

Requirements Specification for EatOut Eorder

**July 2018
Version 0.1**

**Prepared By
Tango**

Change History

Date	Version	Description	Updated By
2 July 2018	0.1	Initial Draft	Tango

Document Approvals

Name	Role	Signature

Table of Contents

1 Introduction.....	4
1.1 Purpose.....	4
1.2 Document Conventions.....	4
1.3 Project Scope.....	4
1.4 References.....	4
2 System Description.....	4
3 Functional Requirements.....	4
3.1 System Features.....	4
3.1.1 System Feature 1.....	5
3.1.2 System Feature 2.....	5
3.2 Use Cases.....	5
3.2.1 Use Case Diagrams.....	5
3.2.2 Use Case 1.....	5
3.2.3 Use Case 2.....	5
3.3 Entity Relationship Diagrams.....	5
3.4 Data Dictionary.....	6
3.4.1 Entity 1.....	6
3.4.2 Entity 2.....	6
4 External Interface Requirements.....	6
5 Technical Requirements (Non functional).....	6
5.1 Performance.....	6
5.2 Scalability.....	6
5.3 Security.....	6
5.4 Maintainability.....	6
5.5 Usability.....	6
5.6 Multi lingual Support.....	6
5.7 Auditing and Logging.....	6
5.8 Availability.....	6
6 Open Issues.....	7

1 Introduction

1.1 Purpose

本文档是描述 Eat Out E-order 扫码点餐系统的需求说明，读者是本系统的用户，开发人员，以及测试人员。

1.2 Document Conventions

UC: 用例 (Use Case, 以下简称 UC)

DM: 领域模型 (Domain Model, 以下简称 DM)

ER: 实体联系 (Entity Relationship, 以下简称 ER)

Eat Out E-order 扫码点餐系统 (以下简称 EOEO 系统)

1.3 Project Scope

本文档包括系统概述，功能需求，性能需求，外部接口需求，总体约束特性和软件质量特征等几个部分。EOEO 系统的商户端用户定位为小型餐饮店，提供基本的电子餐馆管理功能，暂不考虑复杂的业务数据处理和营销功能。客户端用户定位为日常单人用餐，或多人 AA 制用餐。可以利用数据分析技术，根据顾客以往的消费记录和评价来推断顾客的喜好，实现更人性化的餐馆和菜品推荐。AI 技术发展也给改进这类产品带来了新机会，我们可以在我们的业务扩展模块中，考虑加入一些更智能的交互，如语音交互，给顾客更好的用餐体验。

1.4 References

IT 项目管理 —— 主编：毛志雄 中国铁道出版社

UML 和模式应用 —— Craig Larman 机械工业出版社

830-1984——IEEE Guide to Software Requirements Specifications. 1984

830-1998——IEEE Recommended Practice for Software Requirements Specifications. 1998

2 System Description

随着“互联网+”的热潮，餐饮“互联网+”的概念也越来越火爆，从传统点餐，买单到网络点餐，扫码付款，再到现在的扫码点餐，手机扫描二维码点餐系统已经成为餐饮行业的未来趋势，发展空间巨大。

本项目是一个全新的项目，产品独立且全部内容自含。EOEO 系统可显示丰富的电子菜单信息，实时更新菜单界面，准确显示当前供给的菜品内容，以及每个顾客的订单记录和当前订单进展。EOEO 系统通过自动化的扫码点餐流程，避免了点餐，结账等过程的排队等候，让顾客轻松高效地完成点餐过程，为顾客节省了时间和精力。EOEO 系统有效地为餐厅节省人力成本，提高顾客点餐效率的同时也提高了餐厅翻台率。EOEO 系统的个性化美食推荐能提供人性化的推荐服务，增强用户友好度，增加用户黏性。

本系统有 3 类用户——顾客，商家，服务员。顾客扫码登录后可进入相应的餐厅界面，

选择餐品加入购物车，下订单，支付订单后即可等待用餐。商家登陆后可对其经营的网上餐厅进行管理，可以增加，删除或修改各种菜品的信息。服务员根据订单信息为顾客送餐到指定餐桌。

