الرقم إذا كان لديك وقت خمول لوحدة المعالجة المركزية وطلبات معلقة؛ قم بتقليله إذا أصبحت وحدة المعالجة المركزية محملة بشكل زائد. إذا كان لديك العديد من عملاء HTTP 1.0 (أو عملاء HTTP 1.1 الذين ينقطعون بشكل متكرر)، فاضبط قيمة المهلة لتقليل الوقت الذي يظل فيه الاتصال مفتوحًا.

تتراوح قيم عدد سلاسل الطلبات المناسبة من 100 إلى 500، اعتمادًا على الحمل. إذا كان نظامك يحتوي على دورات CPU إضافية، فاستمر في زيادة عدد مؤشرات الترابط بشكل تدريجي ومراقبة الأداء بعد كل زيادة تدريجية. عندما يشبع الأداء (يتوقف عن التحسن)، توقف عن زيادة عدد الخيوط.

**طريقة حسابه:**

****

**Error rate**

**تعريفه:**

يتم حساب هذا المقياس على أنه نسبة الإجابات غير الصحيحة إلى الإجابات الصحيحة خلال فترة زمنية. النتائج محسوبة بالنسبة المئوية. تحدث الأخطاء عادةً عندما يتجاوز تحميل البرنامج سعته.

تقيس معدلات الخطأ النسبة المئوية للطلبات التي تنطوي على مشاكل مقارنة بإجمالي الطلبات. ليس من غير المألوف وجود بعض الأخطاء مع تحميل عالٍ، ولكن من الواضح أنه يجب تقليل معدلات الخطأ إلى الحد الأدنى لتحسين تجربة المستخدم.

**أهميته:**

معدل الخطأ هو مقياس مهم لأنه يقيس "فشل الأداء" في التطبيق. يخبرك بعدد الطلبات الفاشلة التي تحدث في نقطة معينة في وقت اختبار التحميل.

تكون قيمة هذا المقياس أكثر وضوحًا عندما يمكنك بسهولة رؤية زيادة النسبة المئوية للمشكلات بشكل ملحوظ حيث ينتج عن الحمل الأعلى المزيد من الأخطاء. في العديد من اختبارات الحمل، سيكون هذا الارتفاع في معدل الخطأ حادًا. يخبرك هذا الارتفاع السريع في الأخطاء بالمكان الذي يتم فيه الضغط على النظام المستهدف بما يتجاوز قدرته على تقديم أداء مناسب.

سيرد خادم الويب رمز حالة HTTP في رأس الاستجابة. عادةً ما تكون الرموز العادية 200 (موافق) أو شيء ما في النطاق 3xx يشير إلى إعادة توجيه على الخادم. رمز الخطأ الشائع هو 500، مما يعني أن خادم الويب يعرف أن لديه مشكلة في تلبية هذا الطلب. هذا بالطبع لا يخبرك سبب المشكلة، لكنك على الأقل تعلم أن الخادم يدرك أن هناك مشكلة فنية في النظام في مكان ما.

**طريقة حسابه:**

(Number of fail request / Number of seconds)

بعض المقاييس التي تؤثر ببعضها :

Throughput &&hits per second

في Analyzer، لدينا خيار لدمج الرسوم البيانية، باستخدام ذلك يمكننا دمج عدد الزيارات لكل ثانية والرسم البياني للإنتاجية لفهم كيفية أداء التطبيق.

1. عندما يتم رسم الرسم البياني للإنتاجية أعلى من الرسم البياني عدد الزيارات في الثانية، فإن أداء التطبيق رائع، مما يعني أن الخوادم تعالج الطلبات بمعدل أكبر من الطلبات التي تصل إلى الخادم.

2. عندما يكون الرسمان البيانيان متوازيان أو متداخلين مع بعضهما البعض، يكون أداء التطبيق جيدًا، مما يعني أن الخوادم تعالج الطلبات بنفس معدل وصول الطلبات إلى الخادم.

3. عندما يتم رسم الرسم البياني للإنتاجية أقل من الرسم البياني عدد مرات الظهور في الثانية، فإن التطبيق يواجه مشكلات، مما يعني أن الخوادم تعالج الطلبات بمعدل أقل من الطلبات التي تصل إلى الخادم.

Ram &&committed memory

عند الحديث عن الاختلافات بين الذاكرة الظاهرية والذاكرة، يُرى عادةً أن أكبر تمييز هو السرعة.

ذاكرة الوصول العشوائي أسرع بكثير من الذاكرة الافتراضية. ومع ذلك، تميل ذاكرة الوصول العشوائي إلى أن تكون أكثر تكلفة من الذاكرة الافتراضية.

عندما يحتاج الكمبيوتر إلى مساحة تخزين، فإن ذاكرة الوصول العشوائي هي أول استخدام. يتم استخدام الذاكرة الظاهرية عندما تكون ذاكرة الوصول العشوائي ممتلئة، لأنها أبطأ.

يمكن للمستخدمين إضافة ذاكرة الوصول العشوائي بشكل نشط إلى جهاز الكمبيوتر عن طريق شراء المزيد من شرائح ذاكرة الوصول العشوائي وتثبيتها إذا كانوا يعانون من تباطؤ بسبب عمليات تبديل الذاكرة التي تحدث كثيرًا. يعتمد مقدار ذاكرة الوصول العشوائيRAM على ما يتم تثبيته على جهاز الكمبيوتر. من ناحية أخرى، فإن الذاكرة الافتراضية محدودة بحجم القرص الصلب للكمبيوتر. يمكن غالبًا التحكم في إعدادات الذاكرة الظاهرية من خلال نظام التشغيل

Thread counts &&response time

كلما كانت النياسب المتزامنة أكثر نشاطًا، كلما استغرق الخادم وقتًا أطول للاستجابة

# الفصل الثالث إدارة المشروع

*في هذا الفصل يتم تقديم خطة عمل المشروع، وتحديد المنهجية المتبعة لتطوير النظام.*

## 1.3-مقدّمة

من الأمور المهمة في أي مشروع وجود خطة، تسهل العمل لبناء هذا المشروع، وتحقق ضمان تسليم النظام المطلوب في الوقت المناسب من خلال جدولة العمل، وتضمن سير العمل بما يتوافق مع متطلبات النظام.

## 2.3-خطة إدارة المشروع

في الواقع من الصعب اختيار منهجية مناسبة للتطوير، خصوصاً في حالتنا فنحن لا نعلم العقبات التي يمكن أن تعترضنا، لكن يمكن التنبؤ مسبقاً بمدى صعوبة مرحلة معينة، وتحديد وقت إضافي احتياطاً، وهذا ما تمّ العمل عليه، إذ قسمنا النظام المطلوب إلى عدة أجزاء حسب الوظائف التي يتعين عليه القيام بها، وقدرنا مدى صعوبة كل مرحلة، وتم تحديد الفترة الزمنية المناسبة وفقاً لهذه التقديرات، امتدّ العمل نحو 32 أسبوع بمعدل 6 ساعات يومياً، على عدة مراحل كما يوضح الجدول التالي:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| رقم المرحلة | اسم المرحلة | المدة (يوم) | تاريخ البدء | تاريخ الانتهاء |
| 1 | دراسة نظرية | 56 | 1/11/2020 | 1/1/2021 |
| 2 | تحديد متطلبات المشروع | 10 | 25/1/2021 | 5/2/2021 |
| 3 | تحليل النظام | 15 | 10/2/2021 | 25/2/2021 |
| 4 | التصميم | 20 | 1/3/2021 | 20/3/2021 |
| 5 | التنجيز العملي | 40 | 1/4/2021 | 10/5/2021 |
| 6 | الاختبارات | 20 | 15/5/2021 | 5/6/2021 |
| 7 | التوثيق النهائي | 14 | 1/7/2021 | 14/7/2021 |

الجدول 2: المخطط الزمني لمراحل العمل

استمرّت كتابة أجزاء التقرير طيلة فترة المشروع وتمّ تجميع الأجزاء والتوثيق النهائي في نهاية هذه الفترة.

# الفصل الرابع تحليل المتطلبات

*يعرض هذا الفصل ما يجب أن يقوم به النظام من خلال تحليل للمتطلَّبات المذكورة في الفصل الأول.*

## 1.4-مقدّمة

عرضنا في الفصل الأول متطلَّبات النظام، حيث تعرِّف هذهِ المتطلبات ما يجب على النظام فعله. سنقدم في هذا الفصل عرضاً تحليلياً للنظام.

## 2.4-التطبيقات المشابهة

**JMeter**

تطبيق **Apache JMeter™** هو برنامج مفتوح المصدر، وهو تطبيق Java خالص 100٪ مصمم لتحميل اختبار السلوك الوظيفي وقياس الأداء. تم تصميمه في الأصل لاختبار تطبيقات الويب ولكنه امتد منذ ذلك الحين إلى وظائف الاختبار الأخرى.

**من السلبيات التي نجدها في الموقع:**

- استهلاك الذاكرة: يمكن لـ JMeter محاكاة الحمل الثقيل وتصور تقرير الاختبار. قد يستهلك هذا الكثير من الذاكرة ويمكن أن يؤدي إلى نفاد الذاكرة تحت الحمل الثقيل.

