



جامعة اللاذقية

كلية الهندسة المعلوماتية

السنة الرابعة

قسم البرمجيات ونظم المعلومات

## تحليل نظام إدارة مطعم (Restaurant Management System Analysis)

إعداد الطلاب:

خليل عدنان العماري

اسماعيل نضال طالب

مجد عبدالله جرجب

محمد صادق جمعه عوض

يوسف كمال اسعد

العام الدراسي

2025\_2024

# ❖ مشروع مادة تحليل النظم المالية

## تحليل نظام ادارة مطعم: Restaurant Management System

- يهدف النظام إلى تحسين الكفاءة التشغيلية والمالية للمطعم من خلال أتمتة العمليات اليومية، مثل حجز الطاولات، الطلبات، الفوترة، والتقارير المالية.
- في هذا المشروع سنحلل تفاعل الأطراف المختلفة مع النظام ونوضح الأثر المالي لكل عملية ضمن السياق العام.

### الجهات الفاعلة (Actors) ودور كل منها:

#### 1. الزبون الجديد (New Customer)

- . عرض قائمة الطعام فقط.
- . إنشاء حساب جديد أو تسجيل الدخول.
- . لا يمكنه الطلب أو الدفع دون تسجيل.

#### 2. الزبون المسجل (Customer)

- . تسجيل الدخول والخروج.
- . حجز طاولة.
- . عرض قائمة الطعام.
- . اختيار الأصناف وإضافة ملاحظات.
- . إرسال الطلب.
- . تحديد نوع الطلب (سفري / داخلي).
- . تعديل الطلب أو إلغاؤه قبل التحضير.

- الدفع (نقدًا / بطاقة).
  - استلام الفاتورة الإلكترونية.
  - تقييم الطعام بعد الاستلام.
  - تتبع حالة الطلب.
  - الاطلاع على سجل الطلبات السابقة.
- 

### 3. المدير (Manager)

- عرض التقارير المالية اليومية والشهرية.
  - إدارة الموظفين (إضافة – حذف – تعديل).
  - تعديل قائمة الطعام (إضافة أصناف – تغيير الأسعار – تعطيل عناصر).
  - تتبع أداء النادل والطاهي بناءً على الطلبات.
  - تحليل المبيعات والفترات الزمنية.
  - اتخاذ قرارات استراتيجية مثل التخفيضات أو تعديل أوقات العمل.
- 

### 4. النادل (Waiter)

- استقبال الزبائن وحجز الطاولة لهم.
  - إرسال الطلبات إلى المطبخ في حال كانت من الزبون مباشرة أو بعد مراجعتها.
  - إضافة ملاحظات على الطلب نيابة عن الزبون.
  - متابعة حالة الطلب مع المطبخ.
  - تسليم الطلب للزبائن.
  - تحديث حالة الطاولة (مشغولة – فارغة).
-

## 5. الطاهي (Chef)

- استلام الطلبات المُرسلة من النادل أو النظام.
  - عرض تفاصيل الطلب والملاحظات.
  - تغيير حالة الطلب إلى "قيد التحضير" أو "جاهز".
  - إرسال إشعار عند الانتهاء من التحضير.
  - كتابة ملاحظات (مثلاً: لا يتوفر مكوّن – تغيير وقت التحضير).
- 