3 Functional Requirements

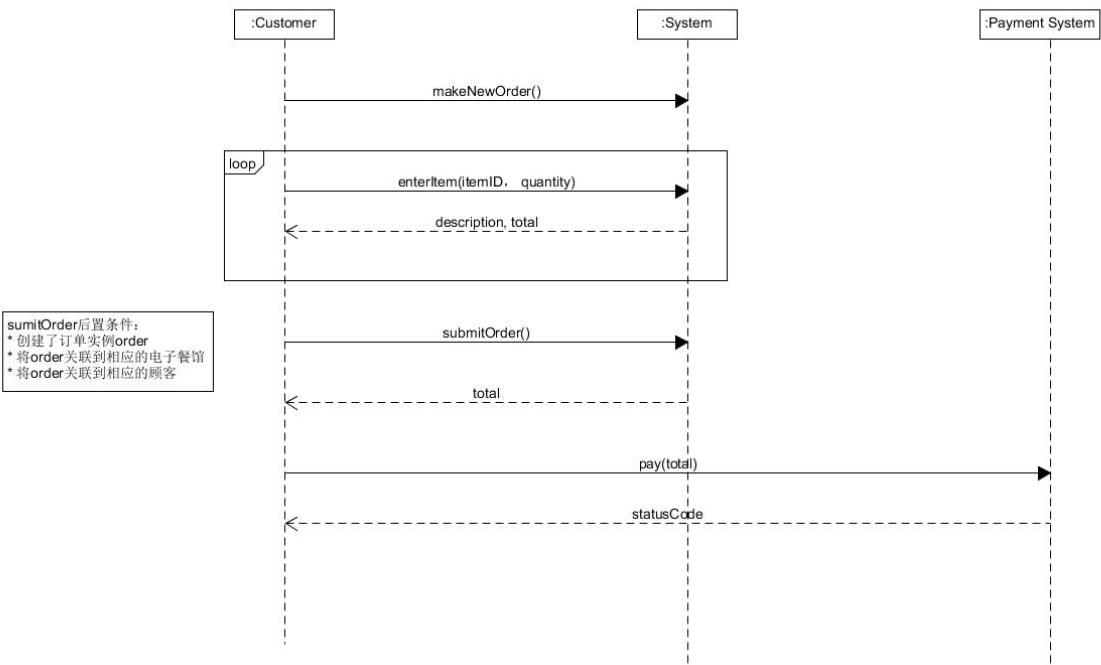
3.1 System Features

本系统主要用 Python，Javascript 语言编写，运行在 Windows 平台或 Android 手机端上。本系统主要包括扫码网页登录，访问数据库两大部分，需要利用到服务器和数据库管理软件，初步设想用 PC 机建立服务器，使用 MySQL 数据库。本系统面向顾客，商家和服务员三类用户，不要求有特殊能力和经验，只需要按页面提示操作即可。

