***Система за поръчки за вкъщи в ресторант***

***1. Бизнес нужди и свойства на системата***

***2. Потребителски случаи,Use Case model & Activity диаграми***

***Автори:  
Величко Вълков, ф.н. 71347***

***„Информационни Системи“, IV курс***

***Мариела Николова, ф.н.71350***

***„Информационни Системи“,IV курс***

***Софийски Университет „Св. Климент Охридски“   
Факултет по Математика и Информатика***

***1.Бизнес нужди и свойства на системата***

**(Business Needs and System Features)**

Интернет епохата оказва толкова голямо влияние на днешните хора, че те използват интернет за поръчване на продукти, ползване на различни услуги, дори и поръчване на храна.

Повечето от днешните собственици на ресторанти се стремят да подобрят процеса на работата си и да имат конкурентно предимство пред другите. Решение на техния проблем е нашата система за поръчване на храна по интернет, която автоматизира и оптимизира целия процес на поръчване на храна за вкъщи. Тази храна може да бъде доставена по домовете или взета на място от ресторанта от лице, което я е поръчало За да се поддържа обратна връзка с клиентите, е възможно споделянето на мнения(както позитивни, така и негативни) относно храната, която са закупили. Мениджърът на ресторанта може да променя менюто на ресторанта, което се показва на клиентите, и да преглежда мненията за храната, които са му предадени от сервитьорите и касиерите (например под формата на книга за мнения). Преди появата на такава система, функциите й се осъществяват от сервитьорите и касиерите в ресторанта. Това може да бъде затрудняващо за тях с нарастването на веригата, откъдето се поражда вариантът този вид поръчки да бъдат автоматизирани чрез нашата информационна система.

Нашата система улеснява осъществяването на поръчки за вкъщи във всяко звено,в което е инсталирана, на веригата от ресторанти. Чрез онлайн каталог (меню) клиентът може да разглежда възможните покупки на храна, да избере каква храна иска да закупи и да я поръча чрез системата. Освен това ще се доставя възможност да се платят по банков път и двата вида поръчки на храна.

Системата ще поддържа влизане в нея на различни потребители (както работещи лица, като касиера или мениджъра, така и на регистрирали се чрез системата клиенти). Нерегистрираният потребител по подразбиране е клиент. За всеки тип потребител системата ще поддържа различни нива на достъп, тоест правата на клиента ще са различни от тези на сервитьора, които пък от своя страна ще са различни от тези на мениджъра. Основните възможности на всеки от тях ще са :

* **Нерегистриран потребител** (достъп до системата по подразбиране) – може да поръчва и разглежда храна от ресторанта
* **Регистриран потребител** – **клиент** – може да поръчва и разглежда храна от ресторанта, както и да запаметява лични данни като адрес и любими храни. Този вид потребител може да подава мнения за храната в ресторанта чрез системата.
* **Сервитьор** – може да създава поръчки и да ги потвърждава като поръчки в процес на приготвяне , поръчки в процес на доставяне и приключили поръчки. Също така може да избира мнения, които да даде под формата на доклад на своя мениджър чрез системата.
* **Доставчик** – може да взема поръчки , които са приготвени и за доставяне по домовете, и да потвърждава, че са приключили
* **Готвач** – може да потвърждава приготвени поръчки
* **Мениджър** – може да променя менюто на ресторанта и да чете доклади с мнения за храната, които са дадени от сервитьора чрез системата.

Ще изброим основните бизнес предимства на нашата система.

**Ползите за клиентите са:**

• Широк достъп на клиентите: Потребителите могат да правят поръчки дори когато ресторантът е затворил. Това означава, че могат да поръчват 24х7х365.

• Специални поръчки от менюто: Преди когато клиентите искаха да направят поръчка те трябваше или да позвънят в ресторанта или лично да го посетят, за да видят специалитетите на деня. Сега те могат да го правят онлайн.

• Лесно управление на менюто: Системата организира ястията от менюто много систематично, така че потребителите могат да го разглеждат лесно без трудности. Също така могат лесно да се правят промени по менюто. Категоризираните менюта винаги привличат повече внимание.