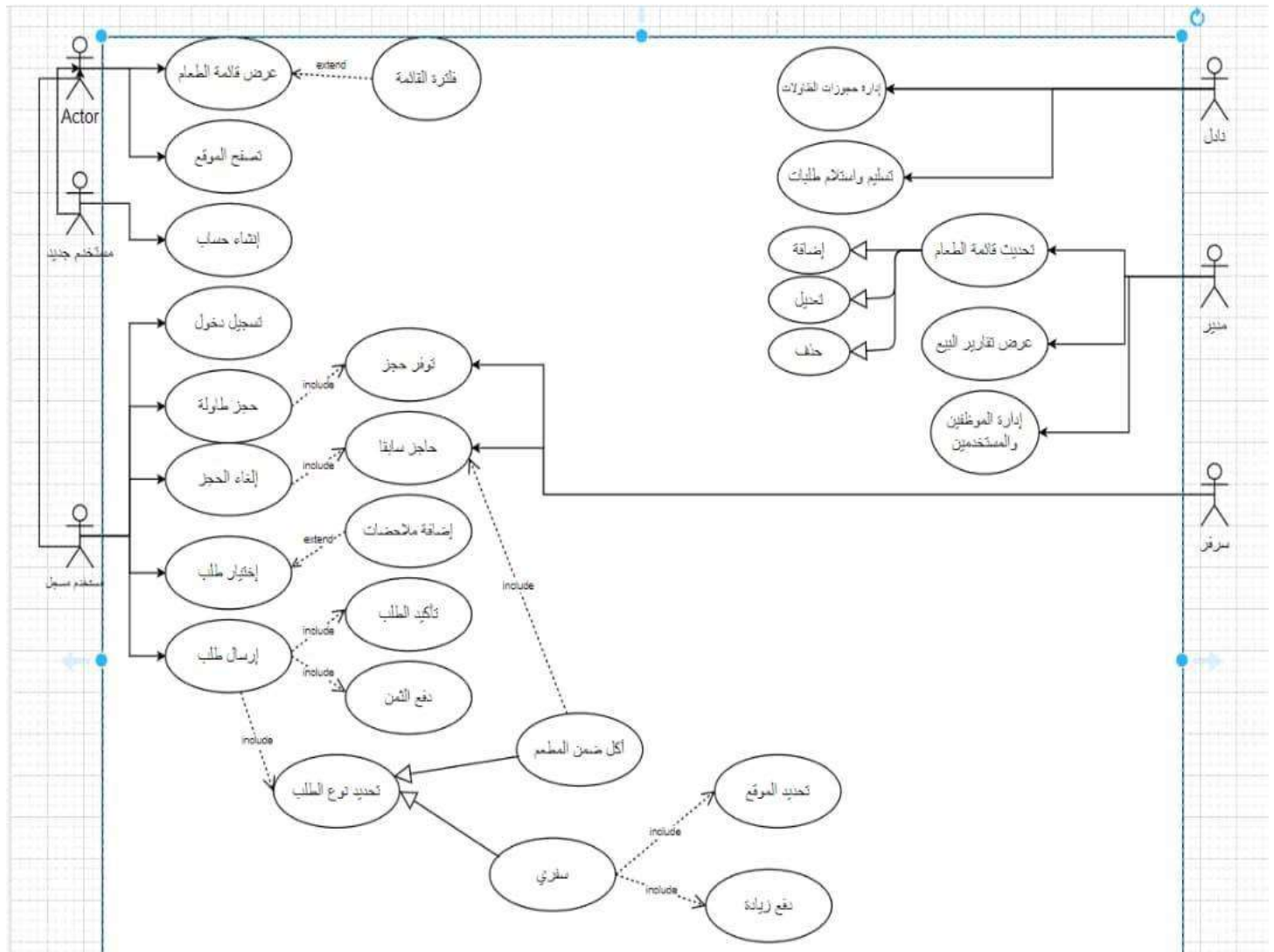
## 6. الكاشير (Cashier)

- عرض الفواتير المفتوحة والطلبات الجاهزة للدفع.
  - تنفيذ عمليات الدفع (نقدًا أو إلكترونيًا).
  - إصدار الفواتير وربطها بالحسابات البنكية أو النقدية.
  - مراجعة مدفوعات اليوم والتقارير المالية اليومية.
  - معالجة أخطاء الدفع (مثل: فشل البطاقة).
  - تقديم وسائل دفع بديلة.
- 

## علاقة ما سبق مع تحليل النظم المالية:

- زيادة الإيرادات: من خلال تحسين دوران الطاولة وتجربة الزبائن.
- خفض التكاليف: عبر تقليص الاعتماد على الورق والأخطاء البشرية.
- ضبط مالي دقيق: من خلال تسجيل آلي للعمليات وربطها بالحساب البنكي.
- تمكين القرار المالي: بتوفير تقارير فورية لدعم اتخاذ القرار المالي السليم.