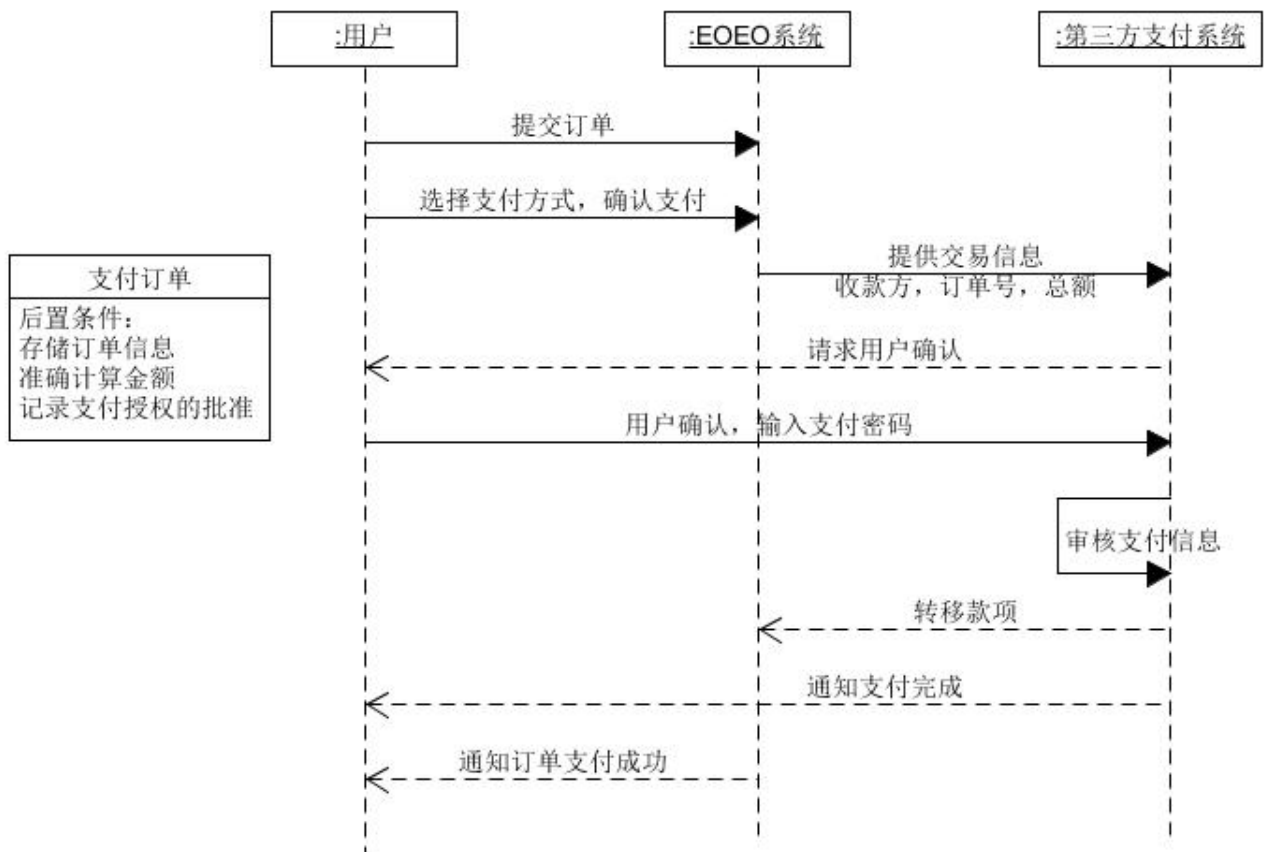
3.1.1 System Feature 1

客户端系统顺序图

系统顺序图——点餐



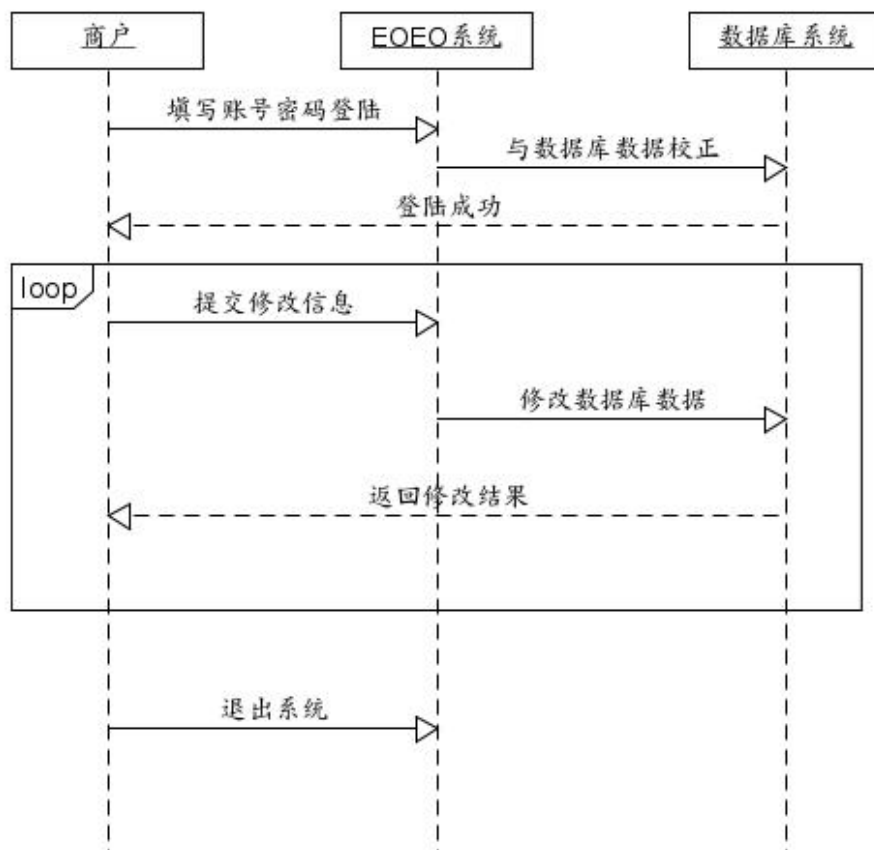
系统顺序图——支付订单



3.1.2 System Feature 2

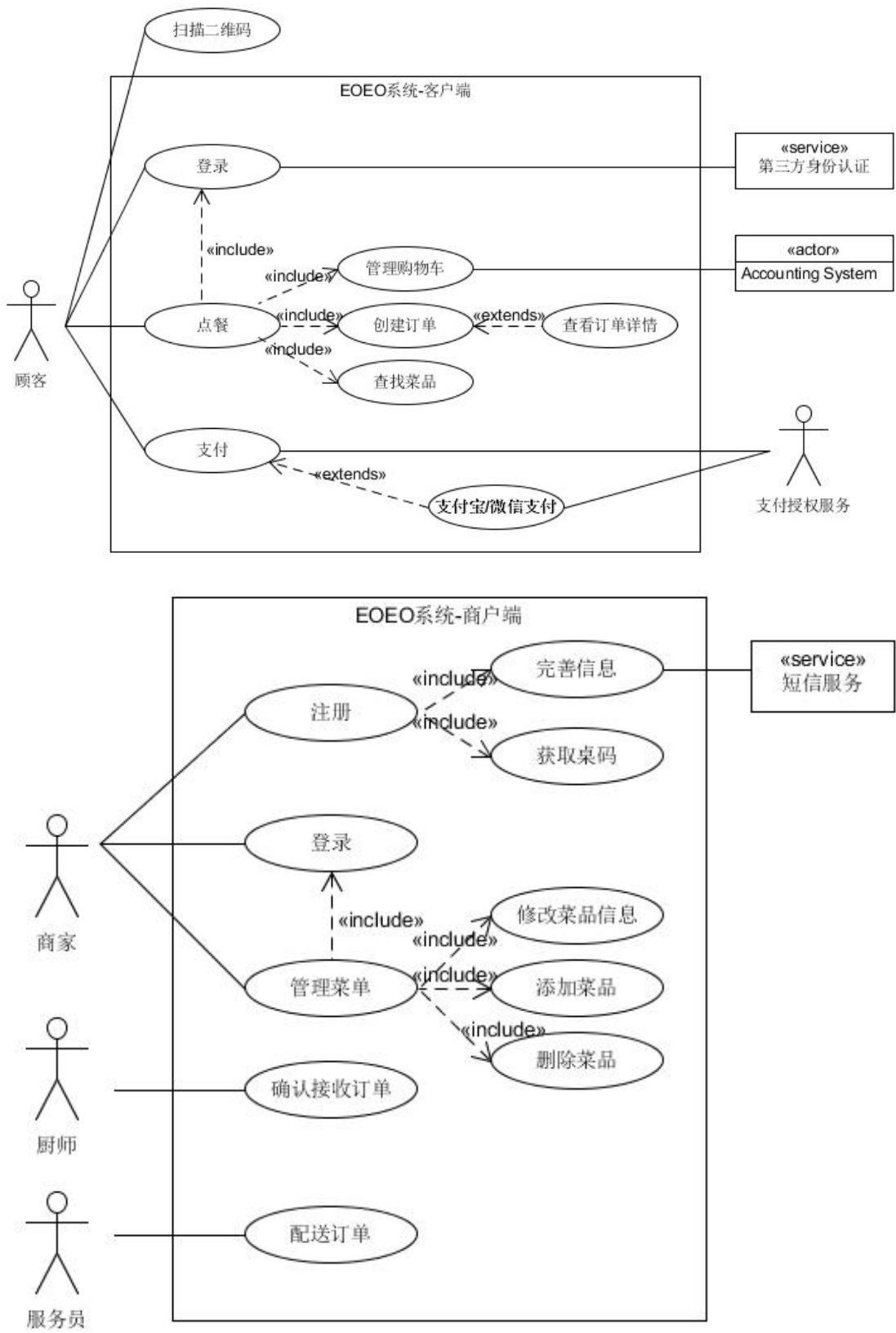
商户型系统顺序图

系统顺序图——商户修改菜单



3.2 Use Cases

3.2.1 Use Case Diagrams



3.2.2 Use Case 1

<This provides a detailed description of the use case. Usually it is captured in the following table format. Add more rows or removing rows depending on your specific requirement.>

ID	UC1-点餐
Description	顾客浏览菜单，查找菜品，将菜品加入购物车，创建订单，提交订单，选择支付方式，支付订单，订单完成后可查看订单详情。
Actors	顾客，商家，服务员，支付授权服务
Preconditions	顾客选择通过扫描二维码来点餐。
Basic Steps	<ol style="list-style-type: none">1. 顾客扫描餐桌上二维码，标识自己的身份。2. 顾客浏览菜单信息。3. 顾客将所选餐品加入购物车。4. 系统记录购物车内餐品信息，显示当前累计额。