

اسم النظام: نظام إدارة مكتبة.

التاريخ: 15/6/2025.

أعضاء الفريق:

1. عتاب محمد ونوس

2. جوى محمد اسماعيل

3. غزل ثابت سليمان

4. ليث علي مرعي

5. كرم محسن غانم

1. مقدمة:

1.1. الغرض:

تهدف هذه الوثيقة إلى تحديد وفهم المتطلبات الوظيفية وغير الوظيفية لنظام إدارة مكتبة، وتستخدم كمرجع واحد وشامل خلال مراحل التحليل، التصميم، البرمجة والاختبار. كما تضمن تطوير نظام يلبي احتياجات المستخدمين ويساعد في تنظيم العمل داخل المكتبة.

2.1. الجمهور المستهدف:

هذه الوثيقة موجهة إلى:

- لفريق التطوير Developers
- فريق الاختبار Testers
- أصحاب المصلحة Stakeholder
- أمين المكتبة Librarian
- مدير النظام System Administration
- المستخدمون Users

3.1. الاستخدام المقصود:

تمنح الوثيقة صورة شاملة للنظام بأكمله ومصدر واحد لتوضيح متطلبات النظام لجميع الأطراف المعنية. وبذلك تضمن تلبية جميع المتطلبات الوظيفية وغير الوظيفية، وتنسيق الجهود بين فرق التطوير والاختبار

4.1. نطاق المنتج:

نظام إدارة المكتبة يتيح للمستخدمين (الزبائن):

1. البحث عن الكتب إلكترونياً
2. استعارة الكتب وتتبع حالاتها
3. المشاركة في فعاليات المكتبة
4. المشاركة في فعاليات المكتبة

5. استلام إشعارات (Notifications) في حال التأخر عن إعادة الكتب

بينما يتيح للمسؤولين (أمين المكتبة ومدير النظام):

1. إدارة عمليات الاستعارة والإرجاع

2. متابعة الحسابات

3. إضافة أو تعديل الكتب والفعاليات

4. إرسال إشعارات في حال تأخر الزبون بإعادة الكتاب

5.1. التعاريف والاختصارات:

المصطلح	المعنى
UML	Unified modelling language اللغة الموحدة للنمذجة
UI	واجهة مستخدم
MySQL	نظام إدارة قواعد بيانات مفتوح المصدر لتخزين البيانات واسترجاعها باستخدام لغة SQL
API	واجهة برمجة التطبيقات
Flutter	إطار عمل مفتوح المصدر يستخدم لإنشاء تطبيقات ويب، موبايل و سطح مكتب (تصميم واجهات أمامية)
laravel	إطار عمل مفتوح المصدر لتصميم واجهات خلفية (التعامل مع قواعد البيانات)
Use Case Diagram	مخطط حالات الاستخدام
Class diagram	مخطط الصفوف
ERD	مخطط الكيانات

2. الوصف العام:

1.1.. احتياجات المستخدم:

يحتاج المستخدمون إلى واجهة بسيطة وسهلة تمكنهم من:

- البحث عن الكتب بسهولة
- استعارة الكتب إلكترونياً

كما يحتاج موظفو المكتبة إلى أدوات لإدارة الكتب تشمل:

- إضافة الكتب الجديدة
- تعديل بيانات الكتب
- حذف المستعار منها

2.1. الافتراضات والتبعيات:

يعمل النظام ضمن بيئة تعتمد على:

- الاتصال دائم بالإنترنت
- قاعدة بيانات MySQL لتخزين البيانات
- تطوير واجهة أمامية باستخدام Flutter
- تطوير واجهة خلفية باستخدام laravel php
- الربط بين الواجهة الأمامية والخلفية باستخدام API

3. مميزات النظام ومتطلباته:

1.3. المتطلبات الوظيفية:

➤ للمستخدم: User

1. تسجيل الدخول إذا كان مستخدم قديم، أو إنشاء حساب إذا كان مستخدم لأول مرة.
2. البحث عن كتاب عن طريق اسمه أو اسم مؤلفه أو حتى نوعه.
3. استعارة كتاب إذا كان متوافر.
4. إرجاع الكتاب قبل انتهاء مدة الاستعارة.
5. عرض قائمة بالكتب المستعارة مع تاريخ كل منها.

6. تحديد مدّة الاستعارة أو طلب تمديدّها.
7. حجز موعد في فعاليات المكتبة.
8. الدّفع قبل كلّ عملية استعارة.