- تطبيق الويب فقط: يعد JMeter أداة جيدة لاختبار تطبيق الويب ولكنه ليس أداة مناسبة لاختبار تطبيقات سطح المكتب.

- عدم وجود دعم لـ JavaScript: JMeter ليس مستعرضًا، لذلك لا يمكنه تشغيل JavaScript في تطبيق الويب. لديها دعم محدود لمعالجة JavaScript أو Ajax، وهذا قد يؤثر على دقة المحاكاة.

**من أهم ميزات الموقع:**

- مجاني ومفتوح المصدر.

- يستخدم على أنظمة تشغيل مختلفة **(Linux، Windows، Mac).**

**Web Runner**

LoadRunner هي أداة اختبار برمجية من Micro Focus. يتم استخدامه لاختبار التطبيقات وقياس سلوك النظام وأدائه تحت الحمل. يمكن لـ LoadRunnerمحاكاة آلاف المستخدمين في نفس الوقت باستخدام برنامج تطبيقي، وتسجيل أداء المكونات الرئيسية للتطبيق وتحليلها لاحقًا.

**من سلبيات الموقع:**

* يحدد السبب الجذري لمشاكل أداء التطبيق.
* يقلل تكاليف البرامج والأجهزة من خلال التنبؤ بسعة البرامج وقابليتها للتوسع.

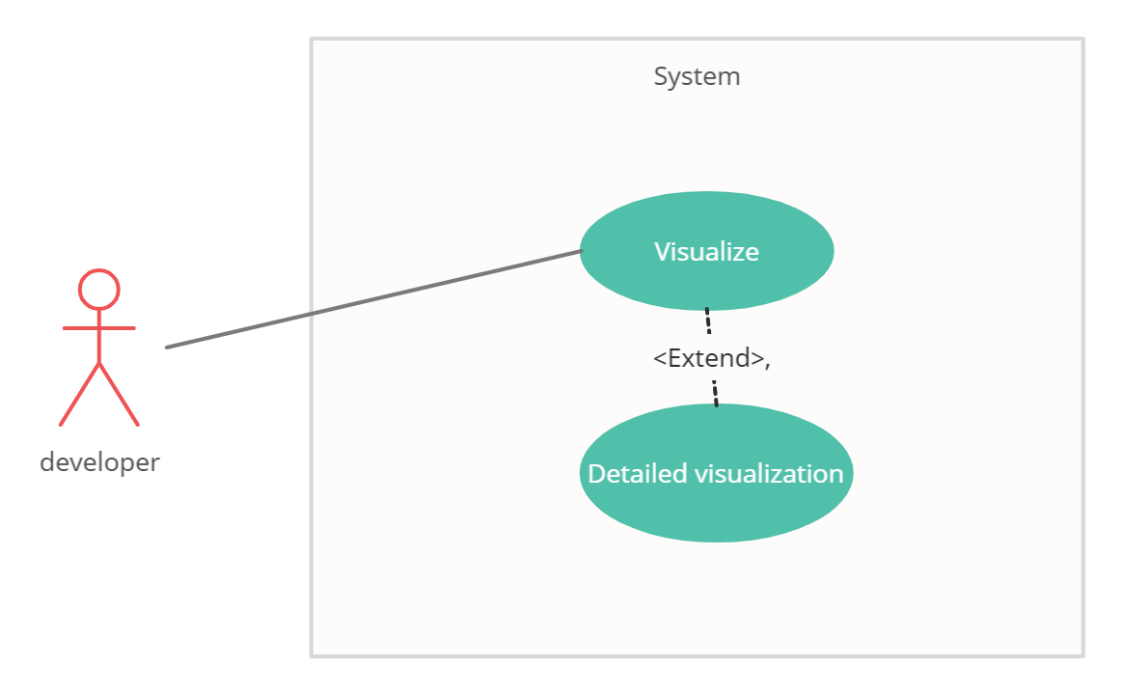
**من أهم ميزات الموقع:**

* يستخدم قدرًا كبيرًا من الذاكرة ويؤدي إلى حدوث أعطال إذا لم يتمكن النظام من تلبية متطلبات الحوسبة الخاصة به.
* القدرة على التصحيح يحتاج إلى تحسين غالي جدا.

## 3.4-مخطط حالات الاستخدام

نعرُض هنا مخطّط حالات الاستخدام الذي يوضح كيفية تفاعل النظام مع البيئة التي يعمل ضمنها، وكيفية تفاعل مستخدم النظام معه.

يوضح الشكل 10 مخطط حالات الاستخدام، حيث تمّ استخدام الأسهم الموجهة بين الفاعل الأساسي وحالة الاستخدام. ونلاحظ في الشّكل الوظائف الأساسية التي يجب أن يؤمّنها النظام.



الشكل 10: مُخطّط حالات الاستخدام

## 4.4-توصيف حالات الاستخدام"Narrative Use Case"

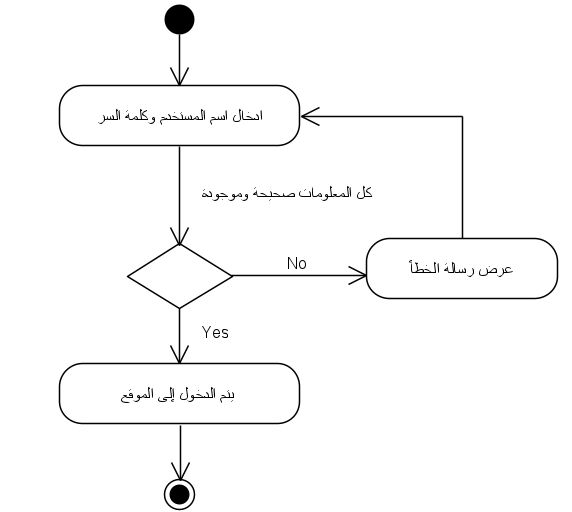
نكتب لكل حالة استخدام توصيف يشرح تدفق الأحداث ومختلف السيناريوهات الممكنة.

|  |  |
| --- | --- |
| حالة الاستخدام | **إظهار الموارد** |
| رقم الحالة | 1 |
| شرح مختصر | يقوم التطبيق بعرض متوسطات النتائج بالنسبة لزمن تنفيذ المشروع |
| شريحة المستخدمين | المطور والمحلل. |
| الشروط السابقة | توافر الاتصال مع الخادم (الاكسير). |
| خطوات حالة الاستخدام | 1. يقوم المستخدم بإدخال المعلومات المطلوبة. 2. يقوم بالضغط على زر إظهار. |
| الحالة اللاحقة | يتم عرض الرسم البياني. |
| التدفق البديل | في حال عدم توفّر اتّصال مع المخدم لن يتم اظهار الرسم البياني وسيتم تتبع الخطوات التالية:   1. التأكد من الاتصال بالخادم 2. يقوم المستخدم بإدخال المعلومات المطلوبة. 3. يقوم بالضغط على زر إظهار. |

الجدول 1: توصيف حالة إظهار الموارد

## 4.5-مخطط النشاطات"Activity Diagram"

حالة تسجيل دخول:



# الفصل الخامس تصميم النظام

*في هذا الفصل نعرض توصيفاً للنظام وكيفية عمله، حيث نعرض شرح مفصل للتصميم المقترح*

## 1.5-مقدّمة

تعرض مرحلة التصميم البنى البرمجية التي سيتم استخدامها، نقدم فيها أيضاً واجهات التخاطب مع المستخدم، بالإضافة الى المكاتب التي يستخدمها النظام لإنجاز مهامه، نتكلم أيضاً عن دخل النظام وخرجه.

