الجمهورية العربية السورية
جامعة اللاذقية
كلية الهندسة المعلوماتية
قسم البرمجيات و نظم المعلومات

Pizza Shop Management System

تقديم :

نغم حيدر
رياض حمدان
محمود حلواني
محسن حبيب
محمود الخنسا

بإشراف :

د.غيث بلال
م.زينب محفوض
م.حمزة دواي

العام الدراسي 2024_2025

تحليل نظام متجر لبيع البيتزا pizza shop Management System:

هذا النظام يهدف الى إدارة متجر بيتزا بكفاءة بدءاً من استقبال الطلبات وحتى توصيل الطلب

يساعد هذا النظام على عدة أمور حيث أنه يقوم ب أربعة أمور رئيسية :

١- تبسيط عمليات الدفع والطلب

٢- إدارة المخزون والموظفين

٣- تتبع الطلبيات وتحسين خدمة العملاء

٤- إعداد تقارير مالية بالمبيعات والأرباح والمشتريات

.....

الفاعلين **Actors** لدينا: زبون **Client**، أمين الصندوق **Cashier**،الطاهي **Chef**، السائق **Delivery**، المدير **Manager**:

مهام الزبون: طلب الحصول على بيتزا قد يكون هذه الطلب إما عن طريق موقع الانترنت أو من داخل المحل بشكل مباشر و تخصيص البيتزا من خلال اختيار الحجم والصلصات و دفع الفاتورة من خلال الكاش أو البطاقة.

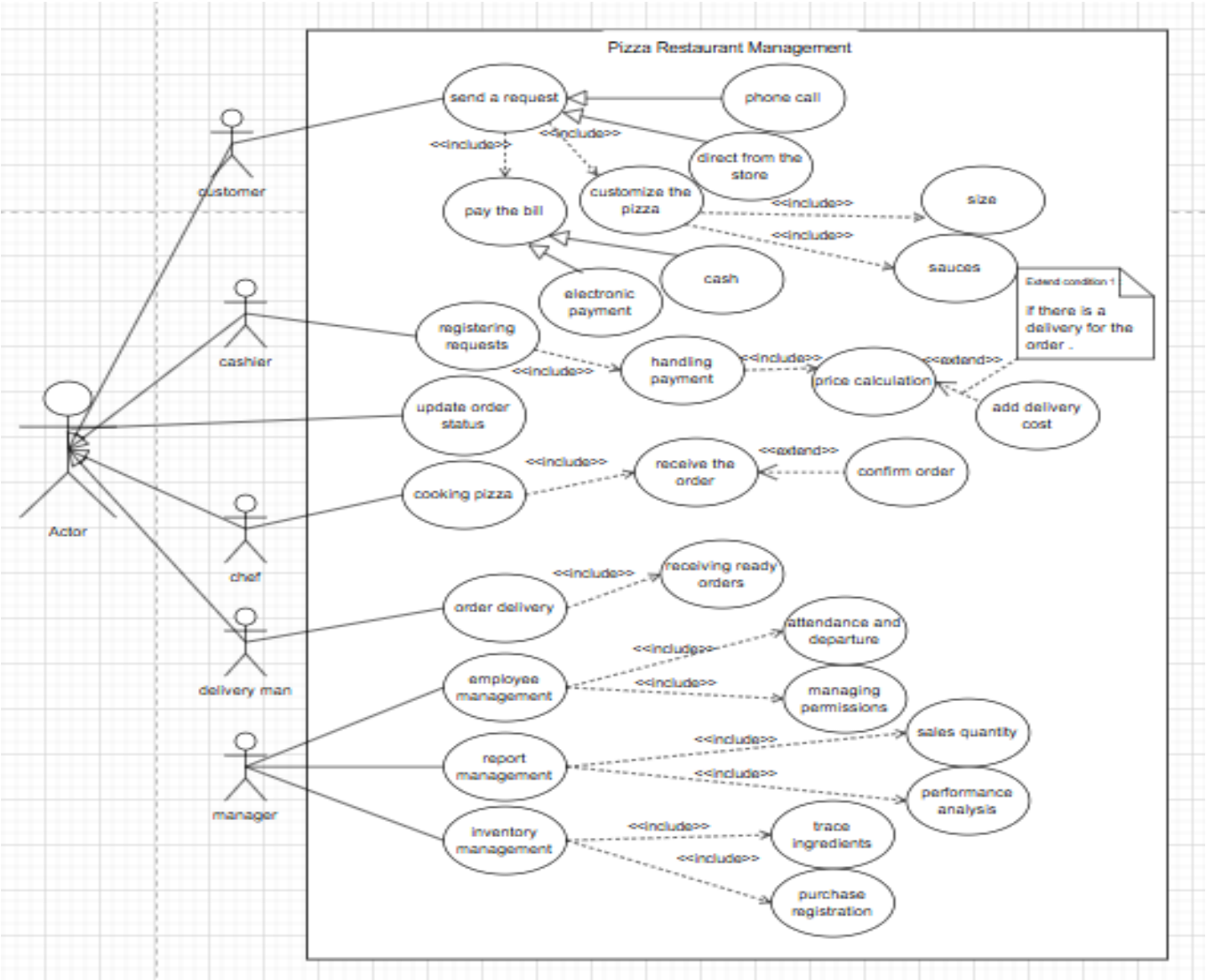
مهام أمين الصندوق: هو الذي يقوم بتسجيل طلبات الزبون والتعامل مع الدفع من خلال حساب السعر.