价格通过一组价格规则计算。5. 顾客重复 2~4 步，直到所需餐品选择完毕。6. 顾客下订单。7. 系统显示订单餐品信息和总额。8. 顾客确认订单，完善订单信息，提交订单。9. 顾客付款，系统处理支付。10. 系统记录完整订单信息，并将交易和支付信息发送到厨房的订单处理系统。显示当前订单状态。
Alternate Steps	<p>*a. 商家/老板在任意时刻要求进行电子餐馆管理操作：</p> <ol style="list-style-type: none">1. 系统进入管理授权模式。2. 商家执行某一管理模式的操作。例如：降低餐品价格，标明某项餐品已售罄，添加新的餐品信息等。3. 系统变更客户端菜单信息，回复到厨房授权模式。 <p>*b. 网络在任意时刻发生故障：</p> <ol style="list-style-type: none">1. 系统向顾客报告错误，并建议他退回到前一步。2. 顾客选择退出此用例，或者重新再试。 <p>3a. 当餐品有多种信息需要确认时（如口味，温度，加料等），先选择规</p>

	<p>格再将餐品加入购物车。</p> <p>3b. 当顾客发生点餐错误时，将误点餐品移出购物车。</p> <p>6-8a. 顾客选择取消订单：</p> <ol style="list-style-type: none"> 1. 系统取消订单交易。 <p>6-8b. 顾客延迟订单交易：</p> <ol style="list-style-type: none"> 1. 系统记录订单交易信息，保存购物车状态，使其能在登录中恢复操作。 2. 系统显示用于恢复交易的餐品项目和订单交易 ID。 <p>9a. 