➤ **لنظام المكتبة: Library System**

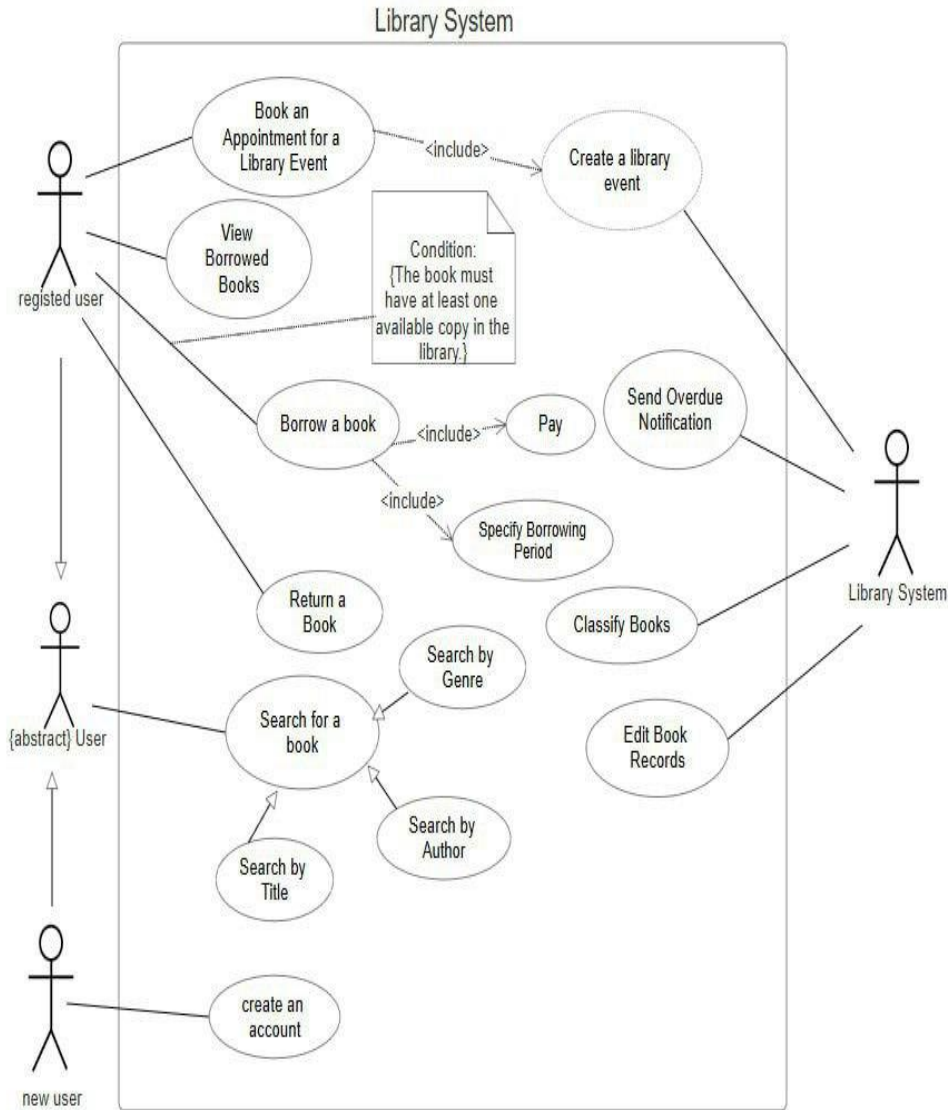
1. تصنيف الكتب حسب النوع أو المؤلف لتسهيل البحث.
2. تعديل سجلات الكتب بعد كلّ عملية استعارة أو إرجاع كتاب.
3. إرسال إشعار للمستخدم لتذكيره بتجاوز مدة الاستعارة.
4. تنظيم فعاليات في المكتبة.

2.3. المتطلبات غير الوظيفية:

1. يجب أن يكون النظام متاحاً 99 % على مدار الساعة طوال أيام الأسبوع.
2. يدعم النظام الأجهزة المحمولة iOS. /Android
3. زمن الاستجابة قصير.
4. واجهة استخدام سهلة تناسب جميع الأعمار وفئات الناس.
5. الموثوقية والأمان عند الدّفع.
6. يدعم النظام اللغة الانكليزية والعربية.

