

المشروع الأول لمادة تحليل النظم المالية

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 :

- 1- تبسيط عمليات الدفع والطلب
- 2- إدارة المخزون والموظفين
- 3- تتبع الطلبيات وتحسين خدمة العملاء
- 4- 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.

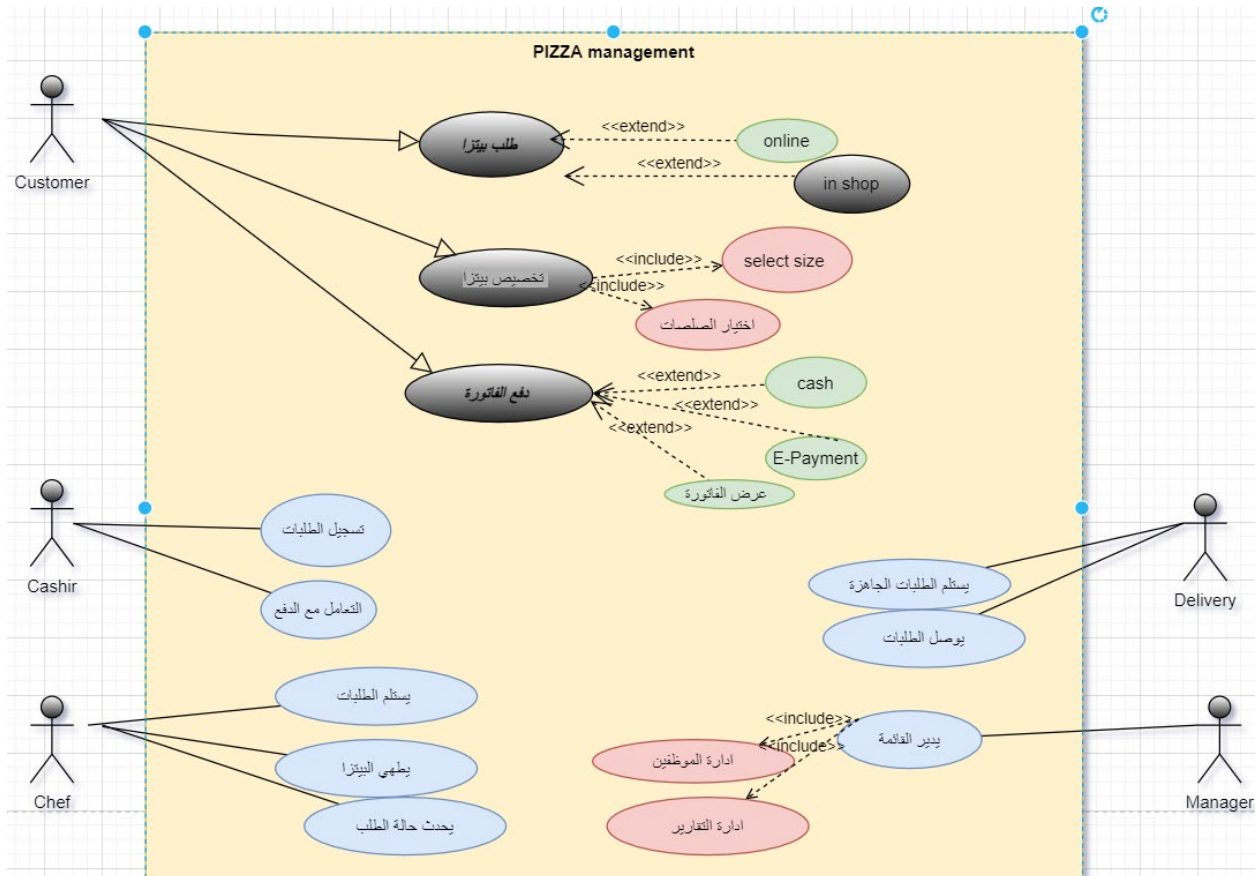
المشروع الأول لمادة تحليل النظم المالية

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



المشروع الأول لمادة تحليل النظم المالية

المتطلبات الوظيفية :