مهام الطاهي: يستلم الطلبات ويقوم بتلبية الطلب وطهي البيتزا ويحدّث حالة الطلب .

مهام السائق: يستلم الطلبات الجاهزة و يوصل الطلبات للعميل .

مهام المدير: يدير قوائم الموظفين و التقارير.

والان نرسم مخطط use case diagram :



المتطلبات الوظيفية :

١- إدارة الطلبات Order Management :

الوظيفة :	الوصف :
إنشاء طلب جديد	يتم من خلال اتصال هاتفي أو محلي
تخصيص البيتزا	اختيار حجم والمكونات الإضافية وإزالة المكونات غير المرغوب بها

حساب السعر	احتساب التكلفة بناء على الطلبات + ضرائب + رسوم توصيل
تحديث حالة الطلب	قيد التحضير ، جاهز، قيد التوصيل، مكتمل

٢- إدارة المخزون Inventory Management :

الوظيفة :	الوصف:
تتبع المكونات	مراقبة كميات الجبن والصلصة وغيرها
تنبيهات بالنفاذ	اشعار المدير عند اقتراب انتهاء أي مكون
تسجيل المشتريات	إضافة كميات جديدة بعد التوريد

٣- إدارة الموظفين Staff Management :

الوظيفة :	الوصف:
تسجيل الحضور والانصراف	حضور الطهاة والسائقين
إدارة الصلاحيات	تحديد أدوار الموظفين (تعديل الأسعار من خصائص المدير)

٤- التقارير والتحليلات Reporting :

الوظيفة	الوصف

تقاير المبيعات اليومية	معرفة الطلبات الناجحة والملغاة
تحليل الاداء	تحديد اكثر المبيعات مبيعاً واوقات الذروة

المتطلبات غير الوظيفية :

١- **الأداء** : تحمل ١٠٠ طلب متزامنا خلال ساعات الذروة، وقت الاستجابة اقل من ٥ دقائق لأي عملية.

٢- ***الأمان*** : تشفير بيانات الدفع.

٣- ***التوافقية***: دعم متصفحات حديثة و العمل على أجهزة حديثة للسائقين لتحديد موقع الطلب.

- **توصيف حالات الاستخدام :**

الحالة الأولى :

١. الاسم:

تقديم طلب جديد**(Place Order)**

٢. الوصف الموجز:

يسمح للعميل بإنشاء طلب بيتزا جديد عن طريق اختيار نوع البيتزا، الحجم، الإضافات، وطريقة الدفع، وتأكيد الطلب.

٣. الشرط المسبق:

- يجب على الزبون أن يتصل للطلب .

- يجب أن يكون هناك اتصال بالإنترنت (إذا كان الطلب عبر التطبيق أو الموقع).
- يجب أن يكون المخزون متاحًا للمكونات المطلوبة.

٤. الشرط اللاحق:

- يتم حفظ الطلب في قاعدة البيانات مع حالة "قيد المعالجة."
- يتم خصم الكميات المطلوبة من المخزون.
- يتم إرسال إشعار تأكيد للعميل (عبر البريد الإلكتروني أو الرسائل النصية).

٥. حالات الخطأ:

- **الخطأ ١** :عدم توفر أحد المكونات (مثل نوع معين من الجبن).
- **الخطأ ٢** :فشل عملية الدفع (إذا كان الدفع إلكترونيًا).
- **الخطأ ٣** :إدخال عنوان توصيل غير صحيح أو خارج نطاق التوصيل.

٦. حالة النظام عند حدوث خطأ:

- يعرض النظام رسالة خطأ توضح السبب (مثال: "نوع الجبن غير متوفر حاليًا").
- يحتفظ النظام بالطلب في حالة "معلق" حتى يتم تصحيح الخطأ.
- في حالة فشل الدفع، يُطلب من العميل إعادة المحاولة أو اختيار طريقة دفع أخرى.

٧. الجهات الفاعلة (Actors) :

- **الزبون (Client)** :المستخدم الرئيسي الذي يقوم بإنشاء الطلب...

٨. المُحفز (Trigger) :

- يقوم العميل بتقديم طلبه في المتجر .
- يتصل العميل هاتفيًا بالمتجر لطلب بيتزا.

