# تحليل نظام الإدارة المالية والإدارية لفندق متكامل الخدمات

# 

* الهدف من المشروع :

يهدف النظام الى تحسين الكفاءة المالية والتشغيلية للفندق من خلال أتمتة العمليات اليومية مثل حجز الغرف والفوترة والتقارير المالية ويهدف الى منح الزبون سهولة في التعامل مع الفندق واجراء العمليات ك حجز غرفة ودفع الرسوم بسرعة وسهولة

* الجهات الفاعلة :
* الضيف (Guest) | شخص خارجي يقوم بالحجز أو الدفع أو طلب خدمات
* موظف الاستقبال (Receptionist) مسؤول عن إدخال الحجز وإصدار الفواتير واستلام المدفوعات
* المحاسب (Accountant) يدير المصروفات اليومية ويصدر التقارير المالية
* مدير الفندق (Manager) يطّلع على مؤشرات الأداء ويقرّر بناءً عليها
* علاقة كل ما سبق بتحليل النظام
* زيادة الإيرادات :من خلال إعادة تدوير الغرف بشكل أفضل
* خفض التكاليف: من خلال تقليل الإعتماد على الورق والأخطاء البشرية
* دقة في الحسابات :من خلال تسجيل الي لكل المعاملات المالية
* المتطلبات الوظيفية
* إدارة الحجوزات: تسجيل الحجوزات، التحقق من توافر الغرف، التعديل أو الإلغاء.
* إدارة الغرف: تصنيف الغرف حسب النوع، الحالة (متاحة/محجوزة/تحت الصيانة).
* تسجيل الوصول والمغادرة: إجراءات Check-in وCheck-out مع تحديث الحالة تلقائيًا.
* إدارة العملاء: تسجيل بيانات النزلاء، حفظ سجل الإقامات والتفضيلات.
* الدفع والفوترة: إصدار الفواتير، دمج طرق الدفع (نقدًا، بطاقة، تحويل بنكي).
* التقارير: إحصائيات عن نسبة الإشغال، العائدات، تقرير يومي للغرف.
* إدارة الموظفين: صلاحيات متعددة (استقبال، إدارة، تدبير فندقي).
* المتطلبات غير الوظيفية
* سهولة الاستخدام: واجهة مستخدم بسيطة تناسب الموظفين من مختلف الخلفيات.
* الأمان: حماية بيانات العملاء والمعاملات بتشفير وأذونات وصول.
* الاعتمادية: أن يعمل النظام بثبات على مدار الساعة بدون أعطال متكررة.
* الأداء: سرعة في تنفيذ المهام حتى مع حجم بيانات كبير.
* التوافق: أن يعمل على مختلف الأجهزة (كمبيوتر، تابلت) وربما كـتطبيق هاتف.
* الدعم الفني والصيانة: إمكانية التحديث الدوري والدعم عند الحاجة.

# نطاق النظام : النظام سيغطي

# إدارة الغرف والحجوزات.

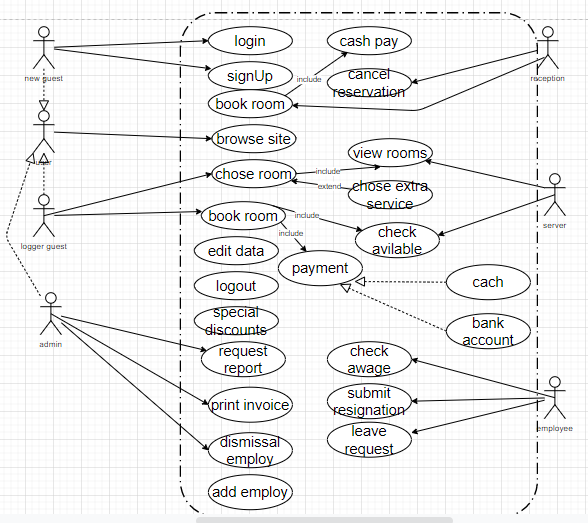
# التعاملات المالية (فواتير، مدفوعات، نفقات).

# إدارة الخدمات الإضافية.

# تقارير فورية ودورية للإدارة.

# صلاحيات متعددة حسب نوع المستخدم.

* مخطط حالات الاستخدام



* توصيف حالتي استخدام
* حالة الاستخدام الأولى:طلب خدمة إضافية (Request Additional Service)
* الوصف الموجز

تتيح هذه الحالة للضيف أن يطلب خدمة غير مشمولة ضمن الإقامة، مثل غسيل الملابس، وجبة خاصة، تنظيف إضافي، أو طلب سيارة، ويتم تسجيل الطلب وربطه بالحجز أو الفاتورة الخاصة به.