## 2.5-التّصميم **الأولي للنظام**

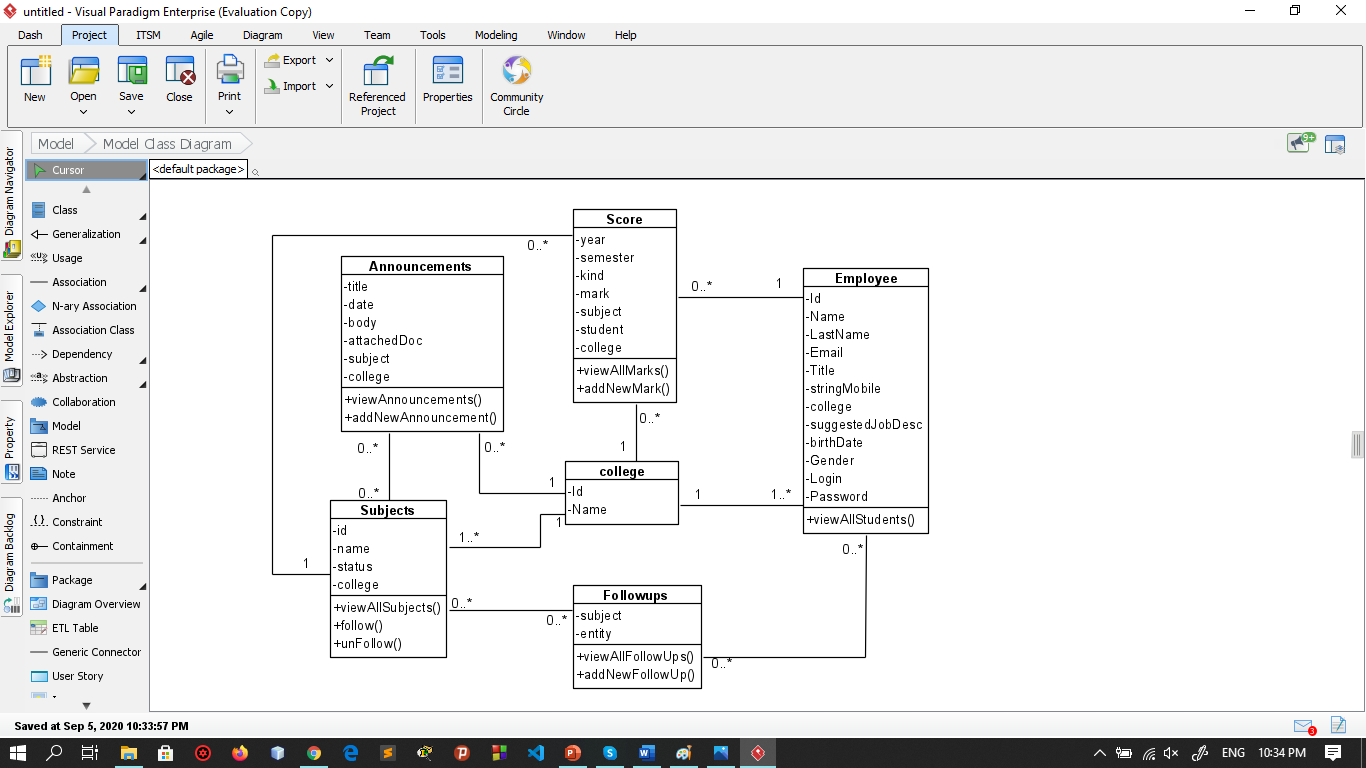
## 

## 3.5-مكونات النظام

هي عبارة عن الصّفوف المستخدمة والمكتبات الخاصّة ببناء هذه الصّفوف، بالإضافة إلى واجهات البرنامج، لتشكّل جميعها بُنية التطبيق، الذي تؤمّن المهام المطلوبة منَ التطبيق التي سُبق وأن ذكرت في الفصل الأول ضمن المتطلبات الوظيفية، سنتكلّم عن هذه المكونات لاحقاً بالتفصيل.

**4.5-مخطط الصفوف"Class Diagram"**

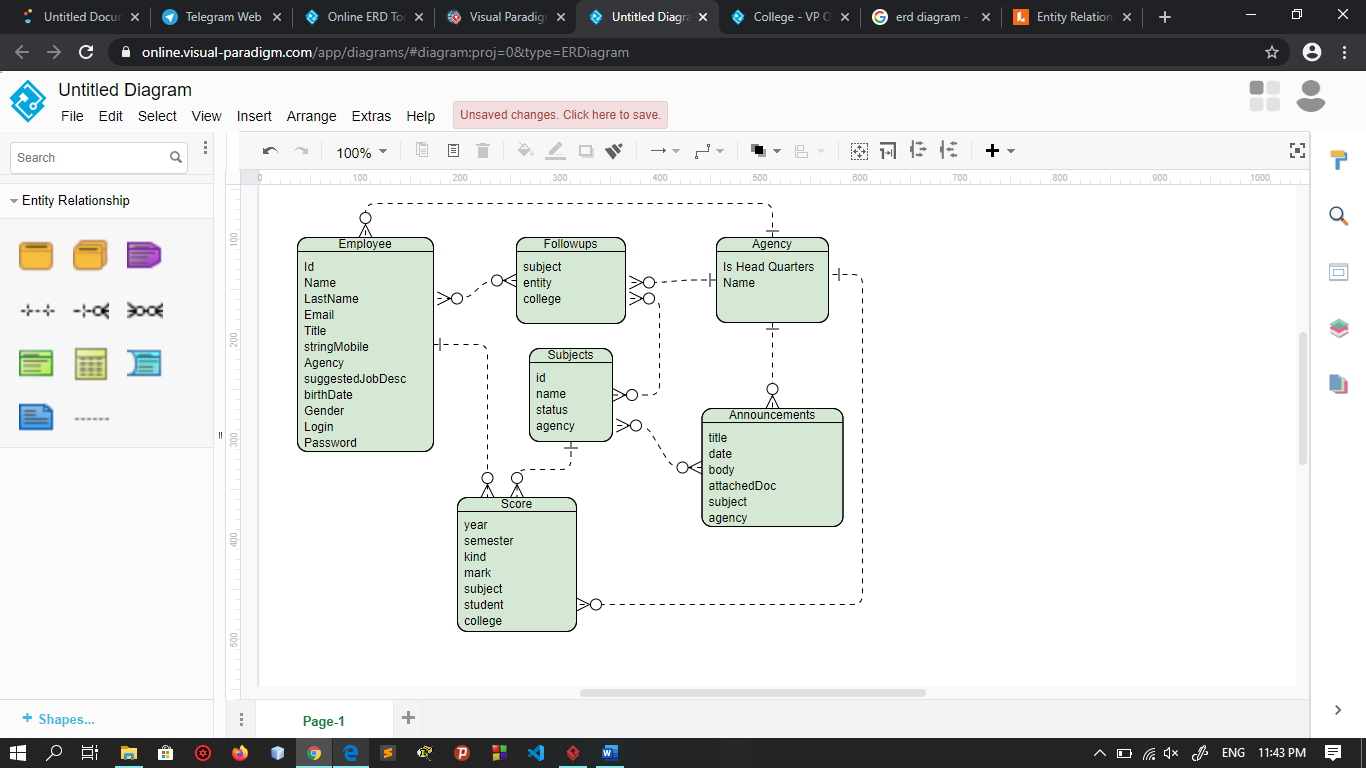
هو نوع من الرسم التخطيطي للهيكل الثابت الذي يصف بينة النظام، من خلال إظهار فئات النظام وخصائصه، وعملياته، والعلاقات بين الكائنات، وقد تم استخراج الصفوف من الهيكلية العامة للنظام والمتطلبات المعطاة.



## 

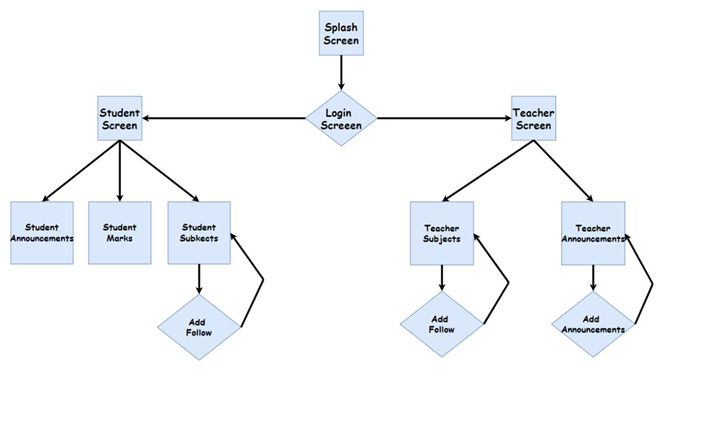
## 5.5-مخطط العلاقات والارتباطات"ERD"

يظهر المخطط قاعدة بيانات الموقع وارتباط جداولها معاً وما يحتويه كل جدول من كائنات وأنواعها.



### 6.5-واجهات النظام

تتميز واجهات النظام بالبساطة والوضوح وهي تتبع المخطّط التالي:



# الفصل السادس التنفيذ والاختبارات

*يعرض هذا الفصل بناء النظام واختباره وتدريب المستخدمين على النظام.*

## 1.6- مقدمة

نتحدث في هذا الفصل عن الأدوات المستخدمة في تنفيذ المشروع، نتطرّق أيضاً للحديث عن المشاكل البرمجية التي واجهتنا أثناء التنفيذ العملي، نعرض بعض الاختبارات للتطبيق، كما نزوّد المستخدم بدليل مساعد على استخدام التطبيق، ونختم هذا الفصل بالدروس المستفادة والخلاصة والمقترحات.

في هذه المرحلة يجري بناء النظام واختباره للتأكد من أدائه للعمل كما جرى تصميمه. نتحدث في هذا الفصل عن الأدوات المستخدمة في تنفيذ المشروع، ثم الحديث عن العوائق البرمجية التي واجهتنا ثم ننتقل إلى خطة الاختبارات المطبقة بعدها نتطرّق للحديث دليل استخدام التطبيق.