سنعرض الآن بعض مخططات UML لفهم النظام بشكل أوضح وأعمق:

مخطط حالات الاستخدام : Use case Diagram



الشكل (1): مخطط حالات الاستخدام

نلاحظ من المخطط السابق وجود الفاعل المجرد **{abstract} User** الذي يمكنه فتح التطبيق والبحث عن الكتاب بثلاث طرق ممكنة بحيث إذا وجد الكتاب الذي يبحث عنه وكان مستخدم جديد **new user** يقوم بتسجيل الدخول إلى النظام ويصبح بعدها مستخدم مسجل **registered user** يستطيع الاستفادة من خدمات النظام كأن يشارك في فعاليات المكتبة إن وجدت ويستعير كتاب متوافر بعد أن يحدد المدة المناسبة له ويدفع وأخيراً يمكنه عرض لائحة كتبه التي سبق واستعارها وإعادة ما انتهى من قراءته قبل انتهاء المدة التي حددها سابقاً.

أما الآن سنعرض الآن جداول توصيف حالات الاستخدام التي تحدثنا عنها سابقاً:

Name الاسم	View borrowed books
Short description وصف موجز	يطلب مستخدم مسجل عرض قائمة كتبه المستعارة
Postcondition الشروط السابقة	يجب أن يكون المستخدم مسجلاً مسبقاً في النظام
Precondition الشروط اللاحقة	يقوم النظام بعرض قائمة بجميع الكتب المستعارة أو يعرض قائمة فارغة في حال عدم وجود كتب
Error situation حالات الخطأ	(1) أن يكون الطالب غير مسجل (2) فشل الاتصال بقاعدة البيانات
System state in the event of error حالة النظام في حال حدوث خطأ	لا يتم عرض قائمة الكتب الخاصة بالمستخدم
Actors جماعات فاعلة	Registered user
Trigger المحفز	رغبة المستخدم بمراجعة قائمة كتبه التي استعارها والتحقق من التاريخ
Stander Process العمليات القياسية	(1) النقر على أيقونة " Borrowed books" (2) يتحقق النظام إذا كان المستخدم مسجلاً سابقاً (3) يرسل طلب إلى قواعد البيانات لاستعادة الكتب المستعارة من قبل هذا المستخدم (4) عرض الكتب للمستخدم
Alternative Processes إجراءات بديلة	(1) إذا كان المستخدم غير مسجل يطلب النظام تسجيل الدخول أولاً (2) إذا حدث فشل في الخادم يطلب إعادة المحاولة مرة أخرى

Name الاسم	Return a book
Short description وصف موجز	يقوم المستخدم المسجل بإعادة الكتاب الذي استعاره قبل انتهاء المدة المحددة.
Postcondition الشروط السابقة	يجب أن يكون المستخدم مسجلاً مسبقاً وقام باستعارة أحد الكتب ولم يتجاوز المدة المسموح بها.
Precondition الشروط اللاحقة	يقوم النظام بتعديل عدد النسخ المتوافرة من الكتاب وإزالته من قائمة الكتب المستعارة عند المستخدم
Error situation حالات الخطأ	فشل الاتصال بقاعدة البيانات
System state in the event of error حالة النظام في حال حدوث خطأ	لا يتم إرجاع الكتاب وبقائه في قائمة الكتب المستعارة
Actors جماعات فاعلة	Registered user
Trigger المحفز	انتهاء المستخدم من قراءة الكتاب والرغبة في إعادته قبل انتهاء المدة
Stander Process العمليات القياسية	(1) النقر على أيقونة " return this book" (2) يتحقق النظام إذا كان التاريخ ضمن الفترة المسموح بها (3) يرسل طلب إلى قواعد البيانات لتعديل بيانات الكتاب والمستخدم (4) إرجاع الكتاب بنجاح
Alternative Processes إجراءات بديلة	يطلب النظام من المستخدم إعادة المحاولة مرة أخرى وإذا استمر العطل فترة طويلة يتم تمديد مدة استعارة الكتاب