* الشرط المسبق (Preconditions)

- الضيف مسجَّل ولديه حجز نشط في النظام.

- النظام متصل بقاعدة البيانات والخدمات فعّالة.

- توجد خدمات متاحة يمكن طلبها ضمن القائمة.

* الشرط اللاحق (Postconditions)

- يتم إنشاء سجل للخدمة المطلوبة مرتبط بالضيف والفاتورة.

- تُضاف تكلفة الخدمة إلى الفاتورة النهائية أو المؤقتة.

- يتم تحديث حالة الخدمة إلى "قيد التنفيذ" أو "بانتظار المعالجة".

* حالات الخطأ المحتملة

- طلب خدمة غير متوفرة في الوقت الحالي.

- إدخال بيانات غير صحيحة (مثلاً تاريخ خاطئ).

- إغلاق واجهة الطلب أثناء العملية.

- فشل الاتصال بالخادم.

* حالة النظام عند الخطأ

- لا يتم حفظ الطلب.

- تظهر رسالة توضح السبب (مثل: "الخدمة غير متاحة حاليًا").

- تبقى البيانات المؤقتة مخزنة (إن أمكن) لإعادة المحاولة.

* الجهات الفاعلة (Actors)

- الضيف (Guest) ← عند الطلب الذاتي (من تطبيق أو شاشة الغرفة).

- موظف الاستقبال (Receptionist) ← عند تسجيل الطلب يدويًا نيابة عن الضيف.

- موظف الخدمة (Housekeeping / Room Service) ← لمعالجة الخدمة عند قبولها.

* المحفّز (Trigger)

رغبة الضيف في الحصول على خدمة إضافية خلال فترة الإقامة.

* العملية القياسية (السيناريو الأساسي)

1. يدخل الضيف أو الموظف إلى واجهة "طلب خدمة".

2. يختار الضيف الخدمة المطلوبة من قائمة الخيارات (مثلاً: تنظيف إضافي).

3. يحدد وقت التنفيذ والتفاصيل المطلوبة (مثلاً "بعد ساعتين").

4. يعرض النظام تكلفة الخدمة ويطلب التأكيد.

5. عند التأكيد، يُنشأ سجل الخدمة وتُربط بالفاتورة.

6. يتم إعلام موظف الخدمة بتنفيذ المهمة.

* العمليات البديلة (Alternate Flows)

- A1: إذا أراد الضيف طلب أكثر من خدمة، يمكن إضافتها دفعة واحدة.

- A2: في حال كانت الخدمة مجانية (عرض ترويجي)، يتم تسجيلها دون إضافة تكلفة.

- A3: في حال لم يكن هناك موظف متاح، يتم جدولة الطلب تلقائيًا لأول وقت شاغر

* حالة الاستخدام الثانية :حجز غرفة (Book a Room)
* الوصف الموجز

هذه الحالة تتيح للضيف أو لموظف الاستقبال حجز غرفة متاحة في الفندق ضمن فترة زمنية محددة. تشمل إدخال بيانات الضيف، اختيار الغرفة، تحديد مدة الإقامة، واحتساب التكلفة المبدئية.

* الشرط المسبق

- وجود غرفة واحدة على الأقل شاغرة في الفترة المطلوبة.

- توفر بيانات أولية عن الضيف (اسم، رقم، وسيلة تواصل).

- دخول المستخدم بصلاحية مناسبة (موظف استقبال أو ضيف عبر البوابة الإلكترونية).

* الشرط اللاحق

- يتم إنشاء سجل حجز جديد في قاعدة البيانات.

- يتم تحديث حالة الغرفة إلى “محجوزة”.

- تُجهَّز فاتورة مؤقتة (Pending Invoice) للحجز.

* حالات الخطأ المحتملة

- الغرف كلها مشغولة في الفترة المطلوبة.

- إدخال تاريخ خروج قبل تاريخ الدخول.

- نقص في معلومات الضيف (مثلاً رقم هاتف غير صالح).

- عدم الاتصال بقاعدة البيانات أثناء عملية الحجز.

* حالة النظام عند الخطأ

- يظهر تنبيه للمستخدم مع شرح السبب (مثلاً: “لا توجد غرف شاغرة”).