支付宝/微信支付：</p> <ol style="list-style-type: none"> 1. 系统向外部支付授权系统发送支付授权请求，并请求批准该支付。 <ol style="list-style-type: none"> 1a. 系统检测到与外部系统协作时的故障： <ol style="list-style-type: none"> 1. 系统向顾客提示错误，请求顾客更换支付方式。 2. 系统收到批准支付的应答，提示顾客输入支付密码。 <ol style="list-style-type: none"> 2a. 系统收到拒绝支付的应答： <ol style="list-style-type: none"> 1. 系统向顾客提示支付被拒绝。 2. 系统请求顾客更换支付方式。 2b. 应答超时： <ol style="list-style-type: none"> 1. 系统显示支付请求超时。 2. 提示顾客重试，或者更换支付方式。 3. 顾客输入支付密码。 <ol style="list-style-type: none"> 3a. 系统提示账户余额不足，请求顾客更换支付方式。 4. 系统记录支付信息，包括支付批准。
Special Requirements	<ul style="list-style-type: none"> • 商家使用大尺寸平面显示器触摸屏 UI。文本信息可见距离为 1 米。 • 90%的授权响应时间小于 10 秒。 • 希望在访问远程服务失败的情况下具有一定的恢复功能。
Pending Issues	<ul style="list-style-type: none"> • 研究远程服务的恢复问题。