Name الاسم	Search for a book
Short description وصف موجز	يقوم المستخدم سواء كان مسجلاً مسبقاً أو جديد بالبحث عن الكتاب الذي يريده.
Postcondition الشروط السابقة	يجب أن يعرف المستخدم اسم الكتاب أو اسم المؤلف أو حتى نوعه.
Precondition الشروط اللاحقة	إذا أدخل المستخدم اسم الكتاب وكان متوفر يتم عرضه إذا أدخل اسم المؤلف يتم عرض جميع أعماله وإذا أدخل نوع الكتاب يتم اقتراح كتب شبيهة به
Error situation حالات الخطأ	(1) الكتاب غير متوفر حالياً (2) لم يتم التعرف على الاسم المدخل
System state in the event of error حالة النظام في حال حدوث خطأ	(1) عرض الكتاب ولكن عدد النسخ صفر لا يمكن الاستعارة (2) عرض لائحة خالية
Actors جماعات فاعلة	user
Trigger المحفز	رغبة المستخدم في البحث عن كتاب محدد والضغط على مربع البحث "search"
Stander Process العمليات القياسية	(1) فتح التطبيق والضغط على مربع البحث وكتابة الاسم (2) يقوم النظام بالتحقق من البيانات المدخلة (3) يعرض النظام قائمة موافقة لطلب المستخدم
Alternative Processes إجراءات بديلة	يطلب النظام من المستخدم إدخال اسم صالح أو يحدد له التاريخ الذي سيتوافر فيه الكتاب مرة أخرى

Name الاسم	Create an account
Short description وصف موجز	يقوم المستخدم الجديد بإنشاء حساب على التطبيق.
Postcondition الشروط السابقة	المستخدم لا يملك حساب مسبقاً.
Precondition الشروط اللاحقة	يقوم النظام بحفظ بيانات المستخدم ثم يفعل له الخدمات المقدمة للمستخدمين المسجلين.
Error situation حالات الخطأ	(3) اسم المستخدم غير صالح. (4) البريد الإلكتروني غير موجود
System state in the event of error حالة النظام في حال حدوث خطأ	إظهار رسالة خطأ ولا يتم إنشاء الحساب.
Actors جماعات فاعلة	new user
Trigger المحفز	رغبة المستخدم باستعارة كتاب متوافر على التطبيق ولكنه لا يملك حساب بعد.
Stander Process العمليات القياسية	(1) فتح التطبيق والضغط على زر "create account" (2) يدخل المستخدم اسمه وبيده الإلكتروني (3) يقوم النظام بالتحقق من صحة البيانات المدخلة ثم يضيفها إلى قواعد البيانات (4) يتم إنشاء الحساب بنجاح
Alternative Processes إجراءات بديلة	إذا كان الاسم غير صالح يقترح النظام اسم مناسب آخر وإذا كان البريد خاطئ يمكن للمستخدم تسجيل الدخول باستخدام حساب google أو facebook

Name الاسم	Pay
Short description وصف موجز	يقوم المستخدم المسجل بالدفع قبل عملية استعارة أي كتاب
Postcondition الشروط السابقة	المستخدم لديه حساب دفع كاش قام المستخدم بالبحث عن الكتاب المطلوب ووجده متوفر
Precondition الشروط اللاحقة	يقوم النظام بسحب المبلغ من حساب الشخص وإضافته إلى حساب التطبيق
Error situation حالات الخطأ	المبلغ الموجود في حساب المستخدم أقل من سعر الكتاب
System state in the event of error حالة النظام في حال حدوث خطأ	لا يتم الدفع ولا يتم إكمال عملية الاستعارة يظهر رسالة خطأ للمستخدم ليقوم بشحن رصيده
Actors جماعات فاعلة	Registered user
Trigger المحفز	رغبة المستخدم باستعارة كتاب والضغط على زر الدفع
Stander Process العمليات القياسية	(1) يدخل المستخدم رقم حسابه (2) يقوم النظام بالتحقق من أن الرصيد أكبر أو يساوي سعر الكتاب (3) تتم عملية الدفع
Alternative Processes إجراءات بديلة	يتم عرض للمستخدم كتب مجانية بديلة عن التي طلبها في حال كان رصيده لا يكفي أو لحظة سريعة عن الكتاب المطلوب

Name الاسم	Edit book records
Short description وصف موجز	يقوم أمين المكتبة بتحديث عدد نسخ الكتب بعد كل عملية استعارة أو إرجاع.
Postcondition الشروط السابقة	قام الزبون بعملية استعارة أو إرجاع بالإضافة إلى أن المستخدم الذي يقوم بالتعديل يملك صلاحيات النظام
Precondition الشروط اللاحقة	عدد النسخ المكتوبة الظاهرة للمستخدم في واجهة التطبيق توافق عدد النسخ الفعلية
Error situation حالات الخطأ	حدث خطأ أثناء الاتصال بالخادم
	عدد النسخ لم تتحدث
Actors جماعات فاعلة	Library System
Trigger المحفز	حدث عملية استعارة أو إرجاع لأحد الكتب
Stander Process العمليات القياسية	(1) يدخل أمين المكتبة إلى التطبيق (2) يضغط على زر تعديل معلومات الكتاب الذي تم استعارته ويعتدّل عدد النسخ. (3) يضغط على تأكيد التعديل مع الانتباه لوجود الانترنت لتجنب حدوث أخطاء.
Alternative Processes إجراءات بديلة	إرسال اشعار تنبيه بحدوث خطأ بالاتصال وطلب إعادة المحاولة مرة أخرى بعد التحقق من وجود الاتصال.