## والان ننتقل الى رسم مخطط حالات الاستخدام use case diagram:



## ➤ المتطلبات الوظيفية (Functional Requirements)

### 1. إدارة الطلبات Order Management :

- تمكين الزبائن من استعراض قائمة الطعام وتقديم طلباتهم إلكترونياً.
- تسجيل الطلبات وتوجيهها إلى المطبخ بشكل تلقائي.
- تحديث حالة الطلب (قيد التحضير - جاهز - مُسَلَّم).
- إمكانية تعديل أو إلغاء الطلب قبل تحضيره.
- إصدار الفاتورة تلقائياً وربطها بالطلب.

### 2. إدارة المخزون Inventory Management :

- تحديث تلقائي للمخزون عند إرسال طلب إلى المطبخ.
- إرسال تنبيهات في حال انخفاض كميات المكونات.
- عرض تقارير المخزون اليومية والأسبوعية للمدير.
- ربط الأصناف بقوائم المكونات لتتبع الاستهلاك.

### 3. إدارة الموظفين Staff Management :

- تسجيل بيانات الموظفين (نُدل، طهارة، محاسبين).
- تتبع ساعات العمل والحضور والانصراف.
- تحديد صلاحيات الوصول إلى النظام حسب الدور.
- تقييم أداء الموظفين بناءً على الطلبات المُعالجة وسرعة الخدمة.

## 4. التقارير والتحليلات :Reporting:

- توليد تقارير مالية يومية وشهرية.
  - تحليل الإيرادات والمصروفات حسب الصنف أو الفترة الزمنية.
  - رصد الطلبات الأعلى مبيعاً والمواسم ذات الإقبال العالي.
  - تقديم رسوم بيانية لمساعدة المدير في اتخاذ قرارات استراتيجية.
- 

## ➤ المتطلبات غير الوظيفية (Non-Functional Requirements)

### 1. الأداء (Performance):

- سرعة في تحميل القوائم واستجابة النظام لطلبات المستخدم.
- قدرة على معالجة عدة طلبات متزامنة دون بطء.
- الحفاظ على زمن استجابة أقل من 3 ثوانٍ في المتوسط.

### 2. الأمان (Security):

- تشفير بيانات الزبائن وعمليات الدفع.
- اعتماد كلمات مرور قوية وصلاحيات دخول متعددة المستويات.
- تسجيل عمليات الدخول والخروج ومراجعة النشاطات.
- حماية النظام من هجمات الاختراق أو التلاعب بالبيانات.

### 3. التوافقية (Compatibility):

- توافق النظام مع متصفحات الويب الرئيسية. (Chrome, Firefox, Edge).
- إمكانية التشغيل على أنظمة مختلفة. (Windows, Android, iOS).
- دعم الربط مع الأنظمة المحاسبية الخارجية وتحديثات قواعد البيانات.
- واجهة مرنة تدعم التفاعل عبر اللمس والحاسوب.

---

### ➤ والآن ننتقل لقسم توصيف حالتي استخدام:

#### أ. الحالة الأولى: تقديم/اختيار/ طلب (Place Order)

##### 1. الاسم (Name):

- Place Order

##### 2. الوصف الموجز (Short Description):

- زبون يحدد أطباقه من القائمة ويرسل الطلب للمطبخ.

##### 3. الشرط المسبق (Precondition):

- الزبون مسجل دخول.

##### 4. الشرط اللاحق (Subsequent Condition):

- الطلب مُسجل ومُرسل للطاهي.



## 5. حالات الخطأ (Error Condition):

- صنف غير متوفر / بيانات ناقصة.

## 6. حالة النظام عند الخطأ (System Statuses At Error):

- عرض رسالة تنبيه واقتراح بدائل أو طلب تصحيح.

## 7. الجهات الفاعلة (Actors):

- الزبون، النادل.

## 8. المحفز (Trigger):

- اختيار الزبون للأطباق والضغط على "تأكيد الطلب".

## 9. العملية القياسية (Main Flow):

- تشمل:

1. عرض قائمة الطعام.
2. اختيار الأصناف.
3. إدخال ملاحظات اختيارية.
4. تأكيد الطلب.
5. النظام يُرسل الطلب للمطبخ.