٩. العملية القياسية (Main Flow) :

- يدخل العميل إلى النظام (تطبيق/موقع/هاتف).
- يتصفح قائمة البيتزا المتاحة.
- يختار نوع البيتزا والحجم (مثال: بيتزا مارغريتا – وسط).
- يضيف الإضافات (إن وجدت) مثل جبن إضافي أو فطر.
- يحدد عنوان التوصيل وطريقة الدفع (نقدي/بطاقة).
- يؤكد الطلب.
- يعرض النظام رسالة تأكيد مع رقم الطلب.

١٠. العمليات البديلة (Alternative Flows) :

- **إذا كان أحد المكونات غير متوفر:**
  - يعرض النظام خيارات بديلة (مثال: استبدال الجبن بنوع آخر).
  - أو يسمح للعميل بإلغاء الطلب.

- **إذا كان العنوان خارج نطاق التوصيل:**
  - يقترح النظام أماكن استلام قريبة (مثل الفرع الأقرب).

الحالة الثانية :

١. الاسم:

دفع الفاتورة **(Pay Bill)**

٢.الوصف الموجز:

تسمح هذه الحالة للعميل أو الموظف بإتمام عملية الدفع مقابل طلب البيتزا، سواء كان ذلك نقداً عند الاستلام أو عبر وسائل الدفع الإلكتروني. تدعم الحالة طرق دفع متعددة وتوفر تأكيداً للعملية.

٣. الشرط المسبق:

- يجب أن يكون الطلب موجوداً في النظام وحالته "جاهز للدفع."
- يجب أن يكون العميل قد اختار طريقة دفع (نقدي/إلكتروني).
- في حالة الدفع الإلكتروني، يجب أن يتوفر اتصال بالإنترنت.
- يجب أن يكون لدى العميل رصيد كافٍ (في حالة الدفع بالبطاقة).

٤. الشرط اللاحق:

- يتم تحديث حالة الطلب إلى "تم الدفع."
- يتم تسجيل المعاملة في سجل المبيعات.
- يتم إصدار إيصال إلكتروني أو ورقي للعميل.
- في حالة الدفع عند الاستلام، يتم تحديث الحالة إلى "بانتظار التسليم."

٥. حالات الخطأ:

- **الخطأ ١** : فشل عملية الدفع الإلكتروني (نقص الرصيد/انتهاء البطاقة).
- **الخطأ ٢** : انقطاع الاتصال بالإنترنت أثناء الدفع الإلكتروني.
- **الخطأ ٣** : رفض جهاز الصراف الآلي أو نقطة البيع للعملية.
- **الخطأ ٤** : إدخال بيانات دفع غير صحيحة(مثل CVV خاطئ).

٦. حالة النظام عند حدوث خطأ:

- يعرض النظام رسالة خطأ توضح سبب الفشل.
- يحتفظ بحالة الطلب "في انتظار الدفع."
- في حالة الدفع الإلكتروني، يعيد توجيه العميل لإعادة المحاولة أو اختيار طريقة دفع أخرى.
- يسجل محاولة الدفع الفاشلة في سجل النظام.

٧. الجهات الفاعلة (Actors) :

- **الزبون (Client)** :يقوم بإتمام عملية الدفع.

- **الموظف (Employee)** :يساعد في عملية الدفع النقدي عند الاستلام أو حل مشاكل الدفع.
- **النظام (System)** :يدير عملية الدفع ويتكامل مع بوابات الدفع.
- **بوابة الدفع (Payment Gateway)** :(في حالة الدفع الإلكتروني) تطبق عملية التحقق من الدفع.

٨. المُحفز(Trigger) :

- وصول العميل إلى خطوة الدفع في عملية الطلب.
- اختيار العميل لطريقة الدفع وتأكيده الرغبة في الإتمام.

٩. العملية القياسية (Main Flow) :

- يصل النظام إلى خطوة الدفع بعد تأكيد الطلب.
- يعرض النظام خيارات الدفع المتاحة (نقدي/بطاقة/محفظة إلكترونية).
- يختار العميل طريقة الدفع:
  - **نقداً** :يؤكد استلام الطلب عند التسليم.
  - **إلكترونياً** :يدخل بيانات الدفع (رقم البطاقة، تاريخ الانتهاء، CVV)
- يرسل النظام طلب الدفع إلى بوابة الدفع (إذا كان إلكترونياً).
- تجري بوابة الدفع التحقق من البيانات وتوفر الرصيد.
- يتلقى النظام تأكيداً بالدفع الناجح.
- يقوم النظام بما يلي:

- يحدث حالة الطلب إلى "تم الدفع."
- يولد إيصالاً ويوفره للعميل.
- يرسل تأكيداً بالدفع عبر البريد/الرسائل النصية.

١٠.العمليات البديلة (Alternative Flows) :

- **إذا فشل الدفع الإلكتروني (الخطأ ١ أو ٤):**
  - يعرض النظام: "فشل عملية الدفع، الرجاء التحقق من البيانات والمحاولة مرة أخرى."
  - يسمح للعميل بإدخال بيانات جديدة أو اختيار طريقة دفع مختلفة.