## 2.6-أدوات التنفيذ

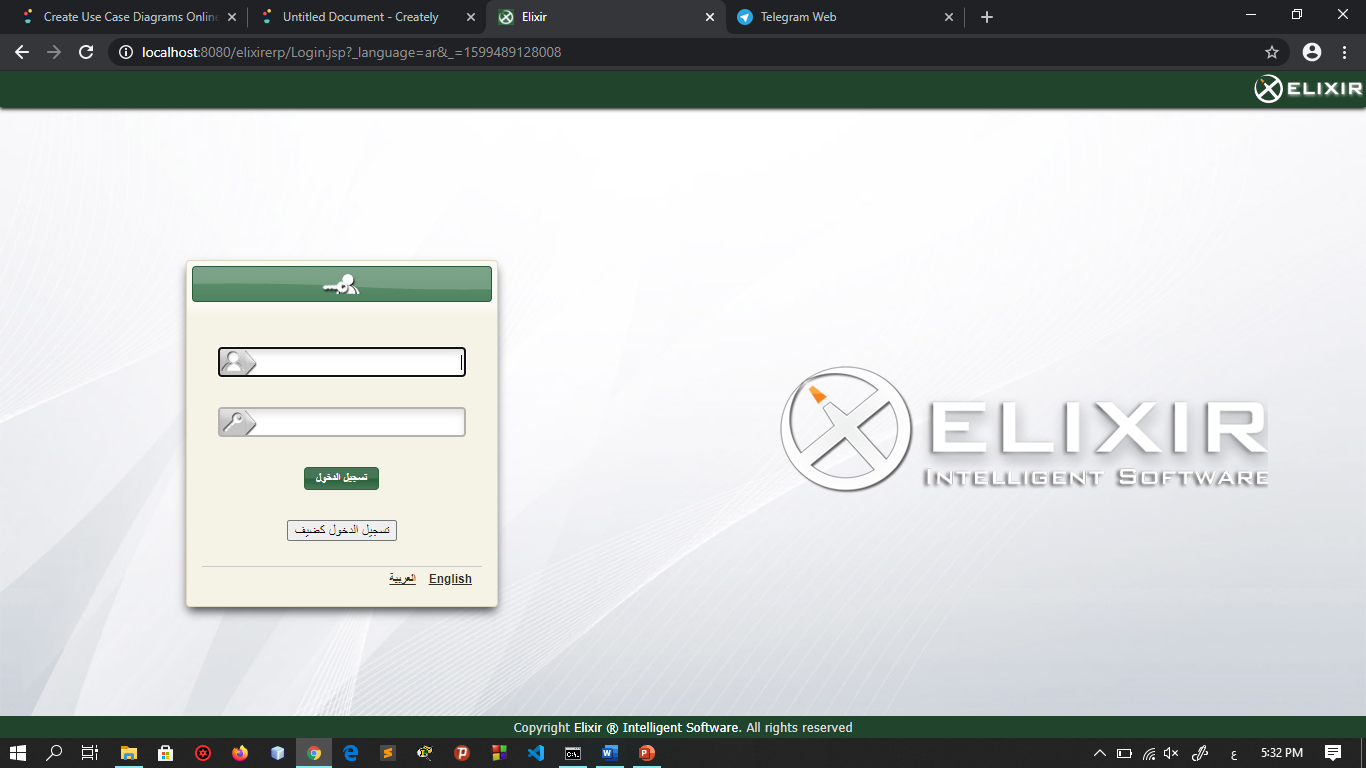
* بيئة العمل Elixir
* Visual Studio code لتصميم وتنفيذ النظام.

## 3.6-الاختبارات

* تمّ اختبار التطبيق في جميع المهام التي يقوم بها، سنعرضهُ أثناء مقابلة الدفاع.