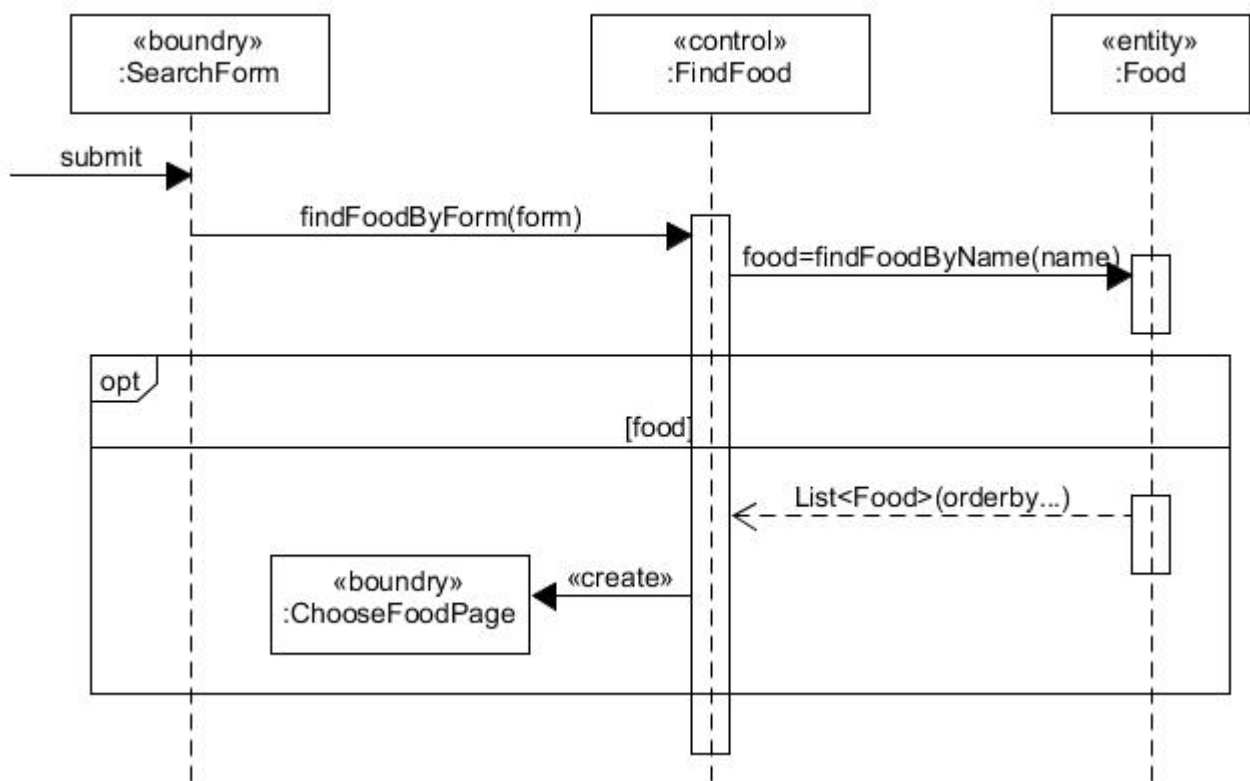
	<ul style="list-style-type: none"> • 针对不同的业务需求怎样进行定制？ • 顾客在支付订单后，是否可以呼叫服务，更改订单信息？
Post-conditions	存储交易信息。记录支付授权的批准。远程云服务站点认可此次点餐事件。

3.2.3 Use Case 2

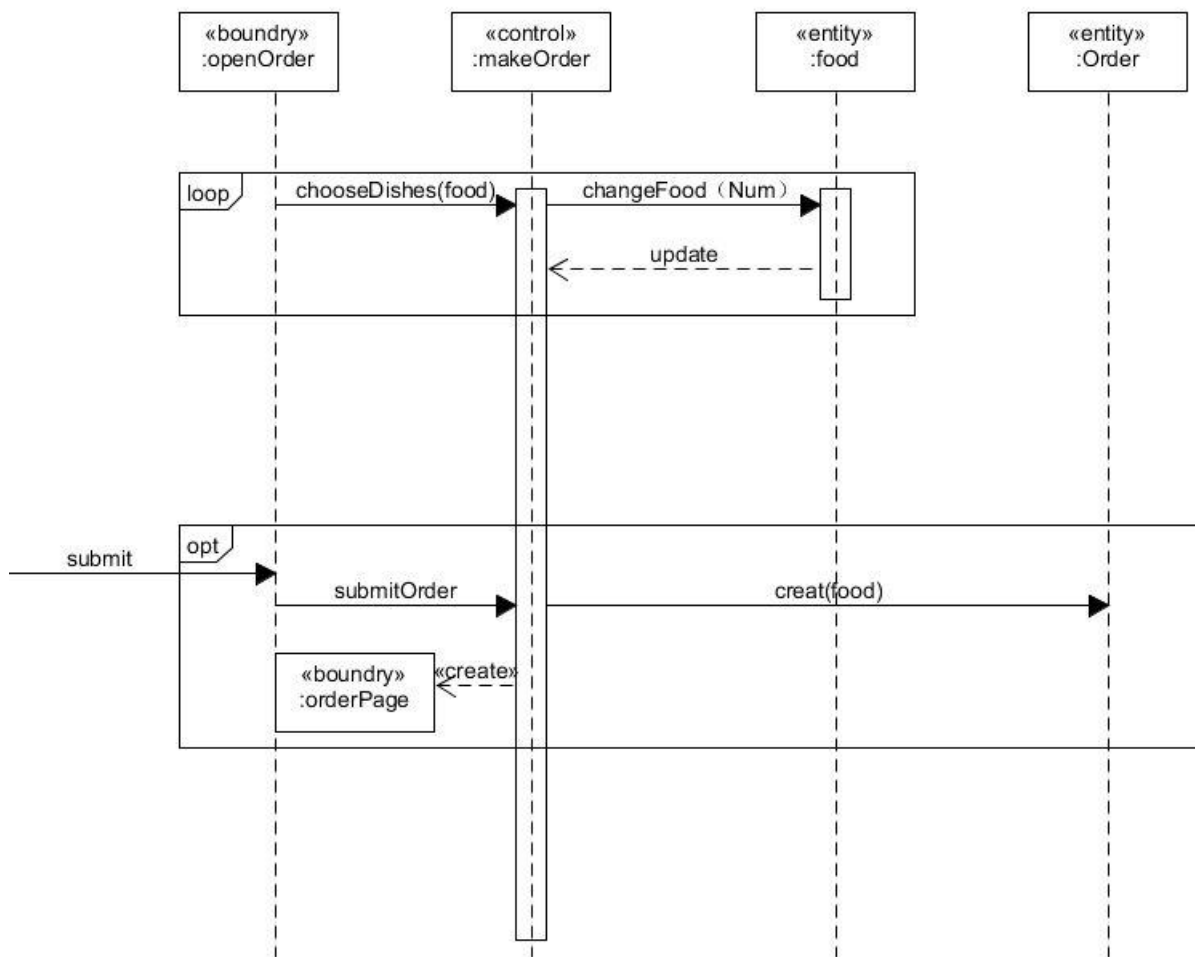
ID	UC2-扫码
Description	顾客扫描餐桌上的二维码，标识自己的身份
Actors	顾客
Preconditions	顾客携带手机，且手机支持二维码扫描。
Alternate Steps	<p>如果顾客重复扫描同一个二维码，系统将记录并显示与该二维码餐桌对应的订单信息和购物车状态。</p> <p>如果顾客在订单完成前重复扫描不同的二维码，系统将提示顾客是否放弃之前的点餐，更换新的餐桌位置。</p> <p>如果多个顾客扫描同一个二维码，系统将通过不同的订单编号来标识对应顾客。</p>
Special Requirements	<ul style="list-style-type: none"> • 用户有网络或流量可使用 • 90%的授权响应时间小于 10 秒。 • 希望在访问远程服务失败的情况下具有一定的恢复功能。
Pending Issues	<ul style="list-style-type: none"> • 研究远程服务的恢复问题。 • 针对不同的业务需求怎样进行定制？

