تطبيقات انترنت/برمجة التطبيقات الشبكية

Version: 2.0

نص الوظيفة (الوصف)

لدينا مجموعة من الملفات التي يعمل عليها مجموعة من الأشخاص ونريد ضمان أن لا يقوم شخصان بالكتابة على التوازي على نفس الملف والمطلوب:

1. منظومة على الويب تمكن المستخدمين من إضافة ملفاتهم إليها ووضع الملفات ضمن حالة إما حر أو بحالة استعمال ومحجوز لمستخدم ما.
2. يجب تنظيم الملفات ضمن مجموعات بحيث لا يستطيع المستخدم الوصول لجميع الملفات وإنما فقط لملفات تنتمي لمجموعات يحق له الوصول إليها.
3. حالات الاستعمال الأساسية هي **check-in** بحيث تسمح لمستخدم ما باستعراض الملفات وأن يقوم بحجز ملف حر لصالحه وتحميله وتعديله في حين تسمح **check-out** الطريقة الثانية للمستخدم بإعادة الملف المعدل و استبدال الملف القديم وإعادته لحالته حر.
4. يمكن للمستخدم أن يختار أكثر من ملف وينفذ عملية check-in ويجب على النظام أن يضمن أن جميعها حرة قبل حجزها إما جميعا أو أن يفشل الحجز لكل الملفات.
5. يجب أن يضمن النظام أن لا يتمكن مستخدمين اثنين من حجز نفس الملف بآن واحد.
6. يجب أن نستطيع تصدير تقارير بعمليات الحجز والتحرير وفقا للملف أو المستخدم.
7. يجب أن يضمن النظام أن يتمكن 100 مستخدم من العمل على التوازي معا.

المطلوب: تطوير نظام للمتطلبات السابق بحيث يحقق الشروط التالية:

المتطلبات البنيوية

المتطلب	آلية التحقق من تنفيذ المتطلب
فصل التطبيق الى 3 طبقات بحيث يساعد ذلك على توزيع العمل بشكل أفضل (3 tier)	طبيقة الواجهة: تكون Android او Web طبيقة النظام: تحوي النظام (يجب أن يكون APIs) طبقة التخزين: تحوي Database engine
يجب على النظام أن يقوم بالفصل بين المتطلبات الوظيفية وغير الوظيفية بحيث لا يوجد تداخل أو ارتباط برمجي فيما بينهم	الفصل بين المكونات بحيث يمكن الاستغناء عن اي مكون بدون مشاكل او تعديل.
فصل بين الكود البرمجي الخاص بالمنظومة والكود الخاص المتعلق بال framework	يمكن نقل الكود البرمجي الى غير framework بدون الحاجة للتغيير
يجب على النظام أن يستخدم تصاميم (design patterns) تدعم توزيع العمل بين أعضاء المجموعة	استخدام صحيح لل design patterns وعدم وجود أخطاء تصميمية.
استخدام APIs من أجل ربط طبقة الواجهة بطبقة النظام	طريقة الوصول للنظام تتم من خلال APIs

استخدام اداة ORM وعدم الدخول مباشرة لقاعدة البيانات	وجود models بالنظام معكوسة بشكل صحيحة على Database engine
تحقيق transaction على مستوى النظام	تحقيق transaction ضمن المتطلبات الوظيفية
Authentication	تحقيق المصادقة ضمن Middleware) Aspect)
Authorization	تأمين التحقق من التفويض ضمن Middleware منفصلة
المتطلبات الوظيفية	
User register	إضافة مستخدم جديد للنظام عن طريق كتابة معلوماته الأساسية مثل: ايميل والاسم الكامل واسم المستخدم وكلمة مرور (يمكنك اضافة ماتشاء). يجب عدم وجود مستخدمين بنفس الايميل او الاسم المستخدم
User Login	توفير ميزة تسجيل الدخول عن طريق اسم المستخدم وكلمة المرور
إنشاء أو حذف ملف	المستخدم يستطيع رفع ملف جديد إلى النظام (بحالة حر) او حذف ملف (في حالة كان الملف حر) يجب التأكد من عدم وجود تنافس على الملف في حالة الانشاء او الحذف
إنشاء أو حذف مجموعات لمشاركة الملفات	يستطيع المستخدم إنشاء مجموعة (لها اسم خاص) بحيث: ● يمكن اضافة ملف الى المجموعة (فقط في حالة كان المالك للملف) ● يمكن حذف ملف من المجموعة (فقط في حالة كان المالك للملف) ● إضافة مستخدمين آخرين للمجموعة ● حذف مستخدمين من المجموعة (في حالة كان المستخدم غير حاجز لأي ملف بالمجموعة) ● حذف مجموعة في حالة لا يوجد ملفات محجوزة فيها من قبل المستخدمين كذلك يحوي النظام على مجموعة خاصة استثنائية تسمى public بحيث يستطيع أي مستخدم إضافة ملفاته عليها ليتمكن أي مستخدم في النظام من الوصول للملف (Check-in و Check out)
عرض الملفات والمجموعات المنشأة	● الملفات المرفوعة من المستخدم ● عرض حالة الملف (حر او محجوز) ● عرض اسم المستخدم الذي حجز الملف في حال كان محجوز ● عرض المجموعات التي يملكها ● عرض الملفات في كل مجموعة (بنفس طريقة العرض السابقة)