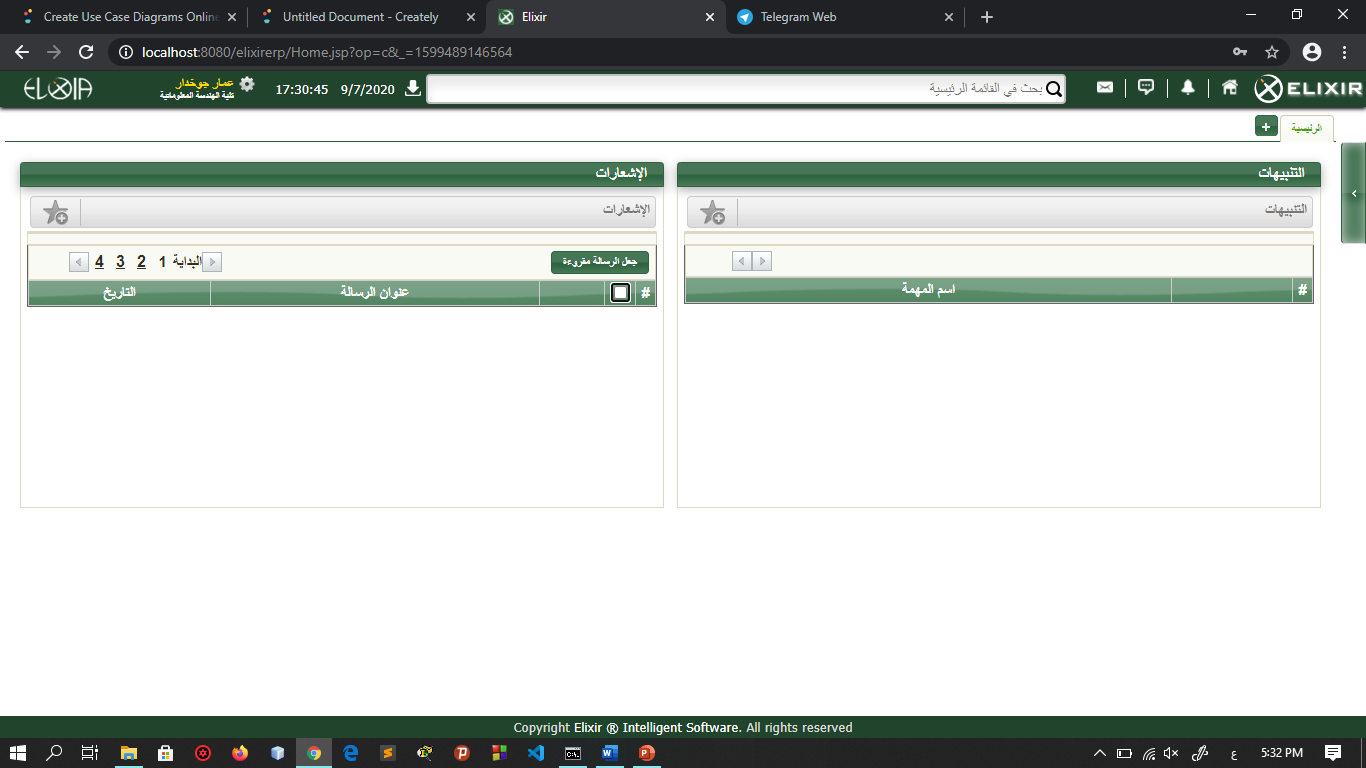
**واجهات التفاعل مع النظام في الموقع:**

واجهة تسجيل الدخول



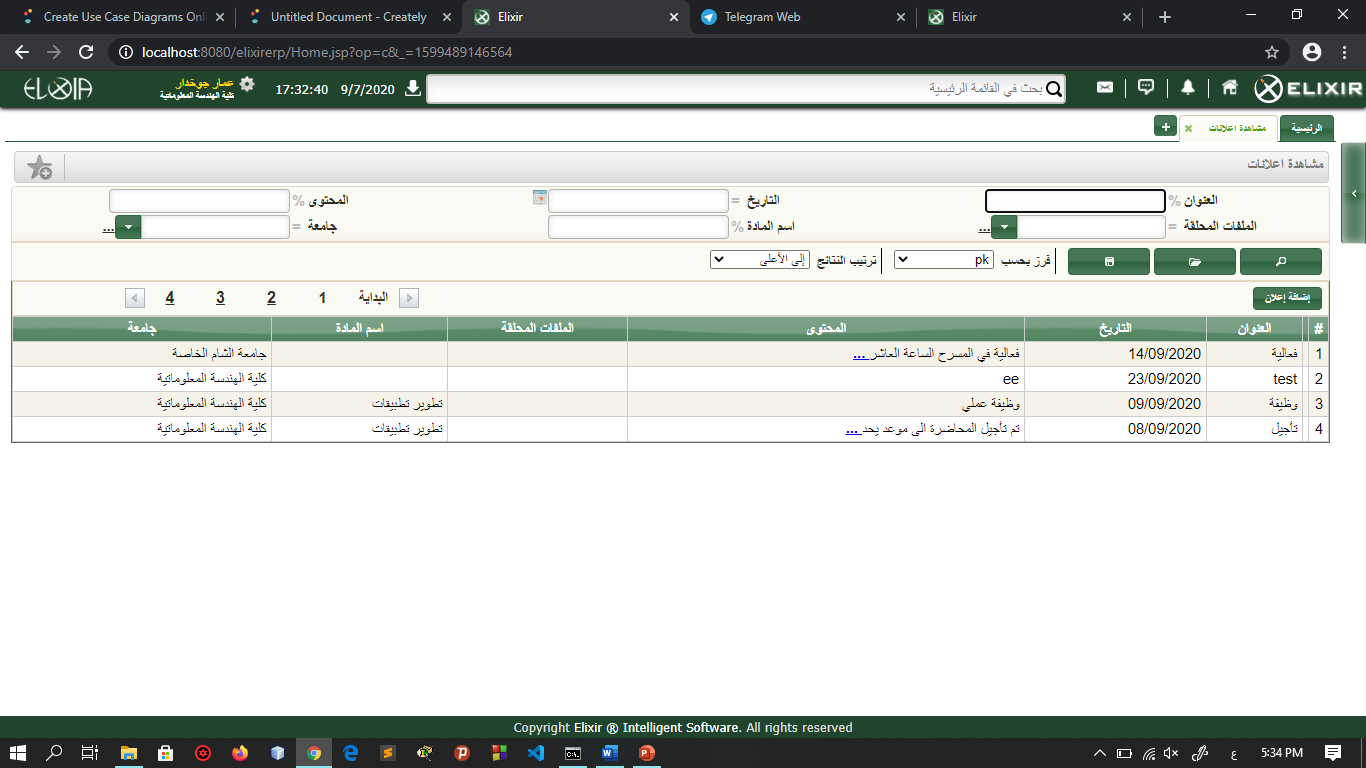
الشكل3: واجهة تسجيل الدخول

الواجهة الرئيسية للموقع

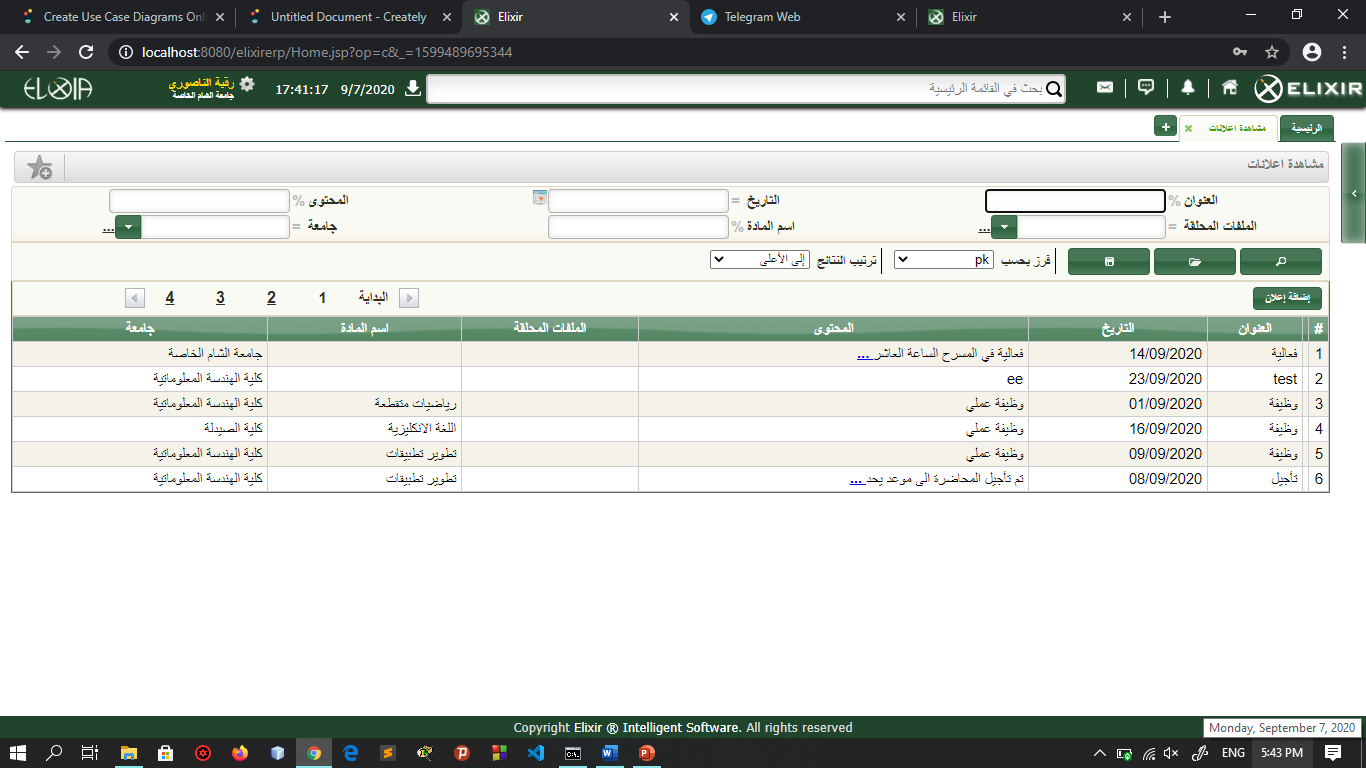


الشكل 4: الواجهة الرئيسية للمواقع

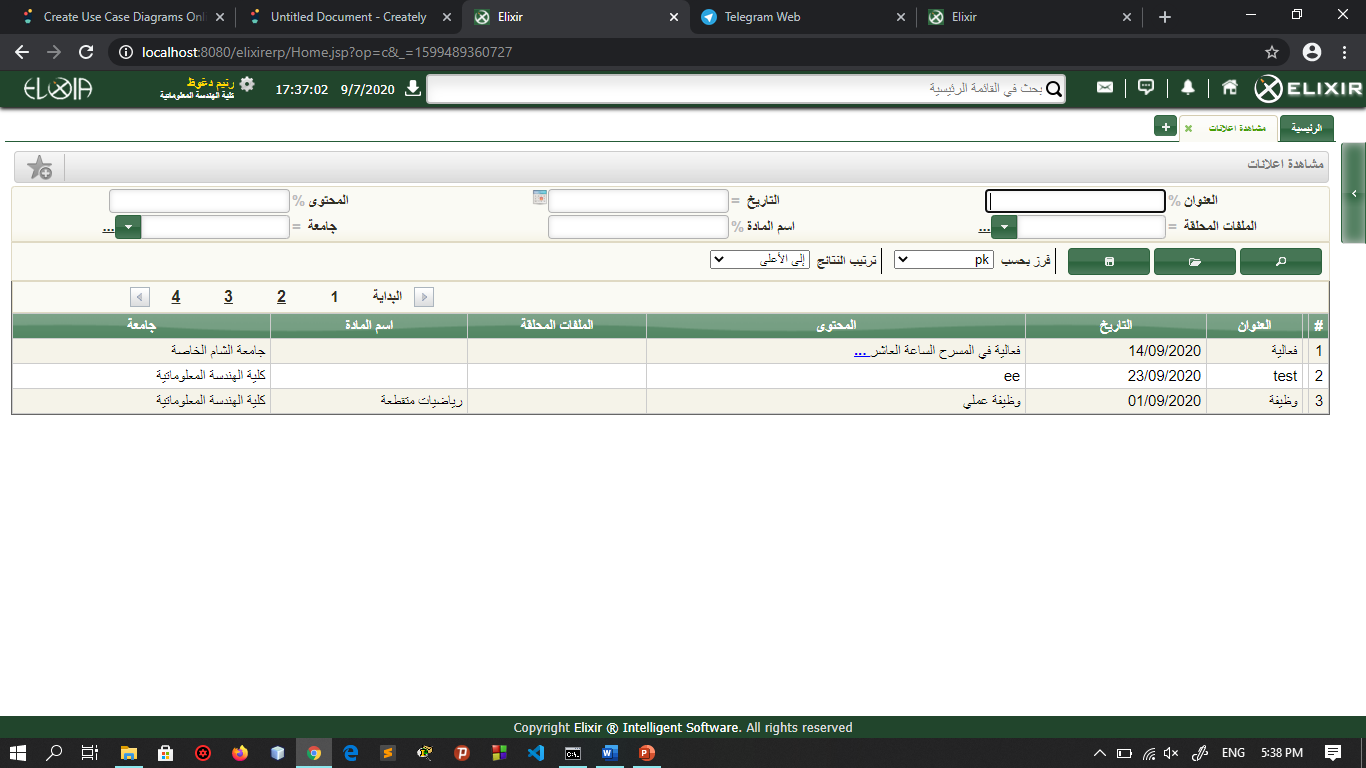
واجهة مشاهدة الإعلانات: للطالب و الأستاذ يتم عرض الإعلانات على مستوى جامعته و كليته و المواد التي قام بمتابعتها، أما الموظف الإداري فيستطيع مشاهدة جميع الإعلانات التي تم نشرها. في حالة كان أستاذ أو موظف إداري سيكون لديه صلاحية إضافة إعلان أما إذا كان طالب فلن يظهر له هذا الخيار.