1- 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

2- إدارة المخزون : Inventory Management

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

3- إدارة الموظفين : Staff Management

المشروع الأول لمادة تحليل النظم المالية

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

4- التقارير والتحليلات : Reporting :

الوظيفة	الوصف
تقارير المبيعات اليومية	معرفة الطلبات الناجحة والملغاة
تحليل الاداء	تحديد اكثر المبيعات مبيعاً واوقات الذروة

المتطلبات غير الوظيفية :

1- الأداء : تحمل 100 طلب متزامنا خلال ساعات الذروة، وقت الاستجابة اقل

من 5 دقائق لأي عملية.

2- الأمان : تشفير بيانات الدفع.

3- التوافقية : دعم متصفحات حديثة و العمل على أجهزة حديثة للسائقين لتحديد

موقع الطلب.

.....

المشروع الأول لمادة تحليل النظم المالية

والان نبدأ بتوصيف حالتني استخدام:

1. الاسم:

تقديم طلب جديد (Place Order)

2. الوصف الموجز:

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

3. الشرط المسبق:

- يجب أن يكون العميل مسجلاً في النظام أو يقوم بإدخال بياناته الأساسية (الطلبات الجديدة).

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

- يجب أن يكون المخزون متاحاً للمكونات المطلوبة.

4. 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).

المشروع الأول لمادة تحليل النظم المالية

5. حالات الخطأ:

- الخطأ 1 :عدم توفر أحد المكونات (مثل نوع معين من الجبن).
- الخطأ 2 :فشل عملية الدفع (إذا كان الدفع إلكترونياً).
- الخطأ 3 :إدخال عنوان توصيل غير صحيح أو خارج نطاق التوصيل.

6. حالة النظام عند حدوث خطأ:

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

7. الجهات الفاعلة (Actors) :

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

8. المُحفز (Trigger) :

- يقوم العميل بالنقر على زر "اطلب الآن" في التطبيق أو الموقع.
- يتصل العميل هاتفياً بالمتجر لطلب بيتزا.

المشروع الأول لمادة تحليل النظم المالية

9. 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.

10. 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).

المشروع الأول لمادة تحليل النظم المالية

نبدأ بحالة الاستخدام التالية :

1. الاسم:

دفع الفاتورة (Pay Bill)

2. 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.

3. 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).

4. الشرط اللاحق:

المشروع الأول لمادة تحليل النظم المالية

- 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".

5. حالات الخطأ:

- الخطأ 1 : فشل عملية الدفع الإلكتروني (نقص الرصيد/انتهاء البطاقة).
- الخطأ 2 : انقطاع الاتصال بالإنترنت أثناء الدفع الإلكتروني.
- الخطأ 3 : رفض جهاز الصراف الآلي أو نقطة البيع للعملية.
- الخطأ 4 : إدخال بيانات دفع غير صحيحة (مثل CVV خاطئ).

6. 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.

المشروع الأول لمادة تحليل النظم المالية

- يسجل محاولة الدفع الفاشلة في سجل النظام.

7. 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.

8. المُحفز (Trigger) :

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

9. 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:

المشروع الأول لمادة تحليل النظم المالية

- **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.

10. العمليات البديلة (Alternative Flows):

- إذا فشل الدفع الإلكتروني (الخطأ 1 أو 4):
- يعرض النظام: "فشل عملية الدفع، الرجاء التحقق من البيانات والمحاولة مرة أخرى."
- يسمح للعميل بإدخال بيانات جديدة أو اختيار طريقة دفع مختلفة.

المشروع الأول لمادة تحليل النظم المالية

• إذا انقطع الاتصال (الخطأ 2):

○ 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.

.....

نبدأ الان برسم مخطط **class diagram**:

المؤلف من:

1. كيانات: وهم:

1- الزبون (client): المسؤول عن طلب البيتزا وعرض سجل الطلبات السابقة.