- **إذا انقطع الاتصال (الخطأ ٢):**
  - يعرض النظام: "فقدان الاتصال، الرجاء المحاولة لاحقاً."
  - يحتفظ بالطلب في حالة "في انتظار الدفع."

- **إذا كان الدفع عند الاستلام:**
  - يحدد النظام حالة الطلب "بانتظار التسليم."
  - يتم تسجيل الدفع يدوياً من قبل الموظف عند التسليم.

الحالة الثالثة :

١. الاسم:

تخصيص البيتزا (Customize Pizza)

٢.الوصف الموجز:

تمكن العميل من تخصيص البيتزا عن طريق اختيار الحجم، نوع العجين، الصلصة، الإضافات، وإزالة المكونات غير المرغوب فيها.

٣. الشرط المسبق:

- يجب أن يكون العميل مسجلًا في النظام أو يقوم بإنشاء طلب جديد.
- يجب أن تكون المكونات المطلوبة متوفرة في المخزون.

٤. الشرط اللاحق:

- يتم حفظ التخصيصات في الطلب.
- يتم تحديث السعر الإجمالي بناءً على التخصيصات.

٥. حالات الخطأ:

– الخطأ ١ : اختيار مكون غير متوفر.

– حالة النظام: يعرض رسالة "هذا المكون غير متوفر حاليًا" مع اقتراح بدائل.

– الخطأ ٢ : عدم اختيار مكون أساسي (مثل الصلصة).

– حالة النظام: يعرض رسالة "يجب اختيار صلصة للبيتزا".

٦. الجهات الفاعلة (Actors) :

– العميل (Client)

– النظام (System)

٧. المُحفز(Trigger) :

– قيام العميل باختيار خيار "تخصيص البيتزا" أثناء إنشاء الطلب.

٨. العملية القياسية (Main Flow) :

١. العميل يختار نوع البيتزا الأساسية.

٢. يختار الحجم (صغير، متوسط، كبير).

٣. يختار نوع العجين (رفيع، عادي، سميك).

٤. يضيف أو يزيل المكونات حسب الرغبة.

٥. يعرض النظام السعر النهائي بعد كل تعديل.

٦. العميل يؤكد التخصيصات.

٩.العمليات البديلة (Alternative Flows) :

– إذا كان المكون غير متوفر، يعرض النظام خيارات بديلة أو يسمح للعميل بإلغاء التخصيص.

– إذا لم يتم اختيار مكون أساسي، يمنع النظام الانتقال للخطوة التالية حتى يتم الاختيار.

الحالة الرابعة :

١. الاسم:

تتبع الطلب (Track Order)

٢.الوصف الموجز:

تمكن العميل من تتبع حالة طلبه الحالي ومعرفة المرحلة التي وصل إليها (قيد التحضير، جاهز، قيد التوصيل، مكتمل).

٣. الشرط المسبق:

– يجب أن يكون العميل قد قام بتقديم طلب مسبق.

– يجب أن يكون الطلب مسجلًا في النظام.

٤. الشرط اللاحق:

– يعرض النظام الحالة الحالية للطلب.

– يتم إرسال إشعارات تلقائية عند تغيير الحالة.

٥. حالات الخطأ:

– الخطأ ١ : إدخال رقم طلب غير صحيح.

– حالة النظام: يعرض رسالة "رقم الطلب غير صحيح".

– الخطأ ٢ : محاولة تتبع طلب منتهي.

– حالة النظام: يعرض رسالة "هذا الطلب قد تم تسليمه مسبقًا".

٦. الجهات الفاعلة (Actors) :

– العميل (Client)

– النظام (System)

٧. المُحفز(Trigger) :

– رغبة العميل في معرفة حالة طلبه الحالي.

٨. العملية القياسية (Main Flow) :

١. العميل يسجل الدخول إلى حسابه.

٢. ينتقل إلى قسم "طلباتي".

٣. يختار الطلب المراد تتبعه.

٤. يعرض النظام الحالة الحالية والوقت المتوقع للتسليم.

٩.العمليات البديلة (Alternative Flows) :

– إذا لم يكن العميل مسجلًا، يمكنه تتبع الطلب بإدخال رقم الطلب ورقم الهاتف.

– إذا كان الطلب متأخرًا، يعرض النظام رسالة اعتذار مع تقدير جديد للوقت.

الحالة الخامسة :

١. الاسم :

إدارة الموظفين (Manage Staff)

٢. الوصف الموجز:

تمكن المدير من إضافة موظفين جدد، تعديل بياناتهم، أو حذفهم من النظام، مع تحديد الصلاحيات لكل موظف.

٣. الشرط المسبق :

- يجب أن يكون المستخدم مسجلًا كمدير.
- يجب أن يكون الموظف غير مسجل مسبقًا في حالة الإضافة.

٤. الشرط اللاحق :

- يتم حفظ التغييرات في قاعدة البيانات.
- يتم إرسال إشعار للموظف في حالة إضافته أو تعديل صلاحياته.