UC design

查找菜品顺序图

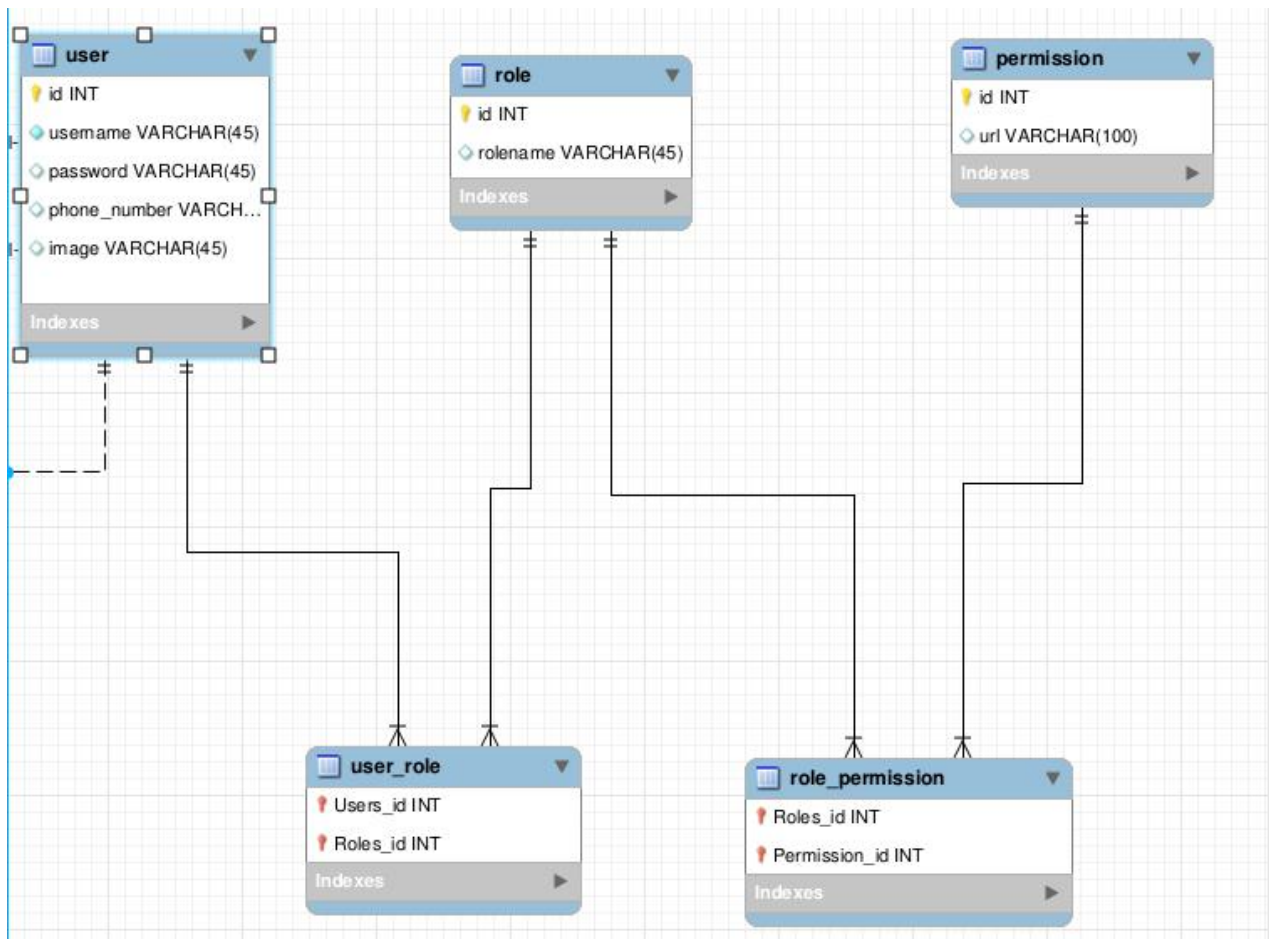


点餐顺序图

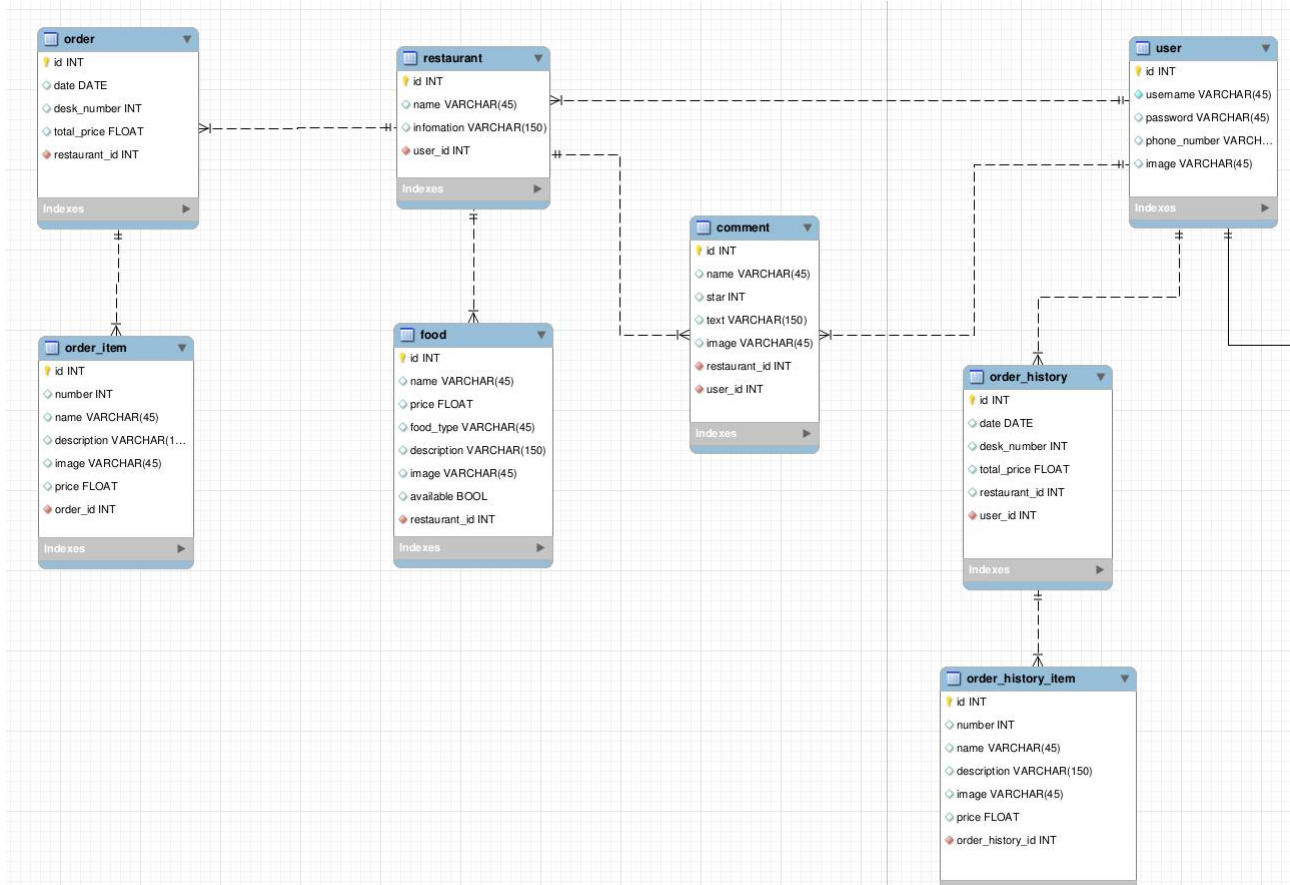


3.3 Entity Relationship Diagrams

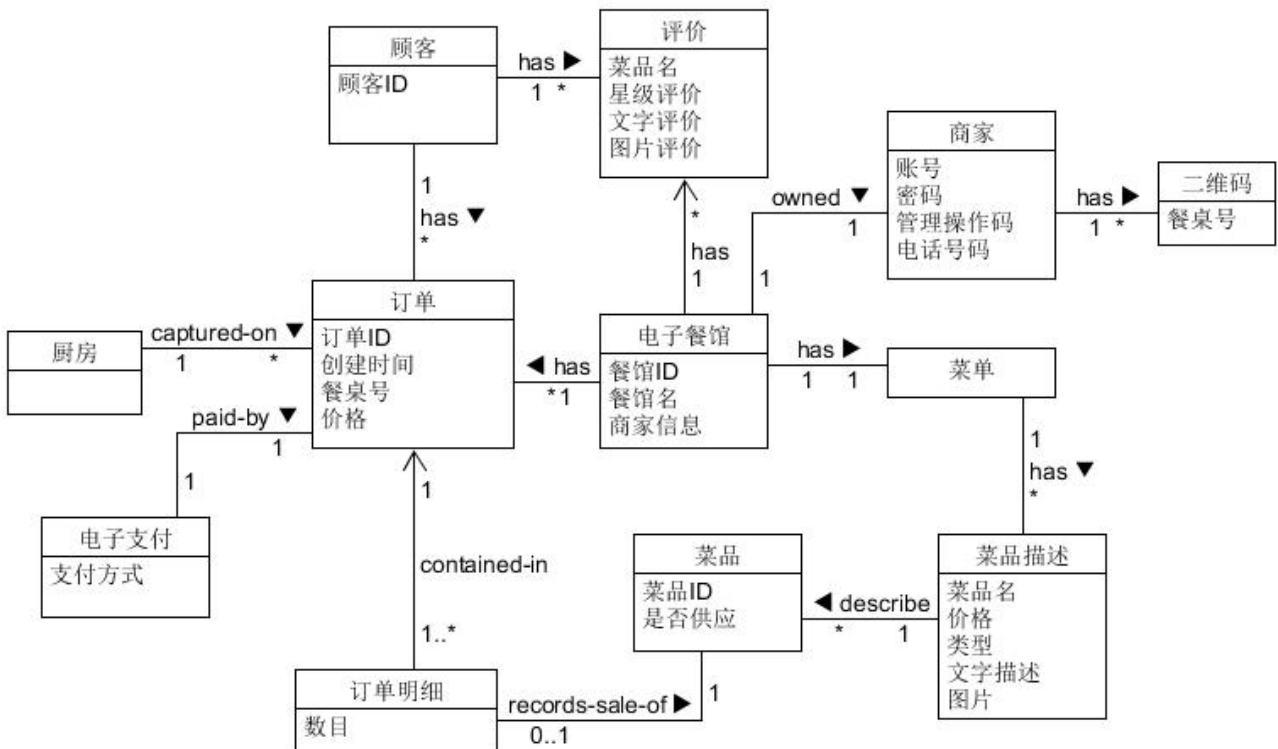
权限系统数据库设计 ER 图



点餐系统数据库设计 ER 图



DM 领域模型



3.4 Data Dictionary

3.4.1 Entity 1

Entity Name: Food

Attribute	Type	Optional?	Notes
<Attribute Name>	<Data type of the attribute>	<Y or N>	<Explain any specific restrictions or rules applcable on this attribute>
id	INT	N	Non-repeatable
name	VARCHAR	N	VARCHAR(45)
price	FLOAT	N	
Food_type	VARCHAR	Y	VARCHAR(45)
description	VARCHAR	Y	VARCHAR(150)
image	VARCHAR	Y	VARCHAR(45)
available	BOOL	N	
restaurant_id	INT	N	Non-repeatable

3.4.2 Entity 2

Entity Name: order

Attribute	Type	Optional?	Notes
<Attribute Name>	<Data type of the attribute>	<Y or N>	<Explain any specific restrictions or rules applcable on this attribute>
id	INT	N	Non-repeatable