المشروع الأول لمادة تحليل النظم المالية

2- الطلب (order) : يحتوي على معلومات الطلب (السعر الإجمالي ، الحالة ، التاريخ).

3- البيتزا (pizza) : تحتفظ بخصائص البيتزا وهم (الحجم ، السعر ، الإضافات).

4- الإضافات (topping) : مثل الجبنة او الفطر

5- الدفع (payment) : معالجة الدفع (المبلغ ، التاريخ ، الطريقة)

6- الموظف (employee) : يدير الطلبات (تحديث الحالة ، اعداد البيتزا)

2. العلاقات :

1- عميل واحد يمكنه عمل عدة طلبات one_to_many

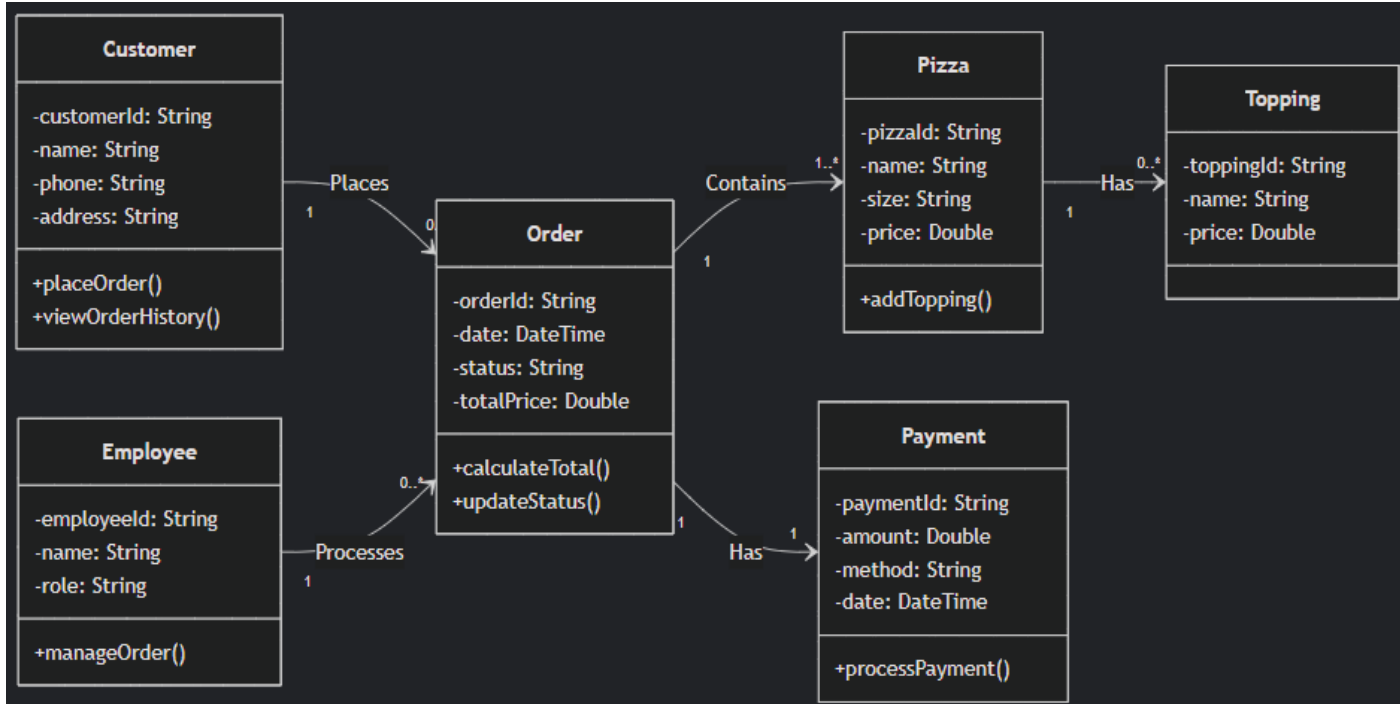
2- كل طلب يحتوي على واحدة او اكثر من البيتزا (Composition)