- تُحفظ البيانات المدخلة مؤقتًا إن أمكن، لتجنب الإعادة.

- لا يُنشأ أي حجز فعلي، وتعود الغرفة لحالة “متاحة”.

* الجهات الفاعلة (Actors)

- الضيف (Guest) – في حالة الحجز الذاتي (عبر الويب أو التطبيق).

- موظف الاستقبال (Receptionist) – عند تنفيذ الحجز داخليًا.

* المحفز (Trigger)

رغبة الضيف في الإقامة بالفندق، سواء بشكل مباشر أو من خلال الاتصال أو الموقع الإلكتروني.

* العملية القياسية (السيناريو الأساسي)

1. يدخل المستخدم (ضيف أو موظف) إلى نظام الحجز.

2. يحدد تاريخ الدخول والخروج وعدد النزلاء.

3. يعرض النظام الغرف المتاحة بناءً على المعايير.

4. يختار المستخدم الغرفة المناسبة.

5. يدخل بيانات الضيف (الاسم، رقم الهوية، الهاتف...).

6. يحتسب النظام التكلفة المبدئية.

7. يتم تأكيد الحجز وتحديث حالة الغرفة.

8. يُولَّد سجل الحجز ويُربط بفاتورة مؤقتة.

* العمليات البديلة (Alternate Flows)

- A1: في حال لم يجد المستخدم غرفة مناسبة، يُعرض عليه خيارات بديلة (تواريخ مختلفة أو غرف أعلى فئة).

- A2: إذا اختار الضيف الدفع المسبق، يتم توجيهه مباشرة إلى واجهة الدفع.

- A3: يمكن للمدير اعتماد حجز “معلق” في حال الرغبة بتجاوز السعة مؤقتًا.

* مخطط الصفوف class diagram
* المستخدم user
* الوصف: يمثل أي مستخدم للنظام (موظف استقبال، محاسب، مدير، إلخ).
* الخصائص:
  + userID: int
  + name: String
  + username: String
  + password: String
  + role: String // تحديد نوع المستخدم
* الدوال:
  + login()
  + logout()
  + changePassword()
* الضيف Guest
* الوصف: يمثل ضيفًا قام بالحجز في الفندق.
* الخصائص:
  + guestID: int
  + fullName: String
  + phone: String
  + email: String
* الدوال:
  + makeReservation()
  + requestService()

---

* الغرفة Room
* الوصف: تمثّل غرفة فندقية قابلة للحجز.
* الخصائص:
  + roomNumber: String
  + type: String // مفردة، مزدوجة، جناح
  + pricePerNight: double
  + status: String // شاغرة، محجوزة
* الدوال:
  + updateStatus()
  + calculateTotalPrice(nights: int): double

---

* الحجز Reservation
* الوصف: يمثل عملية حجز فعلية لغرفة.
* الخصائص:

- reservationID: int

- checkInDate: Date

- checkOutDate: Date

- status: String

- totalAmount: double

* الدوال:

- calculateDuration(): int

- calculateTotal(): double

- cancelReservation()

* العلاقات:

- ترتبط بـ Guest و Room

---

* الفاتورة Invoice
* الوصف: تمثل الفاتورة الناتجة عن الحجز والخدمات.

الخصائص:

- invoiceID: int

- dateIssued: Date

- totalAmount: double

- paymentStatus: String

* الدوال:

- generateInvoice()

- markAsPaid()

- exportToPDF()

* العلاقات:

- مرتبطة بـ Reservation

- يمكن أن تحتوي على عدة `ServiceItem`s

---

* خدمة ServiceItem
* الوصف: أي خدمة إضافية مطلوبة من الضيف (غسيل، طعام...).
* الخصائص:

- serviceID: int

- name: String

- cost: double

- description: String

- dateProvided: Date

* الدوال:

- applyToInvoice()

- updateServiceDetails()

المعاملات Expense

* الوصف: تمثل مصروف إداري أو تشغيلي يُدخله المحاسب.
* الخصائص:

- expenseID: int

- type: String (صيانة، رواتب، تموين...)