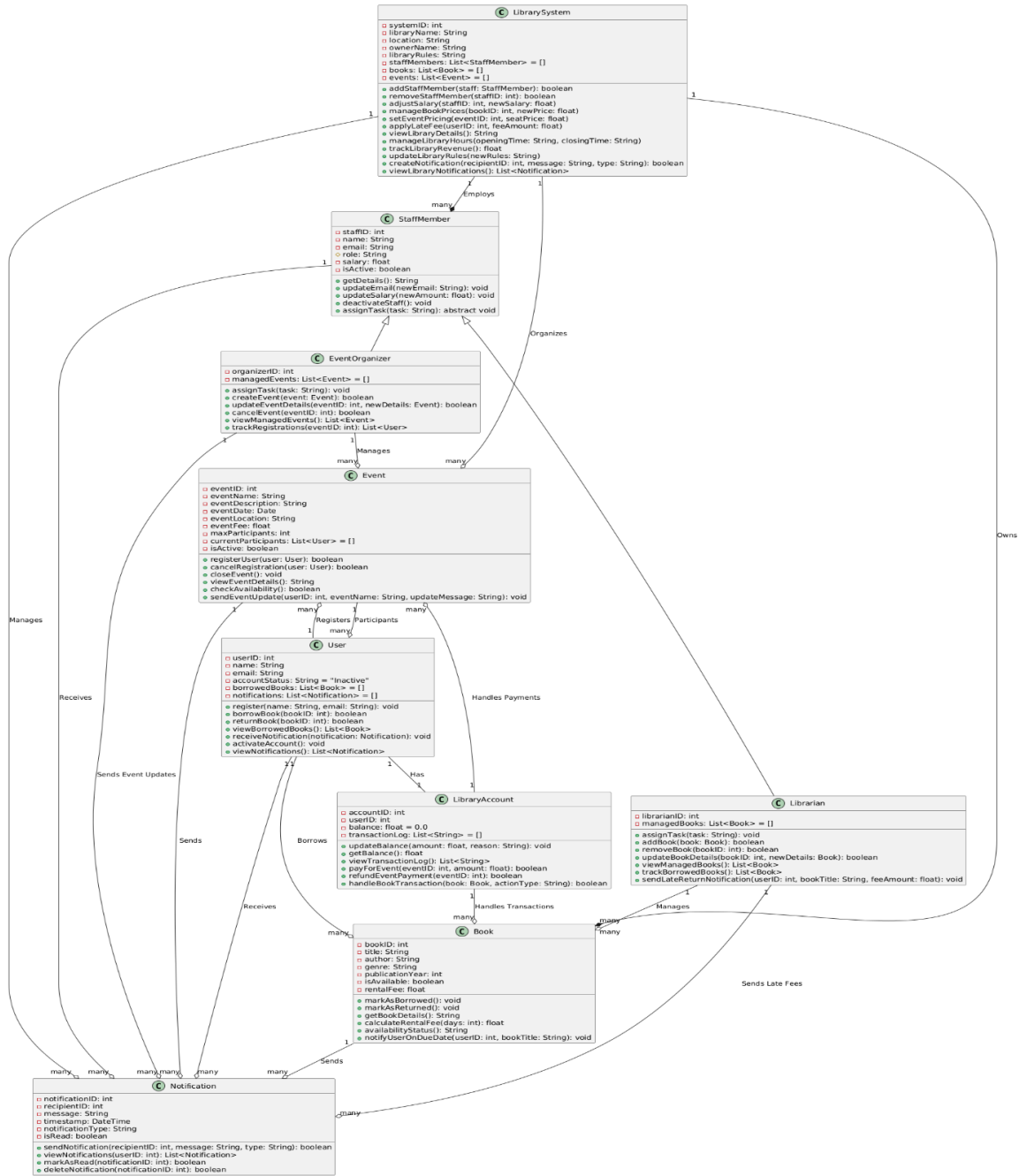
Name الاسم	Classify Books
Short description وصف موجز	تصنيف الكتب وتنظيمها وفقاً لموضوعاتها أو حسب المؤلفين
Postcondition الشروط السابقة	أن يكون المستخدم يملك صلاحيات النظام للتعديل على قواعد البيانات
Precondition الشروط اللاحقة	يقوم التطبيق بعرض الكتب مصنفة حسب رغبة المستخدم لتسهيل عملية البحث
Error situation حالات الخطأ	وضع كتب غير متناسقة في نفس التصنيف
System state in the event of error حالة النظام في حال حدوث خطأ	الكتب مصنفة بشكل خاطئ وصعوبة في إيجاد الكتاب المطلوب
Actors جماعات فاعلة	Library System
Trigger المحفز	الرغبة في الحصول على نظام مكتبة سهل ومنظم
Stander Process العمليات القياسية	(1) الدخول إلى تطبيق المكتبة (2) إنشاء مجلدات خاصة بكل مؤلف (حالة التصنيف حسب المؤلف) (3) وجع جميع كتب المؤلف في مجلد (4) ضغط تأكيد التصنيف
Alternative Processes إجراءات بديلة	اقتراح تصنيف تلقائي من النظام حسب الأحرف الأبجدية.