٥. حالات الخطأ :

- الخطأ ١ : محاولة إضافة موظف موجود مسبقًا.
- حالة النظام: يعرض رسالة "هذا الموظف مسجل بالفعل".
- الخطأ ٢ : محاولة حذف موظف مرتبط بطلبات نشطة.
- حالة النظام: يعرض رسالة "لا يمكن حذف موظف لديه طلبات نشطة".

٦. الجهات الفاعلة (Actors) :

- المدير (Manager)

– النظام (System)

٧. المُحفز(Trigger) :

– حاجة المتجر إلى موظفين جدد أو تعديل في الصلاحيات.

٨. العملية القياسية (Main Flow) :

١. المدير يسجل الدخول إلى النظام.

٢. ينتقل إلى قسم "إدارة الموظفين".

٣. يختار "إضافة موظف" أو "تعديل موظف موجود".

٤. يدخل البيانات المطلوبة (الاسم، الدور، الصلاحيات).

٥. يؤكد الإجراء.

٦. النظام يحفظ التغييرات ويعرض رسالة نجاح.

٩. العمليات البديلة (Alternative Flows) :

– إذا كان الموظف المراد حذفه لديه طلبات نشطة، يقترح النظام نقل الطلبات إلى موظف آخر قبل الحذف.

– إذا نسي المدير إدخال حقل ضروري، يمنع النظام الحفظ حتى يتم إكمال البيانات.

.....

نبدأ الان برسم مخطط **class diagram** :

المؤلف من:

١.كيانات: وهم:

١ – الزبون (client) : المسؤول عن طلب البيتزا وعرض سجل الطلبات السابقة.

٢ – الطلب (order): يحتوي على معلومات الطلب (السعر الإجمالي، الحالة، التاريخ).

٣ – البيتزا (pizza) : تحتفظ بخصائص البيتزا وهم(الحجم، السعر، الإضافات).

٤- الإضافات (topping): مثل الجبنة او الفطر

٥- الدفع(payment) : معالجة الدفع (المبلغ, التاريخ, الطريقة)

٦- الموظف(employee): يدير الطلبات (تحديث الحالة , اعداد البيتزا)

٢. العلاقات :

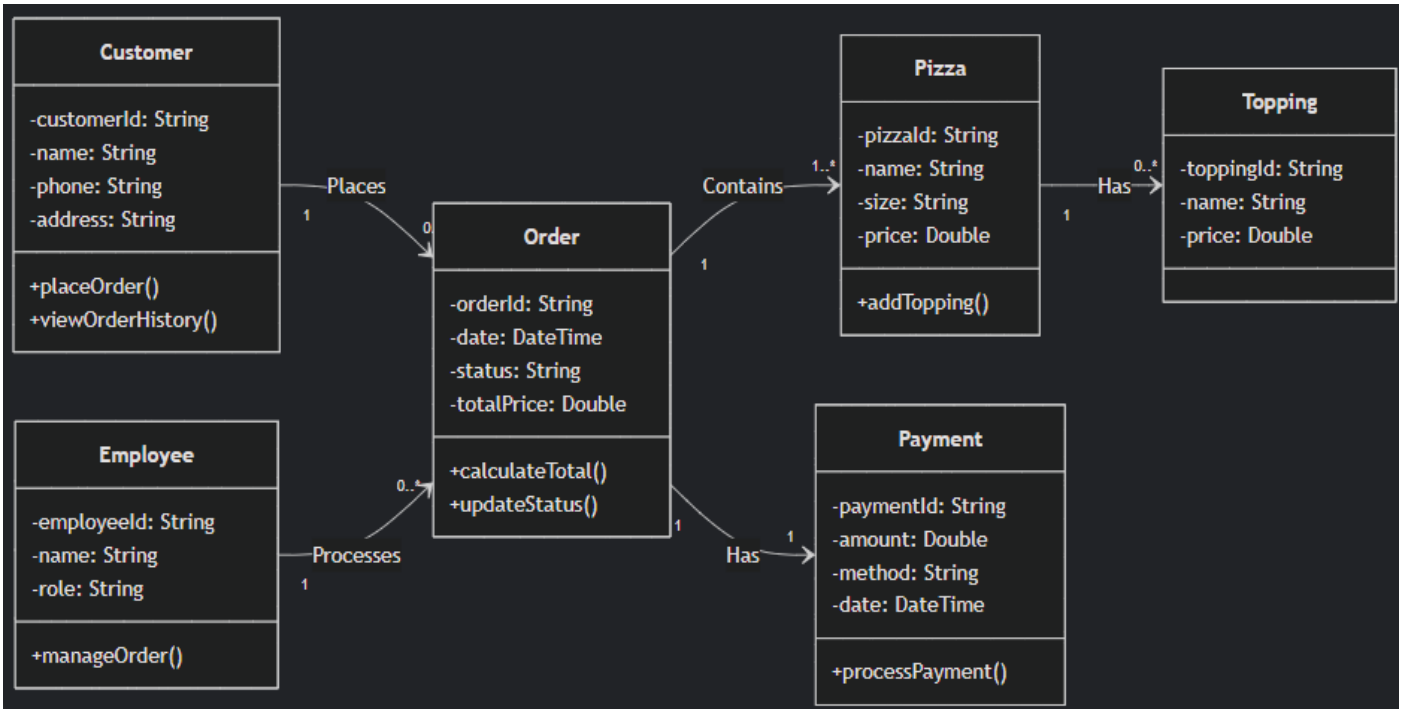
١ – عميل واحد يمكنه عمل عدة طلبات **one _to_many**