• Възможността за объркване на поръчката е елиминирана

• Пести време

**Ползите за собствениците са :**

• Конкурентно предимство пред другите

• Поръчки подадени за секунди

• Обновяване на менюто по всяко време

• Подсигурени опции за плащане

• Осигуряване на плащане чрез карта

• Лесно могат да бъдат пуснати разлини оферти и промоции онлайн

• Правят се по-малко грешки по време на процеса за поръчване

***2. Подробно описание на потребителските случаи:***

**Use-Case: View Food Details**

**Brief Description**

*Клиентът* преглежда каталога с различни *Ястия*, които предлага ресторанта. Може да разгледа различните *Ястия* поотделно и да види цената им и да поиска описание какво представляват.

**Actor Brief Descriptions**

**Клиент –** лицето, което си поръчва храна от ресторанта

**Инвентарната система –** системата, в която се съхранява информация за наличностите

**Preconditions**

*Клиентът е влязъл в системата*

**Basic Flow of Events**

1. *Клиентът* влиза в*Менюто* с *Ястия* на*Системата.*
2. *Системата* изпраща заявка до *Инвентарната Система* за проверка на наличностите.
3. *Инвентарната Система* връща информация за наличностите.
4. *Системата* показва *Менюто* с наличните предлагани *Ястия*.
5. *Клиентът* разглежда *Менюто*.
6. *Клиентът* иска да види детайлна информация за определено избрано *Ястие*.
7. *Системата* показва детайлна информация за това *Ястие* (начин на приготвяне , продукти, снимки).

Потребителският случай е завършил успешно.

**Alternative Flows**

**1.Няма наличности в инвентарната система**

**Ако на стъпка 3** няма наличности в инвентарната система, тогава:

1. *Системата* показва на *Клиента* съобщение, че няма налични *Ястия*

(Потребителският случай завършва неуспешно)

**Post-conditions**

**Successful Completion**

*Клиентът* е разгледал *Менюто* успешно

**Failure Condition**

*Клиентът* не е разгледал *Менюто*.

**Special Requirements**

**Use-Case: Order Food [Logged in]**

**Brief Description**

*Клиентът* поръчва определени ястия за вкъщи и *Системата* регистрира *Поръчката*, след като е проверила дали ястието може да се приготви. Предоставя се избор на *Клиента* как да бъде осъществена *Поръчката*.

**Actor Brief Descriptions**

**Клиент –** лицето, което си поръчва храна от ресторанта

**Готвач –** лице от кухнята на ресторанта

**Preconditions**

*Клиентът има създаден потребителски акаунт в системата и е влязъл в него.*

**Basic Flow of Events**

1. *Системата* предлага възможност за избор на опция
2. *Клиентът* избира опцията да си поръча *Храна*
3. *Системата* предлага възможност на клиента за избор кои *Храни* да поръча.
4. *Клиентът* разглежда *Ястията* използвайки потребителски случай “View Food Details” и избира *Ястия*.
5. *Системата* проверява чрез заявка до *Инвентарната Система* дали избраните *Ястия* за налични.
6. *Системата* предлага на клиента опции за плащане.
7. *Клиентът* избира начин на плащане с дебитна/кредитна карта съгласно потребителски случай “Debit Card Payment” и потвърждава поръчката
8. *Системата* пита *Клиента* дали иска да поръча доставка за вкъщи, или ще вземе храната от *Ресторанта.*
9. *Клиентът* избира да вземе храната от *Ресторанта*.
10. *Системата* показва информация за направената *Поръчка* ( показва избраните *Ястия*, отделните им цени, както и общата дължима сума, включително и таксата за доставка.
11. *Системата* изпраща информация за направената *Поръчка* до *Инвентарната Система* за обновяване на наличностите .
12. *Системата* изпраща информация за направена *Поръчка* до *Готвача*
13. Потребителският случай е завършил успешно. (Продължава в потребителски случаи „Order Food for Pickup” )