3- كل بيتزا يمكن ان تحتوي على عدة إضافات (Aggregation)

4- كل طلب له دفعة واحدة one_to_one

5- موظف واحد يمكنه معالجة عدة طلبات one_to_many

المشروع الأول لمادة تحليل النظم المالية



شرح للتوابع الموجودة ضمن جميع الكيانات :

الوصف	اسم الدالة
ينشئ طلبا جديدا ويربطه بالعميل	placeOrder()
يعرض سجل الطلبات السابقة للعميل	viewOrderHistory()
يحسب السعر الإجمالي للطلب متضمنا الضرائب والاضافات والبيتزا	calculateTotal()

المشروع الأول لمادة تحليل النظم المالية

غير حالة الطلب (مثال: "جاري التحضير"، "تم التوصيل".	updateStatus()
يضيف إضافة الى البيتزا ويحدث السعر	addTopping()
ينفذ عملية الدفع ويسجل التاريخ وطريقة الدفع	processPayment()
يعدل حالة الطلب او يعين موظفا للتحضير للتوصيل	manageOrder()

والان سنبدأ بمخطط ERD:

أولا سنتحدث عن الكيانات الموجودة في المخطط:

- 1-الموظفين: **Employee** : هنا الكيان مكون من عدة سمات منها الرقم المميز و الاسم والدور ورقم الهاتف و الايميل و كلمة المرور والراتب .
- 2-الزبون **Customer**: الكيان الزبون يتكون من عدة سمات منها الرقم المميز ,الاسم, الهاتف, العنوان, الايميل, كلمة المرور.
- 3-الطلب **order** : سماته :الرقم المميز , موعد الطلب ,السعر الإجمالي , الحالة,طريقة الدفع , حالة الدفع ,عنوان التوصيل .
- 4-عنصر الطلب **order-Item**: له رقم مميز وكمية وملاحظات و سعر كل مكون.
- 5- البيتزا **pizza**: لها رقم مميز و اسم و حجم ووصف ,سعرها الأساسي.
- 6-المكون **ingredient**: له رقم مميز ,اسم,سعر,وحدة, كمية المخزون.
- 7- التوصيل **delivery**: له رقم مميز وحالة ووقت التوصيل وملاحظات.
- 8-الدفع **payment** : له رقم و طريقة و حالة ووقت الدفع .

المشروع الأول لمادة تحليل النظم المالية

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

اما **العلاقات** الموجودة لدي فهي كالتالي :

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

2- الطلب وعناصر الطلب 1-N الطلب الواحد يحتوي على عدة عناصر مكونة له مثال (الطلب-عدد الطلب-سعره-تاريخ الطلب)

3- عناصر الطلب و البيئزا 1-N عناصر الطلب تعود الى بيئزا واحدة فقط .