٢ – كل طلب يحتوي على واحدة او اكثر من البيتزا **(Composition)**

٣ – كل بيتزا يمكن ان تحتوي على عدة إضافات **(Aggregation)**

٤ – كل طلب له دفعة واحدة **one_to_one**

٥ – موظف واحد يمكنه معالجة عدة طلبات **one _to_many**



شرح للتوابع الموجودة ضمن جميع الكيانات :

اسم الدالة	الوصف
placeOrder()	ينشئ طلبا جديدا ويربطه بالعميل

viewOrderHistory()	يعرض سجل الطلبات السابقة للعميل
calculateTotal()	يحسب السعر الإجمالي للطلب متضمنا الضرائب والاضافات والبيتزا
updateStatus()	غير حالة الطلب (مثال: "جاري التحضير"، "تم التوصيل".
addTopping()	يضيف إضافة الى البيتزا ويحدث السعر
processPayment()	ينفذ عملية الدفع ويسجل التاريخ وطريقة الدفع
manageOrder()	يعدل حالة الطلب او يعين موظفا للتحضير للتوصيل

والان سنبدا بمخطط ERD:

أولا سنتحدث عن الكيانات الموجودة في المخطط:

١-الموظفين:Employee : هنا الكيان مكون من عدة سمات منها الرقم المميز و الاسم والدور ورقم الهاتف و الايميل و كلمة المرور والراتب .

٢-الزبون Customer: الكيان الزبون يتكون من عدة سمات منها الرقم المميز ,الاسم, الهاتف, العنوان, الايميل, كلمة المرور.

٣-الطلب order : سماته :الرقم المميز , موعد الطلب ,السعر الإجمالي , الحالة,طريقة الدفع , حالة الدفع ,عنوان التوصيل .

٤-عنصر الطلب order-Item: له رقم مميز وكمية وملاحظات و سعر كل مكون.

٥- البيتزا pizza: لها رقم مميز و اسم و حجم ووصف ,سعرها الأساسي.

٦-المكون :ingredient : له رقم مميز ,اسم,سعر,وحدة, كمية المخزون.

٧- التوصيل delivery: له رقم مميز وحالة ووقت التوصيل وملاحظات.

٨-الدفع payment : له رقم و طريقة و حالة ووقت الدفع .

٩-المخزون Inventory: رقم , الكمية الموجودة , اخر تعديل على الكميات ,مستوى الطلب .

اما العلاقات الموجودة لدي فهي كالتالي :

١-بين الزبون و الطلب علاقة 1-N لان كل زبون قد يطلب عدة طلبات اما الطلب فهو يعود لزبون واحد.

٢- الطلب وعناصر الطلب١-N الطلب الواحد يحتوي على عدة عناصر مكونة له مثال (الطلب-عدد الطلب-سعره-تاريخ الطلب)

٣- عناصر الطلب و البيتزا 1-N عناصر الطلب تعود الى بيتزا واحدة فقط .

٤- الموظف والطلب 1-N: موظف يقوم بعدة طلبات .