date	DATE	N	
total_price	FLOAT	N	
restaurant_id	INT	N	Non-repeatable

4 External Interface Requirements

重要硬件和接口：触摸屏（操作系统将此视为普通监视器，且触摸动作也视为鼠标事件），票据打印机。

软件接口：由于存在众多外部协作系统（税金计算器、财务、... ），我们需要采用不同的接口，接入不同的系统。

5 Technical Requirements (Non functional)

5.1 Performance

90%的授权响应时间应小于 10 秒。

5.2 Scalability

不同食客与商家所采用的使用设备不同，所以在开发的过程中需要对此使用不同的 UI 设计和支持。

5.3 Security

因为用户支付过程中可能存在着账号的风险，所以在支付接口对接的部分，还需要尽量考虑操作的安全性设计。不同角色对商家数据进行操作需要不同的权限，高级权限需要经过用户认证。

5.4 Maintainability

如可用性中所说的，需要在食客的本地设备中存储操作状态，从而减小对错误操作的恢复代价。

5.5 Usability

由于人性因素，食客可能因为不清楚扫码规则或者自身变动而进行第二次错误的扫码操作，因此对上一次操作的状态存储是有必要的。厨师的视线通常专注于菜品，因此提示应通过声音的传递而不仅仅是通过图像传递。

5.6 Multilingual Support

EOEO 系统的多语言支持范围暂定为中文。

5.7 Availability

我们建议使用一些开源构件，但是要解决其许可限制的问题，以便使包含开源软件的产品能够转售。法律规定，在销售交易中必须遵从所有的税务规则。同时要注意的是，这些规则可以频繁变更。

6 Open Issues

暂时有以下未决问题：研究远程服务的恢复问题？针对不同的业务需求怎样进行定制？顾客在支付订单后，是否可以呼叫服务，更改订单信息？