**Alternative Flows**

**1.Неналично ястие**

Ако на стъпка 5 от основния сценарий Ястието, което се иска от Клиента, е неналично, се правят следните стъпки:

1. *Системата* се извинява на клиента и му съобщава,че ястието е неналично

(Продължава се със стъпка 3 от основния сценарий)

**2.Плащане по ePay**

Ако на стъпка 7 от основния сценарий Клиентът избере да плати поръчаната храна по ePay, ce правят следните стъпки:

1. *Клиентът* избира начин на плащане с ePay съгласно потребителски случай “ePay Payment” и потвърждава поръчката

*(Продължава се със стъпка 8 от основния сценарий)*

**3.Плащане на ръка**

Ако на стъпка 7 от основния сценарий *Клиентът* избере да плати поръчаната храна на ръка ce правят следните стъпки:

1. *Клиентът* избира начин на плащане на ръка и потвърждава *Поръчката*

*(Продължава със стъпка 8 от основния сценарий)*

**4.Валидиране на Ваучър**

Ако на стъпка 7 от основния сценарий *Клиентът* избере да плати поръчаната храна по начин по свой избор и избере опция да използва *Ваучър,* се правят следните стъпки:

1. *Клиентът* избира начин на плащане , избира опция *„Валидиране на Ваучър“* и потвърждава *Поръчката*

*(Преминава се към потребителски случай „Register Voucher”)*

**5.Доставка за вкъщи**

Ако на стъпка 9 от основния сценарий Клиентът поръча доставка за вкъщи, се правят следните стъпки:

1. *Клиентът* избира да поръча доставка за вкъщи.
2. *Системата* показва информация за направената *Поръчка* ( показва избраните *Ястия*, отделните им цени, както и общата дължима сума, включително и таксата за доставка.
3. Системата изпраща информация за направената *Поръчка* до *Инвентарната Система* за обновяване на наличностите .
4. Системата изпраща информация за направена *Поръчка* до *Готвача*
5. Потребителският случай е завършил успешно. (Продължава в потребителски случаи „Order Food for Delivery” )

**Post-conditions**

**Successful Completion**

*Клиентът* успешно е направил *Поръчка* за храна за вкъщи, която ще му бъде доставена или ще вземе от *Ресторанта*.

**Failure Condition**

*Клиентът* не е направил *Поръчка* за вкъщи.

**Special Requirements**

[SpReq:OF-1] *Клиентът* не може да поръча повече от 10 *Ястия* с една поръчка

[SpReq:OF-2] *Клиентът* не може да поръча две *Поръчки* едновременно.

**Use-Case: OrderFoodforDelivery**

**Brief Description**

Потребителският случай разширява потребителския случай“OrderFood”, като се разглежда вариантът, в който *Клиентът* иска да му бъде доставена *Поръчката* вкъщи.

**Actor Brief Descriptions**

**Клиент –** лицето, което си поръчва храна от ресторанта

**Инвентарната система –** системата, в която се съхранява информация

**Preconditions**

*Клиентът е разгледал менюто и е избрал храна за поръчване чрез потребителски случай “Order Food”, в който се ползва алтернативен сценарий 5.5 (Доставка за вкъщи).*

**Basic Flow of Events**

1. *Системата* предлага на клиента избор да си избере и закупи *Храна* съгласно потребителски случай “OrderFood”.
2. *Системата* иска *Информация* за това кога и къде да бъде доставена *Храната*.
3. *Клиентът* въвежда *информацията*.
4. *Системата* изисква *Телефон* за контакти.
5. *Клиентът* въвежда *Телефон*.
6. *Системата* потвърждава и показва *Информация* за *Поръчката*.

Потребителският случай е завършил успешно.