- amount: double

- date: Date

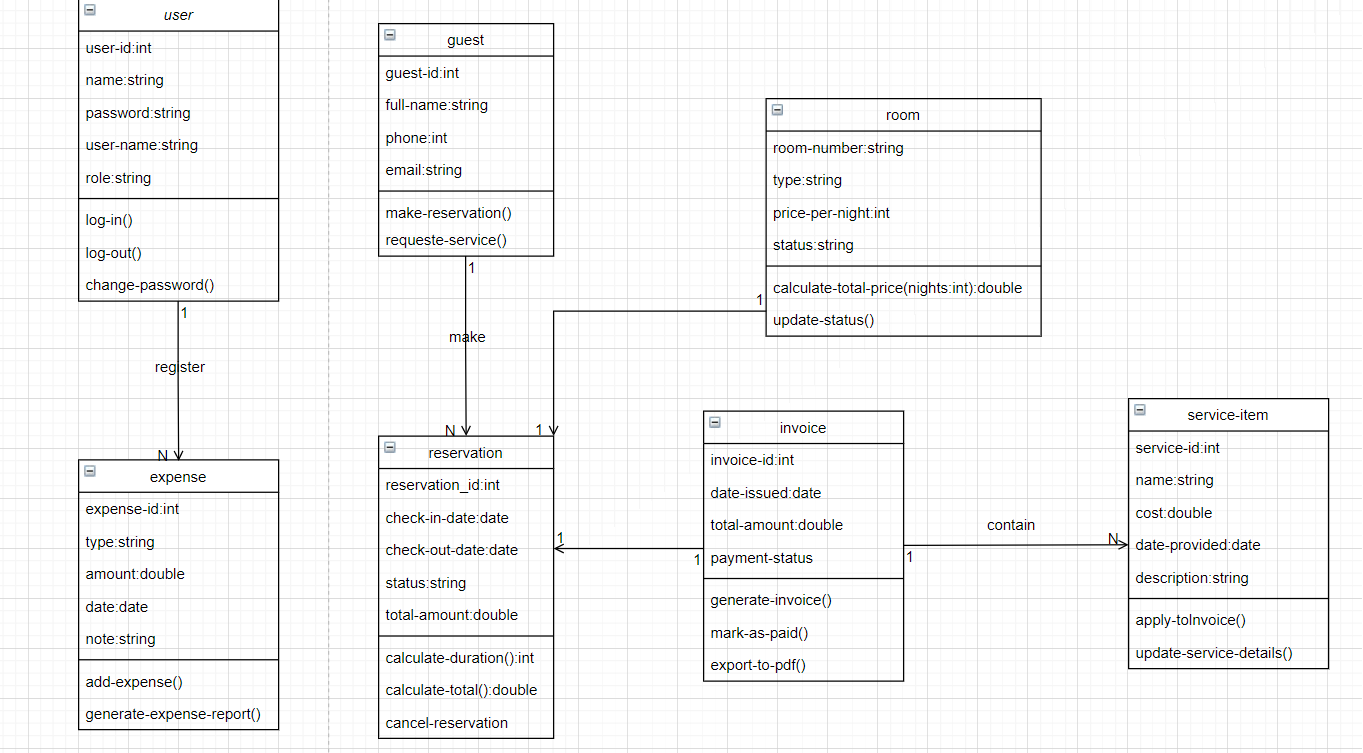
- note: String

* الدوال:

- addExpense()

- generateExpenseReport()

|  |  |  |
| --- | --- | --- |
| العلاقة | التعددية | ماذا تعني |
| غرفة \_حجز | M:1 | الغرفة الواحدة يمكن حجزها عدة مرات |
| حجز\_فاتورة | :11 | ينتج عن كل حجز فاتورة واجدة |
| فاتورة \_خدمات | M:1 | الفاتورة قد تحتوي خدمات متعددة |
| مستخدم\_معاملة | M:1 | المستخدم يمكنه إدخال أكثر من معاملة |



* مخطط الكيانات ER
* Guest (الضيف)

Attributes\*: guestID (PK), fullName, phone, email

* Room (الغرفة)

Attributes: roomNumber (PK), type, pricePerNight, status

* Reservation (الحجز)

Attributes: reservationID (PK), guestID (FK), roomNumber (FK), checkInDate, checkOutDate, totalAmount, status

* Invoice (الفاتورة)

Attributes: invoiceID (PK), reservationID (FK), dateIssued, totalAmount, paymentStatus

* ServiceItem (خدمة إضافية)

Attributes: serviceID (PK), invoiceID (FK), name, cost, description, dateProvided

* Expense (معاملة مالية )

Attributes: expenseID (PK), userID (FK), type, amount, date, note

* User (الموظف)

Attributes: userID (PK), name, username, password, role