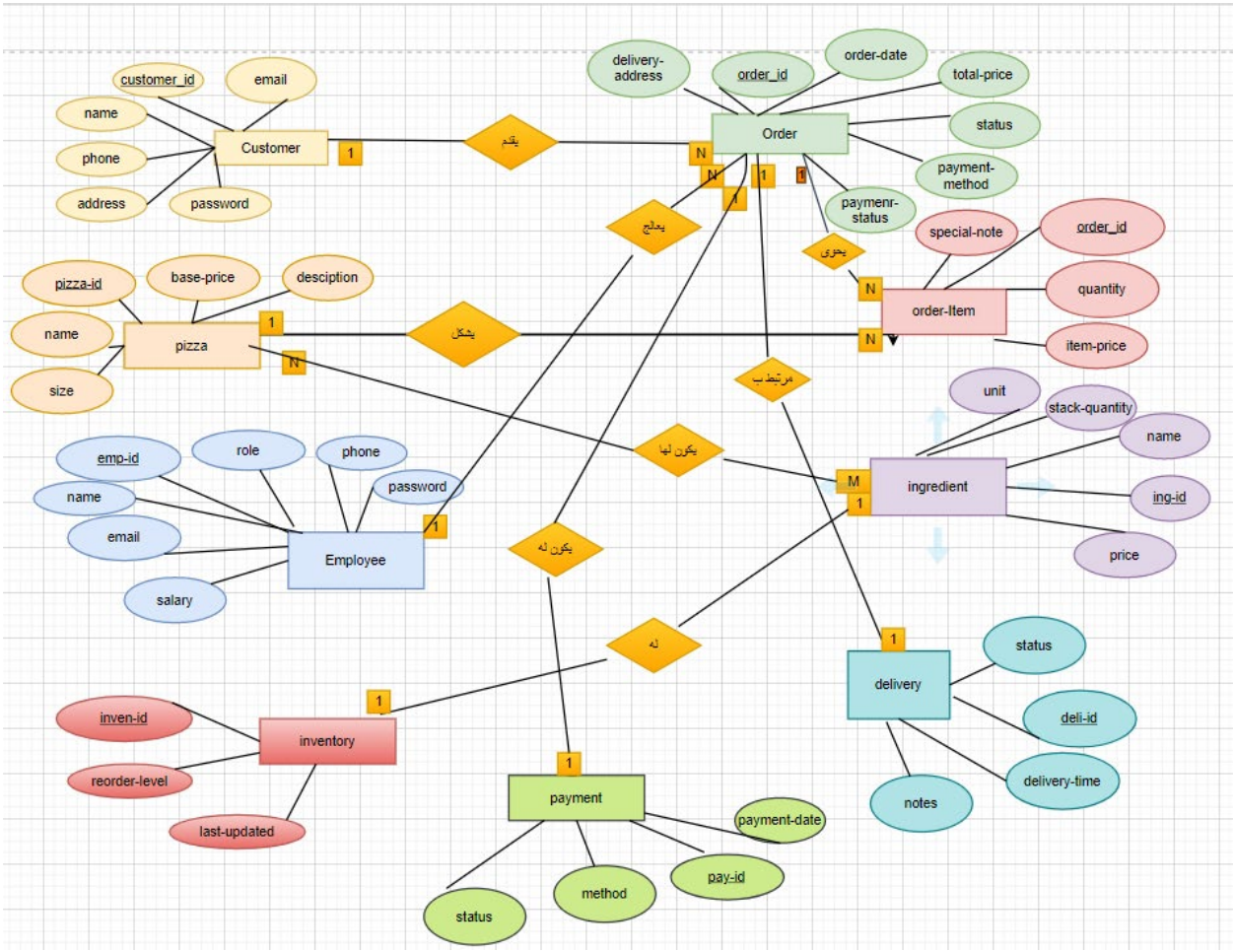
٥- البيتزا : و المكونات n-m لان كل بيتزا لها عدة مكونات وكل مكون يوضع في اكثر من بيتزا(المكونات دقيق,خميرة,طماطم,...)

٦ - المكونات والمخزون :١-١ كل مكون له مخزون واحد فقط.

٧- الدفع والطلب :١- ١ له طلب له فاتورة واحده فقط.

٨- الدلفري و الطلب: ١-١ كل عملية توصيل تكون لطلب واحد فقط.

وبعد عرض كل الكيانات والسمات المكونة منها سنبدأ الان برسم مخطط ERD:



.....

في الجلسة الثالثة تم طلب انشاء وثيقة SRS الان علينا البدأ بتنفيذها:

١-المقدمة: ١-١ الغرض: تهدف هذه الوثيقة الى تحديد المتطلبات الوظيفية وغير الوظيفية لنظام إدارة متجر بيتزا والذي يشمل : ١- إدارة الطلبات والمخزون والموظفين

٢-تتبع المبيعات والتقارير المالية ٣- تسهيل عمليات الدفع والتوصيل .

١-٢ الجمهور المستهدف: *مطورو البرمجيات :لفهم المتطلبات التقنية.

*مديرو المشاريع :لتخطيط الجدول الزمني والموارد .

*المختبرون :لانشاء حالات اختبار بناء على المتطلبات .

*أصحاب المصلحة (مالك المتجر):للتاكد من تغطية احتياجات العمل.

١-٣ نطاق المنتج: النظام يدعم المتطلبات عبر الانترنيت والهاتف .

-يتكامل مع أنظمة الدفع الالكترونية ونقاط المبيع .

-يستخدم داخليا من قبل الموظفين (طهاة,سائقين,مديرين).

١-٤ التعاريف والمختصرات:

التعريف	المصطلح
وثيقة مواصفات متطلبات البرمجيات	SRS
نظام نقاط البيع point of sale	POS
واجهة برمجة المتطلبات	API

٢-الوصف العام :

٢-١ احتياجات المستخدم :

الاحتياجات	المستخدم
تقديم طلب ,تخصيص بيتزا,دفع الكتروني\نقدي.	العميل
تسجيل طلبات معالجة الدفع	امين الصندوق
استلام الطلبات ,تحديث حالة التحضير	الطاهي

تتبع الطلبات الجاهزة وتوصيلها	السائق
مراقبة المخزون وإصدار تقارير مالية	المدير

٢-٢ الافتراضات والتبعيات : الافتراضات :

-اتصال بالانترنت متاح للطلبات عبر الويب.