**Alternative Flows**

**1.Невалидна информация(дата и адрес)**

**Ако на стъпка 3 клиентът** не въведе правилно валидна информация, тогава:

1. *Системата* изисква въвеждането отново на *Информацията*
2. Потребителският случай се връща на **стъпка 2**

**2.Клиентът вече е регистриран**

**Ако на стъпка 4** клиентът вече има регистрация, и е влязъл с нея в системата, тогава:

1. *Системата* обработва вече известната информация
2. Потребителският слуай преминава на **стъпка 6**

**3.Невъведен/Невалиден тел.номер**

**Ако на стъпка 5** клиентът не въведе тел.номер или той е невалиден, тогава:

1. *Системата* съобщава за грешка
2. Потребителският слуай се връща на стъпка 4

**Post-conditions**

**Successful Completion**

*Клиентът* си е поръчал храна за доставка , поръчката е записана в *Системата* и е в процес на изпълнение

**Failure Condition**

*Клиентът* не е успял да си поръча храна за доставка

**Special Requirements**

*Клиентът* не може да направи две *Поръчки* едновременно.

# Use-Case: Register Voucher

# Brief Description

*Клиентът* получава отстъпка към поръчаната от него *Храна*, след като е заявил,че ще ползва свой *Ваучър* за отстъпка.

# Actor Brief Descriptions

## Клиент – лицето, което си поръчва храна от ресторанта

# Preconditions

*Клиентът прави Поръчка в Ресторанта.*

# Basic Flow of Events

1. (Продължава се потребителски случай “Order Food”)
2. *Клиентът* избира опцията за плащане с отстъпка.
3. *Системата* поисква от *Клиента* кода на *Ваучъра* му.
4. *Клиентът* въвежда кода на своя *Ваучър* .
5. *Системата* проверява кода ипоказва списък на *Храните* от поръчката, към които *Клиентът* получава отстъпка и колко е новата цена на *Поръчката*
6. *Клиентът* потвърждава.
7. *Системата* запазва промените.
8. (Преминава се към потребителските случаи за различните видове плащане)

# Alternative Flows

**1. Клиентът е въвел невалиден код на Ваучър**

Ако на стъпка 4. Клиентът въведе невалиден код за *Ваучър*, се правят следните стъпки:

1. *Системата* информира *Клиента*, че кодът му е невалиден и го пита дали иска да пробва отново да го въведе

*(Преминава се към стъпка 4 от основния сценарий)*

**2. Клиентът се отказва от ползването на Ваучър**

Ако на стъпка 6. *Клиентът* не потвърди *Поръчката* си, *Ваучърът* му остава валиден до следващата поръчка, за която ще се използва.

*(Продължава се със стъпка 8 от основния сценарий)*

# Post-conditions

## Successful Completion

*Клиентът* успешно е валидирал своя *Ваучър* и ще получи отстъпка при *Поръчката* си.

## Failure Condition

*Клиентът* не е валидирал своя *Ваучъ*р.

# Special Requirements

[SpReq:WC-1] *Клиентът* не може да ползва два *Ваучъра* за една и съща поръчка.

# Use-Case: Debit Card Payment

# Brief Description

*Клиентът* заплаща за храна , която поръчва, чрез банкова карта.

# Actor Brief Descriptions

## 1.Клиент – лицето, което си поръчва храна от ресторанта

## 2.Банкова система – системата, чрез която се осъществява плащането

# Preconditions

*Клиентът* не е влязъл със своята регистрация в системата*,* разгледал е менюто и си е избрал храна за поръчка. Клиентът си поръчва храна чрез Системата.Системата предлага възможност за избор на тип на плащането. (Стъпка 6 от потребителски случай “OrderFood”)

# Basic Flow of Events

1. *Системата* предлага възможност за избор на тип на плащането.
2. *Клиентът* избира да плати чрез *Дебитна* карта
3. *Системата* иска от *Клиента* информация за *Дебитната* карта( номер, дата на валидност, код за сигурност).
4. *Клиентът* въвежда необходимата информация.
5. *Системата* се обръща към Банковата Система за проверка на *Дебитната* карта
6. *Системата* показва информация за това колко пари ще бъдат изтеглени от *Дебитната* карта на *Клиента*.
7. *Клиентът* потвърждава *Плащането*.
8. *Системата* обработва заявката и съобщава на *Клиента*, че *Плащането* е било успешно

Потребителският случай е завършил успешно.