الشكل 5: واجهة مشاهدة الإعلانات للأستاذ

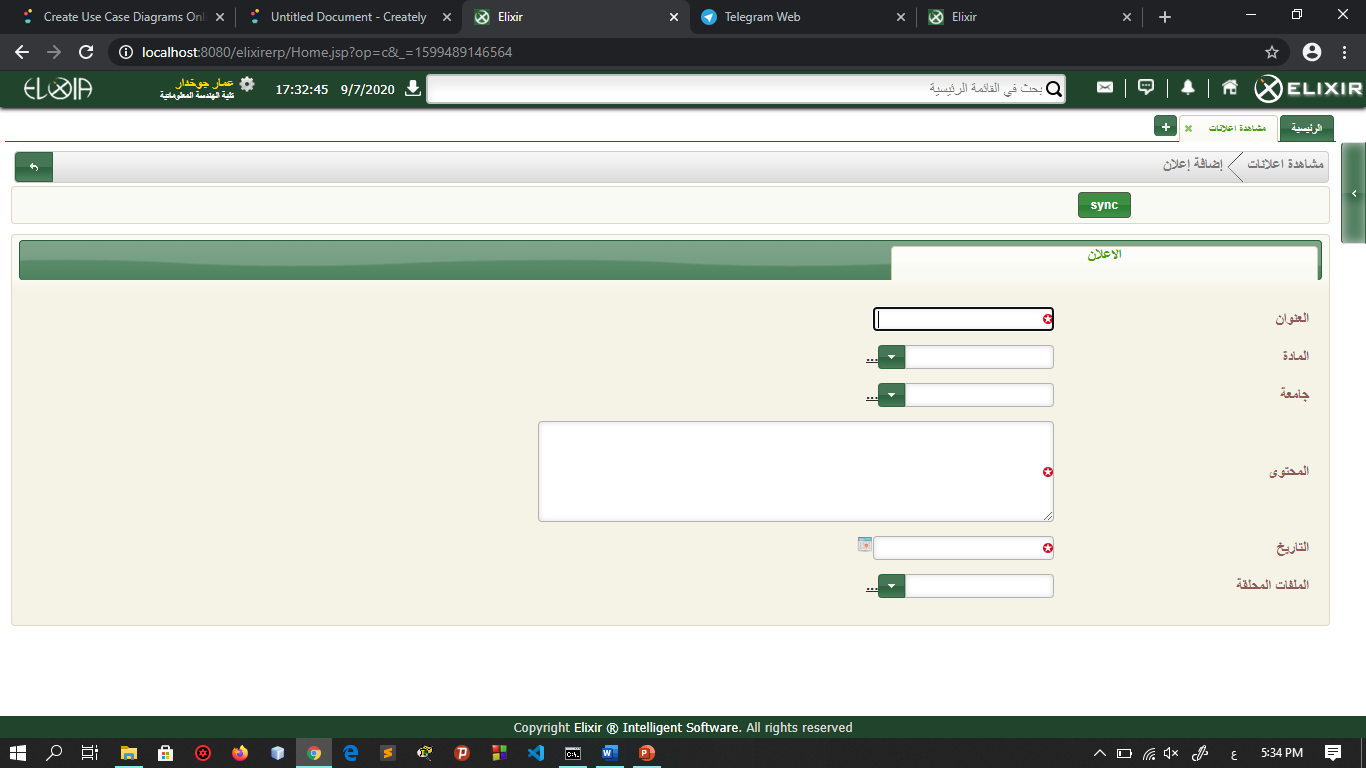


الشكل 6: واجهة مشاهدة الإعلانات للموظف الإداري



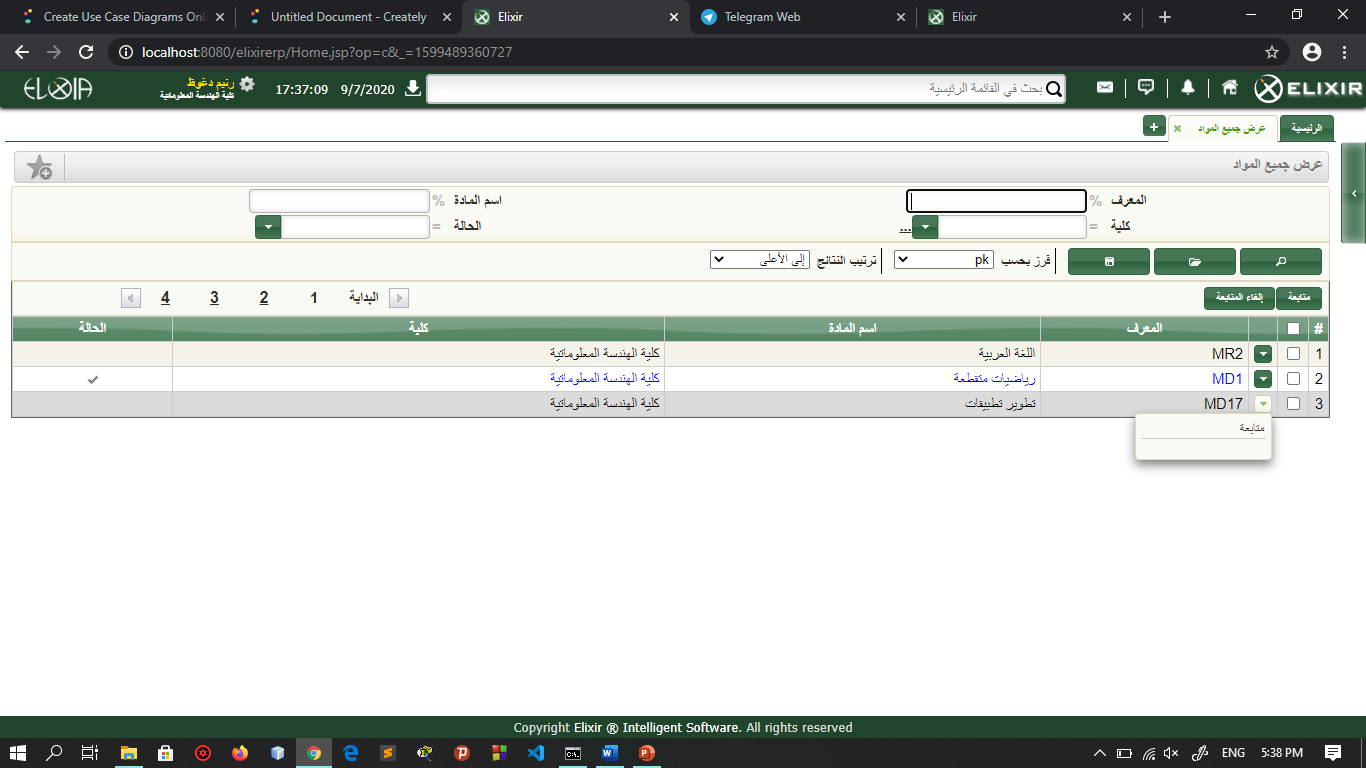
الشكل7: واجهة مشاهدة الإعلانات للطلاب

واجهة إضافة إعلان

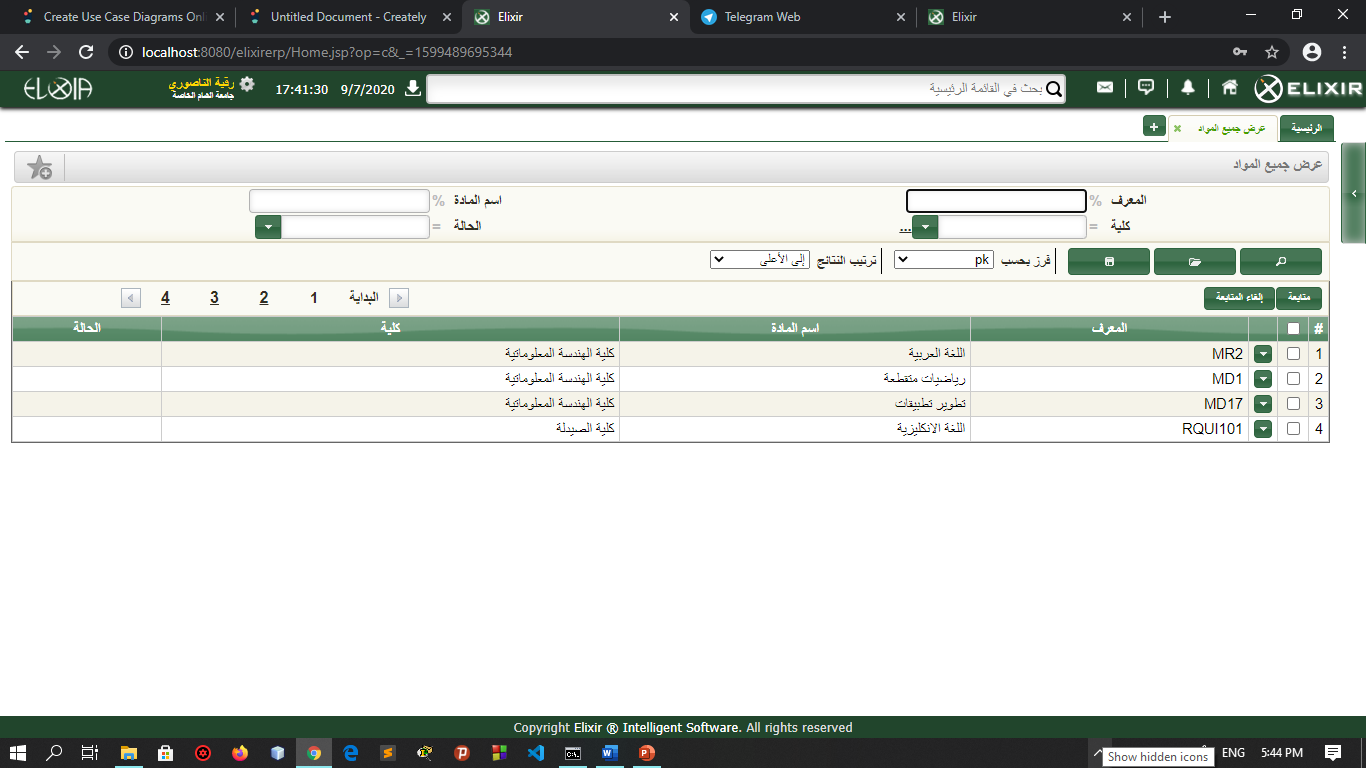


الشكل 8: واجهة إضافة إعلان

واجهة عرض المواد: تظهر عند الأستاذ والطالب مواد كليتهم فقط أما عند الموظف الإداري فتظهر مواد الجامعة كلها، وفي حال كان المادة متابعة فتظهر باللون الأزرق وأما المادة الغير متابعة فتظهر باللون الأسود