Name الاسم	Send overdue Notification
Short description وصف موجز	يرسل أمين المكتبة إشعارات تنبيهية للزبون عند تأخره بإرجاع الكتاب
Postcondition الشروط السابقة	تجاوز المستخدم مدة استعارة الكتاب المسموح بها
Precondition الشروط اللاحقة	إرسال إشعار للمستخدم وبدء احتساب غرامة على التأخير تحديث حالة الكتاب "بانتظار الإرجاع"
Error situation حالات الخطأ	فشل بإرسال الإشعار
System state in the event of error حالة النظام في حال حدوث خطأ	تنبيه أمين سر المكتبة بإعادة إرسال التنبيه للمستخدم
Actors جماعات فاعلة	Library System
Trigger المحفز	تأخر الزبون بإرجاع الكتاب
Stander Process العمليات القياسية	(1) يقوم أمين السرّ بالدخول إلى البريد الإلكتروني الخاص بالزبون (2) يرسل إشعار تنبيهية يتضمن التاريخ الذي يجب إعادة الكتاب فيه (3) يتأكد من وصول الإشعار للمستخدم
Alternative Processes إجراءات بديلة	إرسال إشعار مرة أخرى للزبون أو استخدام وسائل أخرى كرقم الموبايل

Name الاسم	Specify Borrowing period
Short description وصف موجز	يقوم الزبون بتحديد المدة الزمنية التي يحتاجها لإنهاء الكتاب
Postcondition الشروط السابقة	استعارة الكتاب
Precondition الشروط اللاحقة	تحديث حالة لكتاب من "متوفر" إلى "مستعار" وتسجيل المدة المحددة في قواعد البيانات
Error situation حالات الخطأ	(1) تحديد مدة استعارة تتجاوز الحد المسموح (2) عدم انتهاء المستخدم من الكتاب في المدة الزمنية المسموحة.
System state in the event of error حالة النظام في حال حدوث خطأ	(1) يفرض النظام على المستخدم الحد الأقصى المسموح فقط (2) يرسل تنبيهات للمستخدم بتجاوز المهلة
Actors جماعات فاعلة	Registered user
Trigger المحفز	رغبة المستخدم باستعارة كتاب
Stander Process العمليات القياسية	(1) يدخل المستخدم إلى تطبيق المكتبة (2) يحدد الكتاب الذي يريده (3) يحدد المدة الزمنية التي يحتاجها (4) يقوم بإرسال الطلب
Alternative Processes اجراءات بديلة	تحديد المدة تلقائياً من النظام حسب نوع الكتاب.

مخطط الصفوف Class Diagram:

يوضح هذا المخطط الكائنات (Classes) الرئيسية التي يتكوّن منها نظام إدارة المكتبة، إضافةً إلى الخصائص والوظائف المرتبطة بها، والعلاقات التي تربط الكائنات ببعضها. يساعد هذا المخطط على تصور الهيكل العام للنظام قبل البدء في البرمجة.