- متوافقة مع النظام (أجهزة,POS الهواتف).

التبيعات : - بوابة دفع الكتروني (paypal,Stripe)

-قاعدة بيانات (Mysql,MongoDB).

٣المتطلبات الوظيفية

٣-١ المتطلبات الوظيفية :

الوصف	الوظيفة
ادخال بيانات العميل ,اختيار البيتزا,الحجم والاضافات	انشاء طلب جديد
احتساب السعر بناء على المكونات+ضرائب+رسوم توصيل	حساب السعر الاجمالي
تغيير الحالة(قيد التحضير الى جاهز ثم مكتمل)	تحديث حالة الطلب
تتبع المكونات ارسال تنبيه عند نفاد الكمية	إدارة المخزون
دعم الدفع النقدي|الالكتروني وإصدار وصل	تسجيل الدفع
تقارير المبيعات اليومية ,تحليل الأداء.	اصدار تقارير

٣-٢ المتطلبات غير الوظيفية:

الوصف	النوع
تشفير بيانات الدفع باستخدام ssl/tls.	الأمان
تحمل ٢٠٠ طلب متزامن وقت استجابة اقل من ٥ ثوان.	الأداء

دعم أجهزة اندرويد وios ومتصفحات كروم وfirefox.	التوافقية
تصميم النظام ليتحمل زيادة الطلبات بنسبة ٢٠٪ سنويا.	قابلية التوسع

٤مخططات النظام :

٤-١ مخطط use case diagram: في الصفحة ٢ يوضح تفاعل المستخدمين مع النظام (العميل,الموظف,المدير).

٤-٢ مخطط class diagram: يظهر الكيانات الرئيسية (الطلب,البيتزا,الدفع) وعلاقاتها.

٥_التسليم والموافقة:

الموافقة النهائية :توقع من قبل : ١- مدير المشروع:_____

٢-ممثل العميل:_____

تاريخ التسليم: <التاريخ>

الخاتمة : تمثل هذه الوثيقة الخطة الشاملة لتطوير نظام إدارة متجر بيتزا، مع ضمان تلبية جميع المتطلبات الوظيفية وغير الوظيفية. ستكون مرجعًا أساسيًا لفريق التطوير والاختبار وأصحاب المصلحة طوال دورة حياة المشروع.

نبدأ الان مشروعنا بانشاء تطبيق api لنظام متجر البيتزا الذي تحدثنا عنه مع توثيق كامل له :

١- اعداد البيئة والمشروع:

إنشاء البيئة الافتراضية وتفعيلها:

```
python -m venv virtual
virtual \Scripts\activate
```

تثبيت الحزم المطلوبة:

```
pip install django
pip install djangorestframework
pip django-filter
```

٢-انشاء المشروع والتطبيق :

```
django-admin startproject project
cd project
python manage.py migrate
python manage.py startapp Task
```

٣- تكوين الاعدادات Task/settings.py :

```
INSTALLED_APPS = [
     . . .
    'rest_framework',
    'Task',
    'django_filters',
     . . .
]
REST_FRAMEWORK = {
'DEFAULT_FILTER_BACKENDS':
['django_filters.rest_framework.DjangoFilterBackend']
}
```

٤-نماذج البيانات Task/models.py :

```
from django.db import models

class Task(models.Model):
    name = models.CharField(max_length=200)
    size = models.CharField(max_length=100, choices=[
        ('S', 'Small'),
        ('M', 'Medium'),
        ('L', 'Large'),
        ('XL', 'Extra Large'),
    ])
    price = models.DecimalField(max_digits=5, decimal_places=2)
    def __str__(self) :
        return self.name
```

٥-التسلسل Task /serializers.py :

```
from rest_framework import serializers
from .models import Task
class TaskSerializer(serializers.ModelSerializer):
    class Meta:
        model = Task
        fields = '__all__'
```

٦- واجهات العرض Task/view.py :

```
from rest_framework import viewsets
from .models import Task
from .serializers import TaskSerializer
from django_filters.rest_framework import DjangoFilterBackend
class TaskViewSet(viewsets.ModelViewSet):
    queryset = Task.objects.all()
    serializer_class = TaskSerializer
    filter_backends = [DjangoFilterBackend]
    filterset_fields = ["size"]
    def perform_create(self, serializer):
        serializer.save(customer=self.request.user)
    def get_queryset(self):
        return Task.objects.filter(name=self.request.user)
```

٧-المسارات project/urls.py :

```
from django.contrib import admin
from django.urls import path, include
from Task.views import TaskViewSet
from rest_framework.routers import DefaultRouter
router = DefaultRouter()
router.register('orders', TaskViewSet)
urlpatterns = [
path('api/', include(router.urls)),
path('api-auth/', include('rest_framework.urls')),]
```

٨-نقوم بالترحيل وتشغيل الخادم :

```
python manage.py makemigrations
```

```
python manage.py migrate
python manage.py createsuperuser
python manage.py runserver
```