# Alternative Flows

## 1.Плащане по ePay

**Ако на стъпка 2** клиентът избере да плати по ePay, тогава:

1. Потребителският случай преминава към use case “ePay Payment”

(Потребителският случай е завършил неуспешно)

## 2.Плащане при вземане на поръчката

**Ако на стъпка 2** клиентът избере да плати когато си вземе поръчката, тогава:

1. Потребителският случай преминава към use case “Pay On PickUp”

(Потребителският случай е завършил неуспешно)

## 3.Плащане при получаване на поръчката

**Ако на стъпка 2** клиентът избере да плати когато поръчката му бъде доставена, тогава:

1. Потребителският случай преминава към use case “Pay On Delivery”

(Потребителският случай е завършил неуспешно)

## 4.Проблеми с плащането

**Ако на стъпка 6** номерът на картата не е валиден/правилен, тогава:

1. Системата изисква повторно въвеждане на номера
2. Потребителският случай се връща на **стъпка 5**

**Ако на стъпка 6** клиентът няма достатъчно средства за заплащане на поръчката, тогава:

1. Системата съобщава на клиента, че има проблем с плащането
2. Системата съобщава на клиента, че поръчката не може да бъде осъществена

(Потребителският случай е завършил неуспшно)

## 5.Отказване на поръчката

**Ако на стъпка 8** клиентът не потвърди плащането , тогава:

1. Системата съобщава на клиента, че плащането не е било успешно

(Потребителският случай е завършил неуспешно)

# Post-conditions

## Successful Completion

## *Поръчката* е платена. Сметката в картата на *Клиента* е обновена.. Плащането е отчетено в *Системата* и в *Банковата система*. Статусът на поръчката е обновен в *Системата*.

## Failure Condition

*Клиентът* не е платил за Поръчката

# Special Requirements

[SpReq:DCP-1]*Клиентът* не може да въвежда повече от три пъти грешен номер на карта

**Use-Case: Provide Feedback**

**Brief Description**

*Клиентът* дава мнение за храна в ресторанта чрез системата.

**Actor Brief Descriptions**

**Клиент –** лицето, което си поръчва *Храна* от *Ресторанта*

**Preconditions**

*Клиентът* е влязъл в потребителското си име.

**Basic Flow of Events**

1. *Системата* предлага възможност за избор на опция.
2. *Клиентът* избира да въведе мнение
3. *Системата* предоставя възможност да се избере секция за оплакване или хвалба.
4. *Клиентът* избира секция за хвалба.
5. *Системата* показва *Менюто*, от което *Клиентът* да избере *Ястие*
6. *Клиентът* избира *Ястие.*
7. *Клиентът* въвежда и изпраща *Хвалба*.
8. *Системата* потвърждава и обработва заявката.

Потребителският случай е завършил успешно.

**Alternative Flows**

**1.Подаване на оплакване**

Ако на стъпка 4. *Клиентът* избере да подаде *Оплакване*, се правят следните стъпки:

1. *Системата* показва *Менюто*, от което *Клиентът* да избере *Ястие*
2. *Клиентът* избира *Ястие.*
3. *Клиентът* въвежда и изпраща *Оплакване.*
4. *Системата* потвърждава и обработва заявката.

Потребителският случай е завършил успешно.

**Post-conditions**

**Successful Completion**

*Клиентът* е подал *Оплакване* или *Хвалба* за *Ястие* от *Ресторанта*.

**Failure Condition**

*Клиентът* не е подал успешно своето *Оплакване* или *Хвалба*.

**Special Requirements**

[SpReq:PF-1] Едно и също потребителско име не може да подава едновременно *Хвалба* и *Оплакване* за едно и също *Ястие*.

# Use-Case: Update Menu

# Brief Description

*Мениджърът* поглежда текущото меню и го актуализира, като внася промени в него.