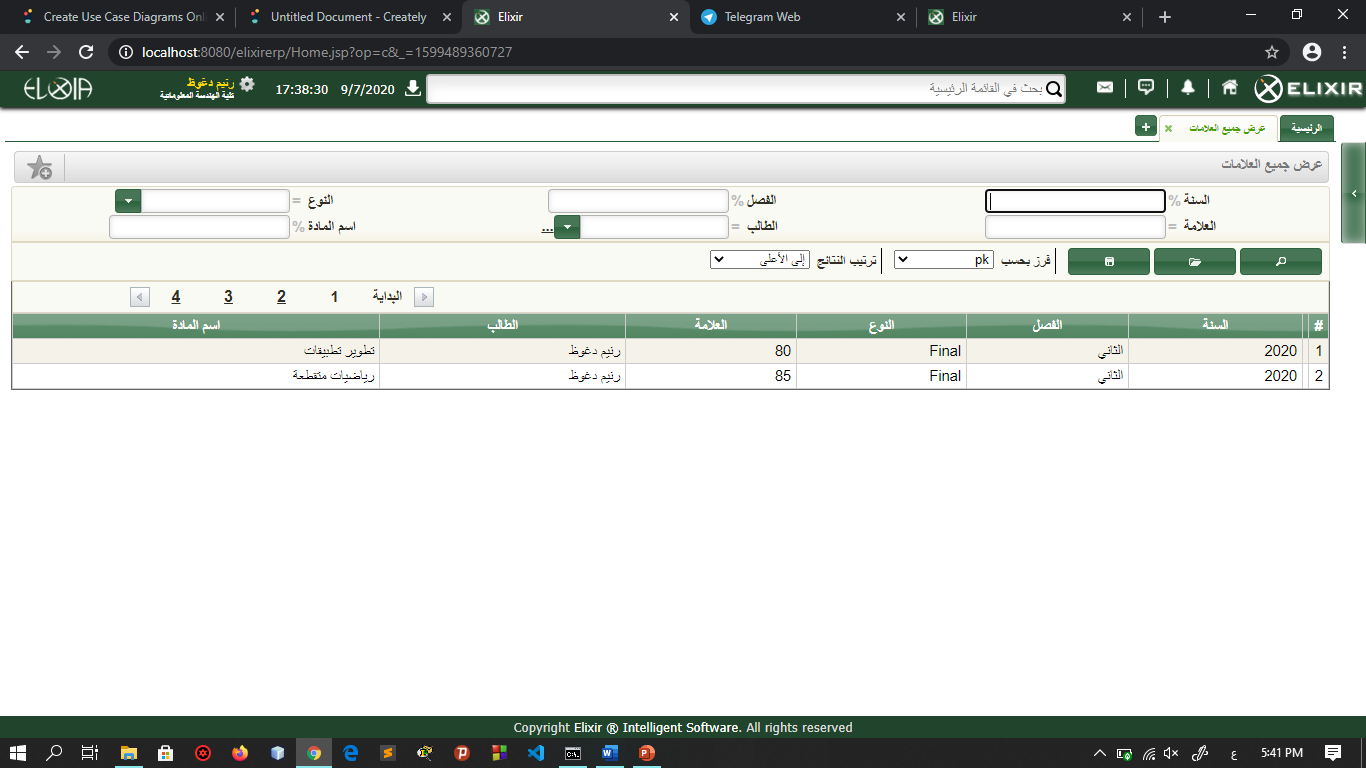


الشكل9: واجهة عرض المواد عند الطالب أو الأستاذ

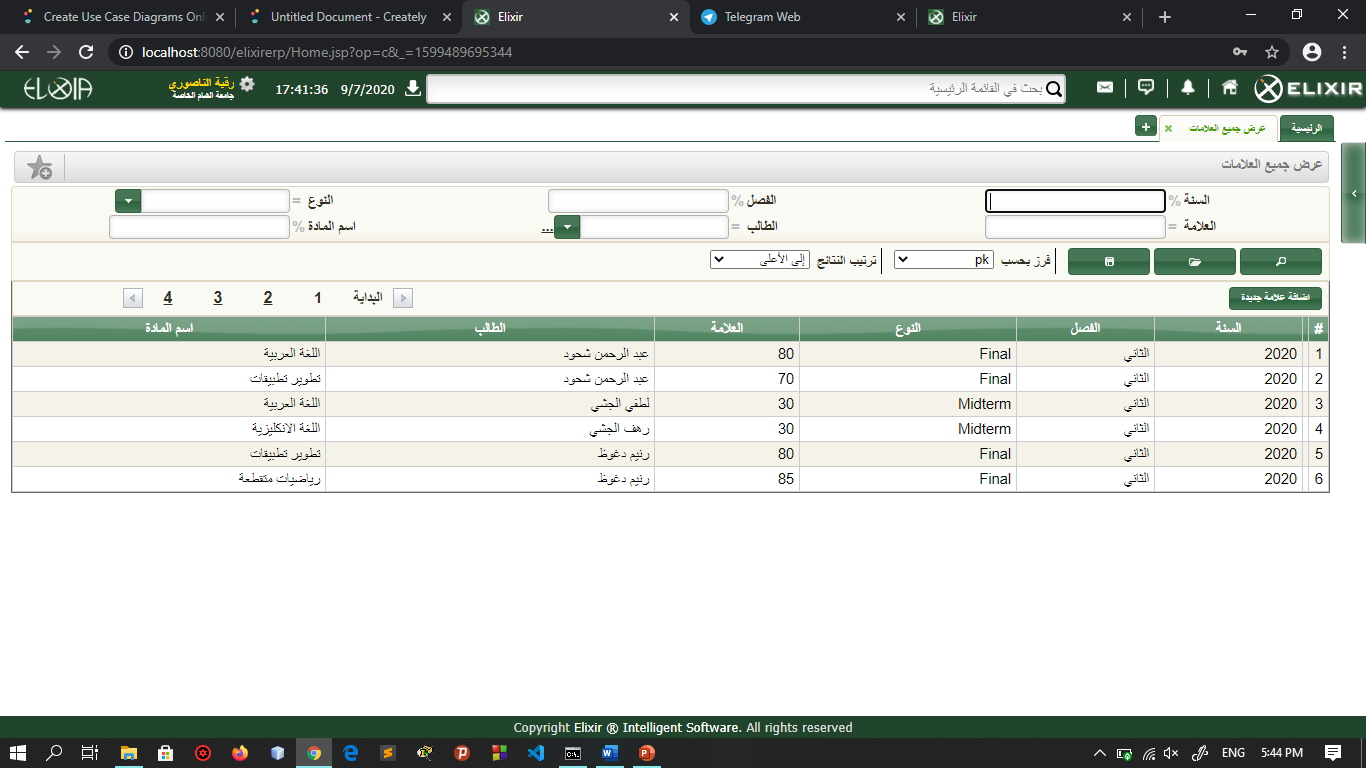


الشكل10: واجهة عرض المواد عند الموظف الإداري

واجهة عرض العلامات: يظهر عند كل طالب علاماته هو فقط ويظهر عن الموظف الإداري علامات جميع الطلاب مع صلاحيته لإضافة علامات للطلاب.