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

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

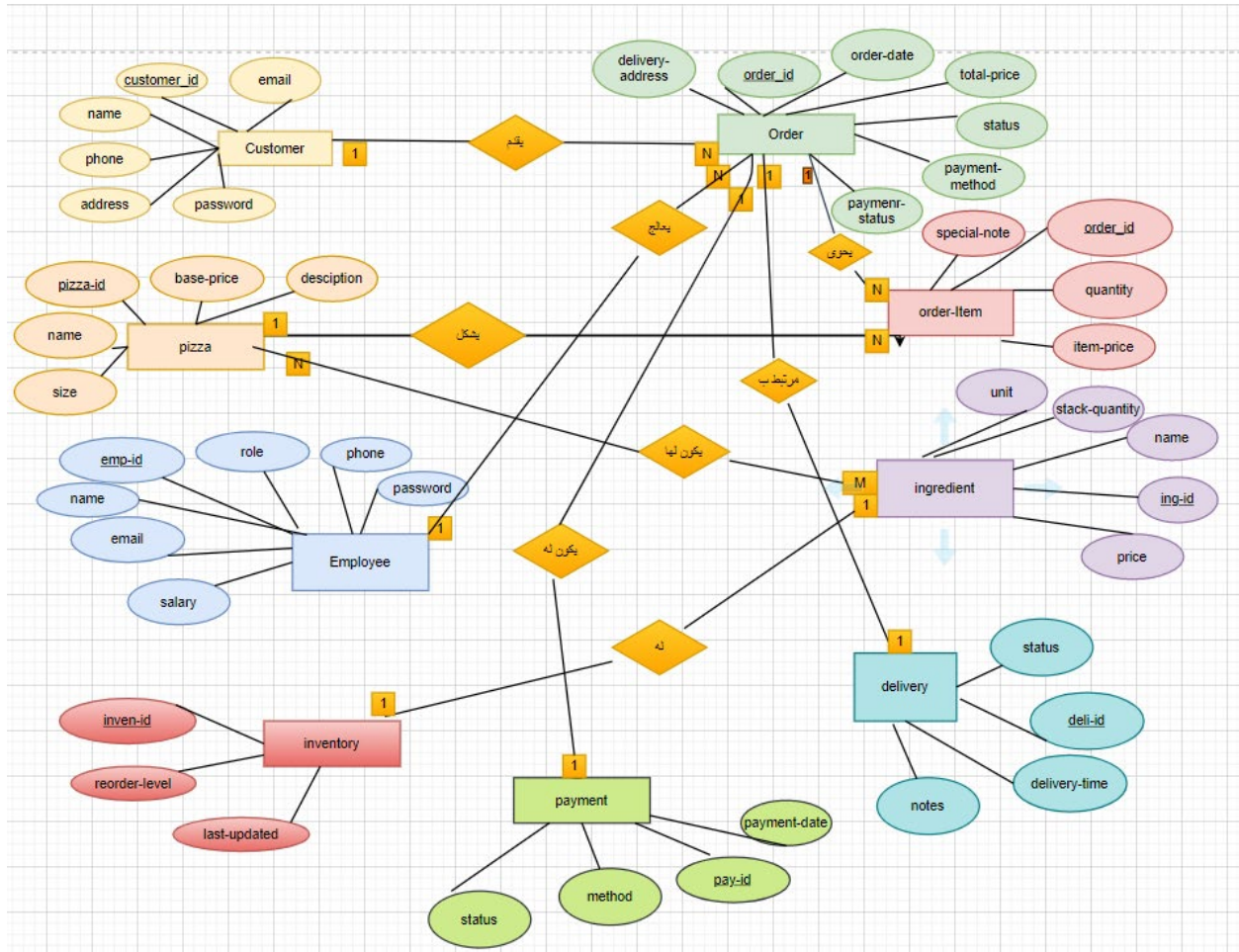
6 - المكونات والمخزون : 1-1 كل مكون له مخزون واحد فقط.

7- الدفع والطلب : 1- 1 له طلب له فاتورة واحد فقط.

8- الدلفري و الطلب : 1-1 كل عملية توصيل تكون لطلب واحد فقط.

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

المشروع الأول لمادة تحليل النظم المالية



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

1-المقدمة: 1-1 الغرض: تهدف هذه الوثيقة الى تحديد المتطلبات الوظيفية وغير الوظيفية لنظام إدارة متجر بيتزا والذي يشمل : 1- إدارة الطلبات والمخزون والموظفين

2-تتبع المبيعات والتقارير المالية 3- تسهيل عمليات الدفع والتوصيل .

1-2 الجمهور المستهدف: *مطورو البرمجيات :لفهم المتطلبات التقنية.

*مديرو المشاريع :لتخطيط الجدول الزمني والموارد .

*المختبرون :لانشاء حالات اختبار بناء على المتطلبات .

المشروع الأول لمادة تحليل النظم المالية

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

1-3 نطاق المنتج: النظام يدعم المتطلبات عبر الانترنت والهاتف .

-يتكامل مع أنظمة الدفع الالكترونية ونقاط المبيع .