## 10. العمليات البديلة (Alternative Flows):

- في حال نفاد صنف، يظهر بديل.
- إذا لم يُسجل الدخول، يتم التوجيه إلى صفحة الدخول.

## ب. الحالة الثانية: معالجة الدفع (Process Payment)

### 1. الاسم (Name):

• Process Payment

### 2. الوصف الموجز (Short Description):

• بعد انتهاء الطلب، يُجري الكاشير عملية الدفع ويُصدر الفاتورة.

### 3. الشرط المسبق (Precondition):

• الطلب جاهز والفاتورة صدرت.

### 4. الشرط اللاحق (Subsequent Condition):

• تم الدفع وتحديث حالة الطلب.

### 5. حالات الخطأ (Error Condition):

• فشل البطاقة البنكية / إدخال مبلغ خاطئ.

### 6. حالة النظام عند الخطأ (System Statuses At Error):

• عرض تنبيه بالكاشير واقتراح وسيلة دفع بديلة.

### 7. الجهات الفاعلة (Actors):

• الكاشير، الزبون.

### 8. المحفز (Trigger):

- رغبة الزبون بالدفع بعد انتهاء الخدمة.

## 9. العملية القياسية (Main Flow):

- تشمل:

1. عرض الطلب.
2. اختيار وسيلة الدفع.
3. التحقق من صحة العملية.
4. خصم المبلغ وإصدار الفاتورة.

## 10. العمليات البديلة (Alternative Flows):

- محاولة الدفع ببطاقة أخرى.
- التحويل إلى الدفع نقدًا.

## ➤والآن لنرسم مخطط الصفوف (Class Diagram):

### ➤ أولاً: الصفوف (Classes)

#### 1. زبون جديد (New Customer):

. الخاصية:

n-name: string (اسم النزِيل).

. الوظيفة:

Bookatable() (يقوم النزِيل بحجز طاولة).

#### 2. الزبون (Customer):

. الخاصية:

c-name: string (اسم الزبون).

. الوظائف:

1. view menu() (عرض قائمة الطعام).

2. Submit a request() (إرسال الطلب).

3. Order confirmation() (تأكيد الطلب).

4. payabill() (دفع الفاتورة).

#### 3. وسيلة الدفع (select):

. الوظائف:

1. cash() (الدفع نقدًا).

2. Bank account() (الدفع عن طريق البنك).

#### 4. الموظف (Employee):

. الخاصية:

type: string (نوع الموظف [نادل، طاهٍ، كاشير...]).

. الوظائف:

1. Bookatable() (حجز طاولة للزبون).

2. payabill() (تنفيذ عملية الدفع).

3. Order confirmation() (تأكيد الطلب).

#### 5. الشيف (Chef):

. الخاصية:

s-name: string (اسم الشيف).

. الوظيفة:

notice() (إضافة ملاحظات حول الطلب).

#### 6. النادل (Waiter):

. الخاصية:

w-name: string (اسم النادل).

. الوظائف:

. notice() (استلام ملاحظات من الزبائن).

. send request() (إرسال الطلب إلى المطبخ).

## 7. المدير (Manager):

. الخاصية:

m-name: string (اسم المدير).

. الوظائف:

. view sales() (عرض تقارير المبيعات).

. view deliy() (عرض التقارير اليومية).

## 8. الكاشير (Cashier):

. الخاصية:

k-name: string (اسم الكاشير).