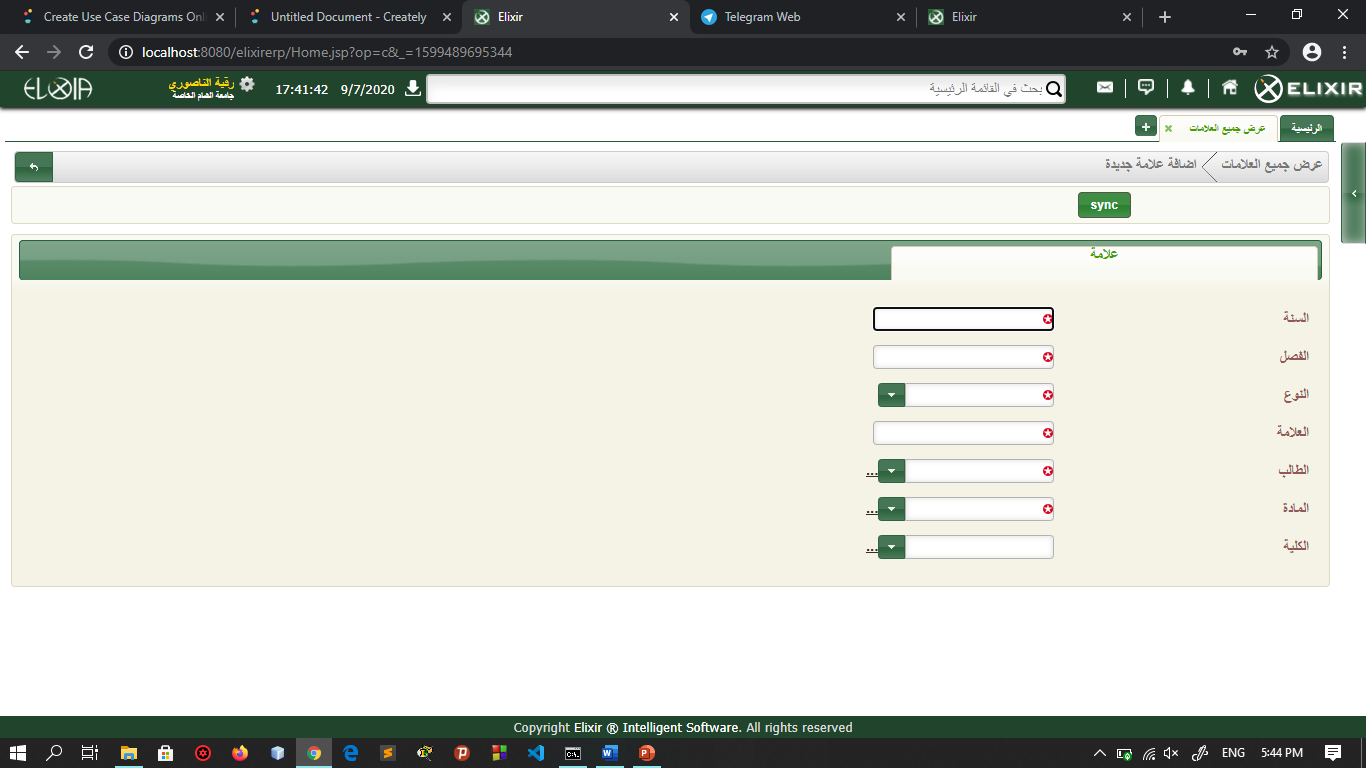


الشكل11: واجهة عرض العلامات عند الطالب



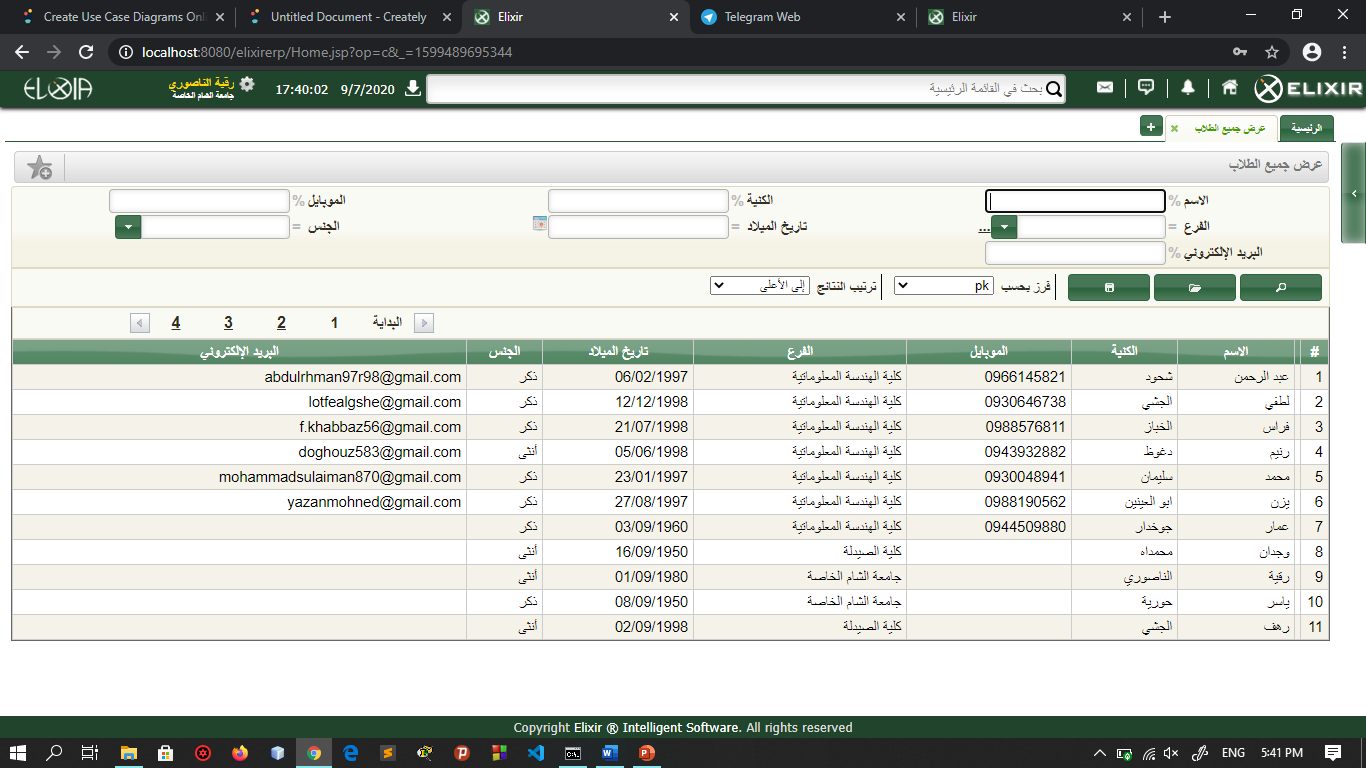
الشكل12: واجهة عرض العلامات عند الموظف الإداري

واجهة إضافة علامة: تظهر عند الموظف الإداري فقط



الشكل13: واجهة إضافة العلامات للطلاب

واجهة عرض جميع الطلاب والموظفين والأساتذة: وتكون صالحة للموظف الإداري فقط



الشكل14: عرض جميع الطلاب والموظفين والأساتذة

## 6.6- مشاكل وصعوبات

خلال مراحل العمل بالمشروع صادفتنا العديد من المشاكل، لكن كانت تقليدية بالنسبة لنا وحلها يتطلّب بذل المزيد من الجهد والوقت.

* **العائق الأول:**

تعقيدات البيئة المستخدمة وخاصة أنها بيئة جديدة وتطلب مني الوقت الطويل للتغلب على مشاكل التنصيب والتعامل.

* **العائق الثاني:**

1. التعرف على بيئات العمل المؤسساتية عموما والإكسير خصوصا
2. التعرف على أغلب مقاييس اختبار الأداء تأثيرها على بعضها البعض
3. التعرف على مكتبات التي تتعامل مع الرسومات البيانية

# الخاتمة والآفاق المستقبلية

قمنا في هذا المشروع بِبناء وتصميم تقارير باستخدام الرسومات البيانية لدعم القرار في النظم المؤسساتية الداعمة للويب وخاصة الاكسير بحيث نقوم بعرض نتائج مقاييس اختبار الأداء، ليتبين لدينا هل تحقق أن النظام قد نفذ مهام محددة بزمن محدد على عتاديات محددة بوجود عدد محدد من المستخدمين المتزامنين، يقومون بعدد محدد من العمليات.

لدينا عدة صفحات لكل منها مهام محددة، فهناك صفحة تظهر ملخص سير سيناريو معين بحيث تعطي جداول فيها متوسطات القيم أو أكبر قيمة أصغر قيمة لمقياس معين مع رسم بياني يحدد نسبة الأخطاء في هذا السيناريو.

ويوجد صفحة لعرض جميع المقاييس المتعلقة بالذاكرة وصفحة لعرض جميع المقاييس المتعلقة بالمعالج وصفحة لعرض جميع المقاييس التي يتم اجراء الاختبار عليها، بحيث يقوم المستخدم بعد اختيار الصفحة التي يريدها بإدخال المعلومات المطلوبة ثم يضغط على زر اظهار، فتظهر متوسطات لهذه القيم بالنسبة لزمن تنفيذ المشروع بشكل رسم بياني.

وفي حال أراد معرفة النتائج بشكل تفصيلي يقوم بالضغط على مكان المتوسط المطلوب على الرسم الباني فتظهر رسمة مفصلة بالنسبة لكل عملية.

ثم إذا أراد معرفة أوقات دخول عمليات محددة، فيقوم بالضغط على أي عملية يريدها من الرسم البياني السابق وستظهر لديه جميع العمليات التي تحمل نفس الاسم في ذلك الوقت بحيث تظهر أوقات بدء هذه العمليات.

يمكن توسيع المشروع بحيث يقوم بإجراء تحليل لهذه النتائج.

# المراجع

|  |  |
| --- | --- |
| [1] | 30 12 2015. [Online]. Available: https://mawdoo3.com/الهواتف\_الذكية. |
| [2] | 1 7 2016. [Online]. Available: https://www.statista.com/statistics/385001/smartphone-worldwide-installed-base-operating-systems/. |
| [3] | [متصل]. Available: https://www.wallaapps.com/2019/11/flutter.html. |
| [4] | [Online]. Available: http://www.json.org. |
| [5] | [Online]. Available: play.google.com. |
| [6] | [Online]. Available: https://en.wikipedia.org/wiki/Google\_Maps. |
| [7] | [Online]. Available: https://geojson.org/geojson-spec.html. |
| [8] | [Online]. Available: https://developers.google.com/maps/terms. |
| [9] | [Online]. Available: http://www.microimages.com/documentation/TechGuides/78googleMapsStruc.pdf. |
| [10] | [Online]. Available: https://www.codeproject.com/Articles/14793/How-Google-Map-Works. |
| [11] | [Online]. Available: https://developers.google.com/maps/documentation/utilities/polylinealgorithm. |
| [12] | [متصل]. Available: https://www.wallaapps.com/2019/11/flutter.html. |
| [13] | [متصل]. Available: https://www.wallaapps.com/2019/11/flutter.html. |