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

1-4 التعاريف والمختصرات:

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

2-الوصف العام :

1-2 احتياجات المستخدم :

الاحتياجات	المستخدم
تقديم طلب , تخصيص بيتزا, دفع الالكتروني/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

2-2 الافتراضات والتبعيات : الافتراضات :

-اتصال بالانترنت متاح للطلبات عبر الويب.

- متوافقة مع النظام (أجهزة, POS, الهواتف).

المشروع الأول لمادة تحليل النظم المالية

التبيعات : - بوابة دفع الكتروني (paypal,Stripe)

-قاعدة بيانات (Mysql,MongoDB).

3المتطلبات الوظيفية

3-1 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

3-2 المتطلبات غير الوظيفية:

الوصف	النوع
تشفير بيانات الدفع باستخدام ssl/tls.	الأمان
تحمل 200 طلب متزامن وقت استجابة اقل من 5 ثوان.	الأداء
دعم أجهزة اندرويد و ios ومتصفحات كروم وfirefox.	التوافقية
تصميم النظام ليتحمل زيادة الطلبات بنسبة 20% سنويا.	قابلية التوسع

المشروع الأول لمادة تحليل النظم المالية

4 مخططات النظام :

4-1 مخطط use case diagram: في الصفحة 2 يوضح تفاعل المستخدمين مع النظام (العميل, الموظف, المدير).

4-2 مخطط class diagram: يظهر الكيانات الرئيسية (الطلب, البيئزا, الدفع) وعلاقاتها.

5_ التسليم والموافقة:

الموافقة النهائية: توقع من قبل: 1- مدير المشروع: _____
2- 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 مرجعاً أساسياً لفريق التطوير والاختبار وأصحاب المصلحة طوال دورة حياة المشروع.

انتهت الجلسة 3 _____

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

المشروع الأول لمادة تحليل النظم المالية

1- اعداد البيئة والمشروع:

إنشاء البيئة الافتراضية وتفعيلها:

```
python -m venv pizza_env
```

```
pizza_env\Scripts\activate # Windows
```

تنصيب الحزم المطلوبة:

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

2- إنشاء المشروع والتطبيق :

```
django-admin startproject PizzaShop
```

```
cd PizzaShop
```

```
python manage.py startapp orders
```

3- تكوين الاعدادات :

```
INSTALLED_APPS =
```

```
...
```

```
'rest_framework',
```

```
'orders',
```

```
'django_filters',
```

```
[
```

```
REST_FRAMEWORK =
```

```
'DEFAULT_FILTER_BACKENDS':
```

```
'django_filters.rest_framework.DjangoFilterBackend']
```

المشروع الأول لمادة تحليل النظم المالية

4- نماذج البيانات: `order/models.py`

```
from django.db import models

class Pizza(models.Model):
    name = models.CharField(max_length=100)
    size = models.CharField(max_length=1, choices=SIZE_CHOICES)
    price = models.DecimalField(max_digits=5, decimal_places=2)
    def __str__(self):
    return name;
}
```

5- التسلسل: `orders/serializers.py`

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

6- واجهات العرض: `order/view.py`

```
from rest_framework import viewsets
from .models import Order
```

المشروع الأول لمادة تحليل النظم المالية

```
from serializers import OrderSerializer
from django_filter.rest_framework import DjangoFilterBackend
class OrderViewSet(viewsets.ModelViewSet):
    queryset = Order.objects.all()
    serializer_class = OrderSerializer
    filter_backends = [DjangoFilterBackend]
    filterset_fields = ['status', 'pizza_size']
    def perform_create(self, serializer):
        serializer.save(customer=self.request.user)
```

6-المسارات: pizzashop/urls.py

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

7-نقوم بالترحيل وتشغيل الخادم :

python manage.py makemigrations

python manage.py migrate

المشروع الأول لمادة تحليل النظم المالية

```
python manage.py createsuperuser
```

```
python manage.py runserver
```