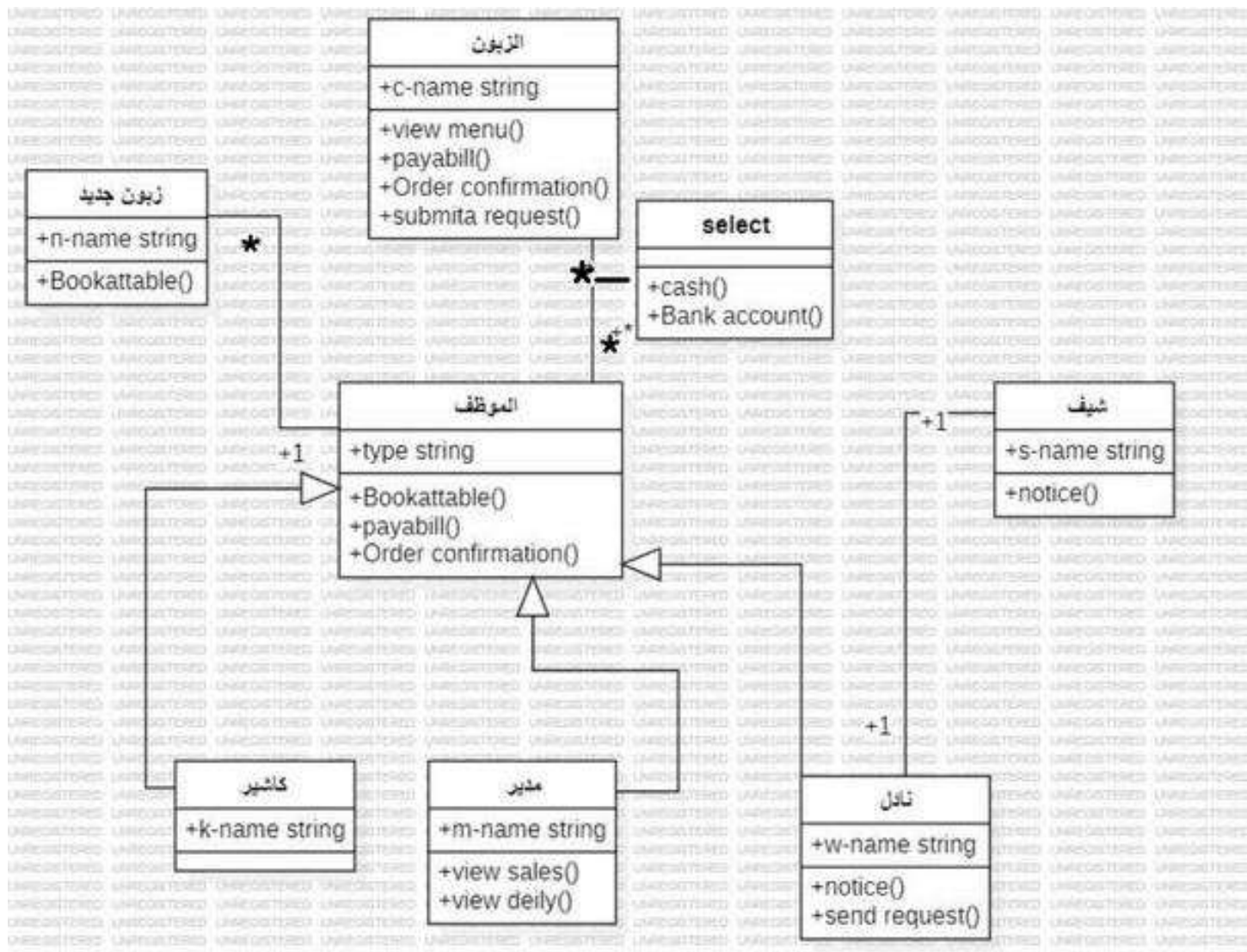
. المهام:

يُفترض به تنفيذ عمليات الدفع، إصدار الفواتير، ومتابعة الحالة المالية.

## ➤ ثانيًا: العلاقات (Relationships)

الشرح	نوع العلاقة	العلاقة بين الكائنات
كل زبون يختار وسيلة دفع واحدة في كل مرة (نقدًا أو عبر حساب بنكي) ولا يستخدم أكثر من خيار في نفس الطلب.	One to One	الزبون ↔ (Customer) وسيلة الدفع (Select)
الزبون يمكنه إجراء عدة طلبات مختلفة على مدار اليوم، لكن كل طلب يعود لزبون واحد فقط.	One to Many	الزبون ↔ (Customer) الطلبات (Orders)