العمليات على الملفات	قراءة ملف: يمكن للمستخدم قراءة الملف بدون تغيير في الحالة حجز الملف Check-in: المستخدم يقوم بعملية الحجز للملف بحيث لا يمكن لأي مستخدم آخر من القراءة أو الكتابة على الملف المحجوز (فقط في حالة كان الملف حر) تعديل: يمكن للمستخدم (الذي حجز الملف) من التعديل على الملف المحجوز الانتهاء من الحجز او Check-out: إلغاء الحجز عن الملف ليعود إلى الحالة الحر عملية حجز جماعية bulk-check-in: بحيث يمكن للمستخدم تحديد اكثر من ملف وحجز جميعهم في ان واحد او تفشل العملية. يحب استخدام Caching من أجل اختزال زمن عرض الملفات الموجودة في أي مجموعة. المقصود هنا توفير زمن الوصول لل Database من أجل معرفة الملفات الموجودة ضمن المجموعة. ملاحظة: المنظومة يجب أن تستخدم File Systems الخاص بالنظام من أجل تخزين الملفات وإدارتها
التقارير	يمكن للمستخدم تصدير التقارير التالية: • تقرير يحوي على history للأحداث كاملة عن ملف ما بحيث يحوي التقرير (تاريخ رفع الملف وتاريخ حجز الملف الذي حجزه بالنظام وتاريخ التعديل وتاريخ إلغاء الحجز) مرتب من الحدث الاحدث الى الاقدم وتحوي اسم المستخدم عند كل حدث
Admin user	• يحوي النظام على admin user بحيث يملك صلاحية إضافية لعرض جميع الملفات بالنظام والمجموعات
ملاحظات	• المجموعة لا تحتوي على مجموعة
ميزات اخرى (اختيارية)	أمثلة: • البحث على ملف • البحث ضمن الملفات
المتطلبات غير الوظيفية	
(اجبارية) Logging	تسجيل جميع Request الواردة و ال Response الصادرة من النظام المرتبطة بال Request السابقة. يجب التأكد من أن Logging سوف يستخدم نفس المناقلة الخاصة بالخدمات من أجل الوصول لـ Database وتخزين المعلومات.

Configuration (اجبارية)	يجب على النظام ان يوفر إمكانية تغيير بعض القيم مثل: • دعم Multi database engine بحيث يمكن تغيير ال Engine من إعدادات النظام. • خيارات تغيير إعدادات الاتصال مع Database (عنوان database او port أو credential) • دعم خيارات للتحكم بال Loglevel • خيارات خاصة بالنظام (مثل: عدد الملفات الأكثر لكل مستخدم)
Performance (اجبارية)	يجب على النظام أي يستطيع تخديم 100 مستخدم على الأقل في آن واحد. يجب إثبات ذلك من خلال استخدام (Apache Jmeter او load test application) تصدير تقرير او اي اثبات أن النظام نجح في تجاوز الشرط السابق.
Scalability	يجب على النظام أن يدعم التصعيدية: • تشغيل Load balancing بين 2 instances أو أكثر من النظام • وجود check health api من أجل فحص حالة كل instance في النظام في حالة استخدام Caching على مستوى المجموعات: يجب التأكد من عدم حصول inconsistency لل cache بين instances
Security and optimization features	• استخدام reverse proxy لاستقبال الطلبات وتوجيها • تسجيل IP المستخدمين • دعم برتوكول http2 • دعم https • دعم api ل check health بال applications • دعم response compression في حال كانت أكبر من 1 ميغابايت • توفير rate limiting لعدد الطلبات المسموحة من مصدر ما (من خلال aspect أو بال reverse proxy) • تحديد حجم request الاعظمي إلى حد معين (مثل 10 ميغابايت) في حالة استخدام aspect لتطوير rate limiting يجب الانتباه لحالة التغير الحاصل بمصدر الطلبات

المطلوب:

- تحقيق جميع المتطلبات البنيوية والمتطلبات الوظيفية والطلبات غير الوظيفية (الإجبارية):
  - عدد الطلاب 3 او اقل: فقط المتطلبات الإجبارية
  - عدد الطلاب 4: تحقيق متطلب غير وظيفي إضافي
  - عدد الطلاب 5: تحقيق متطلبين غير وظيفيين إضافيين