شرح مفصّل عن المخطط السّابق:

• صفّ User يملك العديد من الوظائف منها:

- ✓ Register: دالة لتسجيل المستخدم.
 - ✓ BorrowBook: يسمح للمستخدم باستعارة كتاب ويعد true في حال نجاح العملية و false عكس ذلك.
 - ✓ ReturnBook: يسمح للمستخدم بإرجاع كتاب ويرد قيمة بوليانية كالدّالة السابقة.
 - ✓ ViewBorrowedBooks: يعرض قائمة الكتب التي استعارها المستخدم.
 - ✓ ReceiveNotification: إضافة إشعار إلى قائمة الإشعارات الخاصّة بالمستخدم.
 - ✓ ActivateAccount: تحويل الحساب إلى حالة نشط active
 - ✓ RegisterFoorEvent: للتسجيل في فعالية
 - ✓ CancelEventRegistration: لإلغاء الحجز
 - ✓ ViewRegisterdEvents: لعرض قائمة الفعاليات المسجل بها
 - ✓ ViewNotifications: عرض الإشعارات الخاصّة بالمستخدم
- أمّا بالنسبة للعلاقات:

1. User مرتبط مع libraryAccount
2. User مرتبط مع Book بعلاقة borrowedBooks من النوع M:1
3. User مرتبط مع Event عبر الفعاليات من النوع M:M
4. User مرتبط مع Notification لاستقبال التنبيهات بعلاقة من النوع M:1

● **صف StaffMember هو صف مجرد abstract فيه الوظائف:**

- ✓ GetDetail: لعرض بيانات الموظف
- ✓ UpdateEmail: تعديل البريد الإلكتروني الخاص بالموظف
- ✓ UpdateSalary: لتعديل راتب الموظف
- ✓ DeactivateStaff: يجعل الموظف غير نشط (لم يعد يعمل في المكتبة)
- ✓ AssignTask: طريقة مجردة لتحديد المهام الوظيفية للموظف

أما بالنسبة للعلاقات:

1. علاقة تجميع aggregation بين StaffMember و LibrarySystem حيث إذا تم حذف نظام المكتبة لا يتم حذف الموظفين
2. يرث هذا الصف كل من Librarian , EventOrganizer

● **الصف Librarian أمين المكتبة له الوظائف التالية:**

- ✓ AssignTask: تعيين مهام لأمين المكتبة
- ✓ AddBook: إضافة كتاب جديد إلى قائمة الكتب التي يريدها الموظف
- ✓ RemoveBook: حذف كتاب من القائمة (إذا كان موجود)
- ✓ UpdateBookDetails: تعديل بيانات كتاب
- ✓ ViewManagedBooks: عرض قائمة الكتب التي يريدها هذا الموظف
- ✓ TrackBorrowedBook: عرض قائمة جميع الكتب المستعارة مع بيانات المستخدمين الذين استعاروها

والعلاقات:

1. يرتبط Librarian مع Book بعلاقة من النوع 1:M
2. يرتبط Librarian مع StaffMember بعلاقة وراثية

● **الصف EventOrganizer منظم الفعاليات يقوم بالوظائف التالية:**

- ✓ AssignTask: تعيين مهام لمنظم الفعاليات
- ✓ CreateEvent: إنشاء فعالية داخل المكتبة
- ✓ UpdateEventDetails: تعديل تفاصيل الفعالية (قبل أن تبدأ)
- ✓ CancelEvent: إلغاء الفعالية
- ✓ ViewManagedEvents: عرض قائمة الفعاليات التي ينظمها هذا المنظم
- ✓ TrackRegistrations: عرض قائمة المستخدمين المسجلين بالفعالية
- ✓ SendLateReturnNotification: إرسال إشعار للمستخدم يعلمه بالغرامة

العلاقات:

1. يرتبط EventOrganizer مع Event بعلاقة من النوع M:1
2. يرث EventOrganizer من ألف الأب StaffMember

● **الصف LibraryAccount يحوي الوظائف التالية:**

- ✓ UpadteBalance: التعديل على رصيد لحساب
- ✓ GetBalance: عرض رصيد الحساب
- ✓ ViewTransactionLog: عرض العمليات المالية
- ✓ PayForEvent: الدفع من أجل التسجيل بالفعالية
- ✓ ReturnEventPayment: استعادة المبلغ في حال إلقاء الحجز
- ✓ HandleBookTransaction: التعديل على الرصيد في حال الاستعارة أو إلغاء الاستعارة

العلاقات:

1. يرتبط libraryAccount مع User بعلاقة من النوع 1:1

• الصف Book فيه الوظائف التالية:

- ✓ MarkAsBorrowed: يقول بالتعديل isAvailable=false أيّ النسخة غير متاحة
- ✓ MarkAsReturned: يقول بالتعديل isAvailable=true أيّ النسخة متاحة
- ✓ GetBookDetails: عرض تفاصيل الكتاب
- ✓ CalculateRentalFee: يحسب تكلفة استعارة الكتاب بناءً على عدد الأيام (عدد الأيام * تكلفة الاستعارة ليوم واحد)
- ✓ AvailabiltyStatus: عرض حالة الكتاب في حال كان متوقّر أم لا
- ✓ NotifyUserData: إرسال إشعار تلقائي للمستخدم قبل بيوم من انتهاء المهلة المحددة لتسليم المشروع

أمّا العلاقات:

1. علاقة تركيب بين librarySystem و Book يعني إذا تدمّر نظام المكتبة يتدمّر جدول الكتب (الكتب جزء من النظام)
2. علاقة M:1 بين User ,Book
3. علاقة M:1 بين librarian ,Book (أمين المكتبة مسؤول عن إدارة الكتب داخل المكتبة مثل الحذف والإضافة والتّعديل)
- 4.

• الصف Event له الوظائف :

- ✓ RegisterUser: تسجيل المستخدم بالفعاليّة ويرد true في حال نجاح العملية، و false في حال الفشل
- ✓ CancelRegistration: إلغاء الحجز
- ✓ ViewEventDetails: عرض تفاصيل الفعالية
- ✓ CheckAvailability: التّحقق إذا الفعالية فيها مقاعد متاحة
- ✓ SendEventUpdate: إرسال إشعار للمستخدمين في حال التّعديل على تفاصيل الفعاليّة.

أما العلاقات:

1. العلاقة بين User و Event هي علاقة متعددة إلى متعدد M:M
2. العلاقة بين Event و EventOrganizer هي علاقة 1:1 حيث الفعالية تنظم من قبل شخص واحد، وهذا الشخص يمكنه تنظيم فعالية واحدة
3. العلاقة بين Event و librarySystem هي علاقة تجميع (يتم إدارة الفعاليات ضمن النظام، ولكنها ليست جزء متعلق بالنظام)

• الصف Notification الإشعارات يحوي الوظائف التالية:

- ✓ SendNotification: إنشاء إشعار وإرساله إلى المستلم المحدد (مستخدم، موظف)
- ✓ ViewNotification: يسمح للمستخدم بعرض الإشعارات الخاصة به
- ✓ MarkAsRead: تحديد الرسالة كمقروءة
- ✓ DeleteNotification: حذف الإشعار من النظام بشرط أن يكون المستلم قد قرأه

والعلاقات :

1. يرتبط صف Notification مع LibrarySystem بعلاقة 1:M

حيث يدير النظام عدة إشعارات

2. يرتبط صف Notification مع User و الموظفين بعلاقة 1:M

3. يرتبط مع الكتب والفعاليات بعلاقة 1:M (يمكن أن يكون هناك عدة

إشعارات مرتبطة بكتاب أو فعالية ما).

• وأخيراً الصّف **LibrarySystem** له الوظائف التّالية:

- ✓ AddStaffMember: يضيف موظف جديد إلى المكتبة
- ✓ RemoveStaffMember: يحذف موظف من المكتبة إذا كان موجود
- ✓ AdjustSalary: تعديل راتب موظّف
- ✓ ManageBookPrices: تعديل سعر استعارة كتاب
- ✓ SetEventPricing: تحديد سعر المقعد في فعالية ما
- ✓ ViewLibraryDetails: عرض معلومات المكتبة
- ✓ TrackLibraryRevnuue: إجمالي إيرادات داخل المكتبة (المبيعات، الفعاليات، غرامات)
- ✓ UpdateLibraryRules: تعديل قواعد المكتبة
- ✓ CreateNotification: إنشاء إشعار ضمن النظام ولكن إرساله يكون ضمن جدول Notification
- ✓ ViewLibraryNotification: عرض الإشعارات التي هي تحت إدارة المكتبة

مخطط الكيانات ERD

يهدف هذا المخطط إلى توضيح التصميم المفاهيمي لقاعدة البيانات الخاصة بالنظام حيث يوضح الكيانات الرئيسية، صفاتها، والعلاقات التي تربط بينها