يمكن ربط عدة ملاحظات بطلب واحد مثل "بدون ملح" أو "حار جدًا"؛ كل ملاحظة ترتبط بطلب محدد.	One to Many	الطلب ↔ (Order) الملاحظات (Notice)
عدة طلبات تتم معالجتها من قبل نفس الموظف (مثل النادل أو الكاشير)، لكن كل طلب يمر عبر موظف واحد محدد.	Many to One	الطلب ↔ (Order) الموظف (Employee)
عدة ضيوف يمكنهم مشاركة نفس الطلب مع الزبون الرئيسي، لكن لا يمكنهم إنشاء طلب منفصل دون زبون مسجل.	Many to One	الضيف ↔ (Guest) الطلب الواحد
المدير يستعرض تقارير مختلفة (مبيعات، نشاط يومي...)، وكل تقرير يرتبط بمدير واحد فقط في النظام.	One to Many	المدير ↔ (Manager) التقارير (Reports)
النادل الواحد يمكنه متابعة عدة طلبات من زبائن مختلفين، لكن كل طلب يتبع نادلاً محدداً لمتابعته.	One to Many	النادل ↔ (Waiter) الطلبات
كل جلسة لزبون تتضمن حجز طاولة واحدة، والعكس كذلك، ولكن يمكن للزبون إجراء حجوزات متعددة في جلسات مختلفة لاحقاً.	One to One (لكل جلسة)	الزبون ↔ (Customer) حجز الطاولة (Book a Table) Table)





## ➤ وفي النهاية مخطط ERD

🚩 أولاً: شرح الكيانات (Entities)

### 1. Chef (الطاهي):

- يحتوي على ChefID, Role, Name.
- يمثل الموظف المسؤول عن إعداد الأطباق داخل المطبخ.

### 2. Waiter (النادل):

- يحتوي على WaiterID, Name.
- يتولى خدمة الزبائن وتسليم الطلبات للمطبخ.

### 3. Customer (الزبون):

- يحتوي على CustomerID, Name, Phone, Email.
- يمثل مستخدم النظام الذي يقدم الطلبات ويدفع الفواتير.

### 4. Menu Item (عنصر قائمة الطعام):

- يحتوي على ItemID, Name, Price, Availability.
- يمثل وجبة أو طبقاً يمكن للزبائن طلبه.

### 5. Order (الطلب):

- يحتوي على OrderID, Date, Status.
- يمثل عملية الشراء التي يقوم بها الزبون.

### 6. Order Details (تفاصيل الطلب):

- يحتوي على `DetailID, Price, Quantity`.
- يربط الطلب بالعناصر التي تم اختيارها من القائمة.

## 7. Payment (الدفع):

- يحتوي على `PaymentID, Amount, Payment Method`.
- يمثل عملية الدفع التي تتم بعد إتمام الطلب.

## 8. Cashier (الكاشير):

- يحتوي على `CashierID, Name`.
- الموظف الذي يعالج عملية الدفع ويصدر الفاتورة.

## 9. Invoice (الفاتورة):

- كيان بدون خصائص ظاهرة، لكن يتم ربطه بالكاشير والطلب لغرض تسجيل المعاملات.

## ثانيًا: شرح العلاقات وأنواعها

العلاقة	نوع العلاقة	الشرح
Chef ↔ Waiter	1:N	كل طاهٍ يمكن أن يعمل مع عدة نُدل، ولكن كل نادل مرتبط بطاهٍ واحد فقط.
Waiter ↔ Customer	1:N	كل نادل يخدم عدة زبائن، ولكن كل زبون يتعامل مع نادل واحد في الجلسة.
Customer ↔ Order	1:N	الزبون يمكنه عمل عدة طلبات، لكن كل طلب يتبع لزبون واحد.
Order ↔ Order Details	1:N (أو N:M)	كل طلب يحتوي على عدة تفاصيل، وكل تفصيلة ترتبط بطلب؛ عبر جدول وسيط.

Menu Item ↔ Order Details	1:N	كل عنصر من القائمة يمكن أن يظهر في تفاصيل عدة طلبات.
Order ↔ Cashier	N:1	الكاشير يعالج طلبات عديدة، لكن الطلب يُسجل باسم كاشير واحد فقط.
Cashier ↔ Invoice	1:1	كل فاتورة يصدرها كاشير واحد، وكل كاشير يصدر فاتورة واحدة في كل مرة.
Order ↔ Payment	1:N	يمكن أن يكون للطلب أكثر من عملية دفع (مثل دفعة نقدية وجزء ببطاقة).