# Actor Brief Descriptions

## Мениджър – грижи се за актуализирането на менюто в системата

## Инвентарна система – системата, в която се съхранява информация за наличностите

# Preconditions

*Мениджърът се е идентифицирал в Системата*

# Basic Flow of Events

1. *Мениджърът* изисква от системата да види актуалното *Меню*.
2. *Системата* показва *Менюто* и му предоставя възможност за избор на опция.
3. *Мениджърът* избира опция за добавяне на *Ястие* в *Менюто*.
4. *Системата* показва форма, в която *Мениджърът* да въведе информация.
5. *Мениджърът* въвежда информацията и потвърждава новото *Меню*.
6. *Системата* потвърждава и актуализира *Менюто*.

Потребителският случай е завършил успешно.

# Alternative Flows

## 1.Премахване на ястие от менюто

**Ако на стъпка 3** *мениджърът* избере да премахне ястие от менюто, тогава:

1. *Системата* показва на *Мениджъра* *Менюто*

2.*Мениджърът* маркира желаното/ите *Ястие*/я за премахване от *Менюто*

3.*Системата* изисква потвърждение

4.*Мениджърът* потвърждава промените

5.*Системата* актуализира *Менюто*

Потребителският случай е завършил успешно

## 2.Актуализиране на ястие от менюто

**Ако на стъпка 3** *Мениджърът* избере да актуализира*Ястие* от *Менюто*, тогава:

1. *Системата* показва форма, в която *Мениджърът* да промени информацията.
2. *Мениджърът* въвежда информацията и потвърждава новото *Меню*.
3. *Системата* потвърждава и актуализира *Менюто*.

Потребителският случай е завършил успешно.

# Post-conditions

## Successful Completion

## *Менюто е актуализирано.*

## Failure Condition

*Мениджърът не успял да актуализира менюто.*

# Special Requirements

**Use-Case: Deliver Orders**

**Brief Description**

*Системата* дава задача на *Доставчика* за доставка по домовете. *Доставчикът* я изпълнява и потвърждава.

**Actor Brief Descriptions**

**Доставчик** – лицето, доставящо *Поръчката* до *Клиента*

**Preconditions**

*Клиентът* е направил поръчка през системата и е избрал метод на плащане „Плащане при доставка“.

**Basic Flow of Events**

1. *Системата* информира *Доставчика* за поръчката
2. *Дoставчикът* потвърждава поръчката и я маркира като „*Доставяща се“*
3. *Системата* предлага оптимален маршрут и график за извършване на доставката
4. *Доставчикът* избира опция за влизане в меню за плащане в брой
5. *Системата* регистрира поръчката и я осчетоводява, издавайки касова бележка.
6. *Доставчикът* потвърждава плащането като успешно и маркира поръчката като „*Приключила*“.

Потребителският случай е завършил успешно.

**Alternative Flows**

**1.Отменяне на поръчката**

Ако след стъпка 2. *Клиентът* се обади по телефона и информира *Ресторанта*, че иска да отмени поръчката си, се изпълняват следните стъпки:

1. *Доставчикът* маркира поръчката като *„Неуспешна“.*

Потребителският случай е завършил неуспешно.

**2.Несъществуващ адрес**

Ако след стъпка 2. *Доставчикът* не открие зададеният му от *Клиента* адрес, се изпълняват следните стъпки:

1. *Доставчикът* маркира поръчката като *„Неуспешна“.*

Потребителският случай е завършил неуспешно.

**Post-conditions**

**Successful Completion**

*Клиентът* е получил поръчката си, платил е и *Доставчикът* е потвърдил в системата.

**Failure Condition**

*Поръчката* не е доставена и заплатена.

**Special Requirements**

[SpReq:DO-1] Поръчката може да се отмени до 15 минути след нейното поръчване. След това тя бива изпълнявана задължително.

**Use-Case: Manage Orders**

# Brief Description

# *Сервитьорът* управлява осъществяващите се *Поръчки* чрез *Системата*.

# Actor Brief Descriptions

## Сервитьор– Обслужващо лице в *Ресторанта*

# Preconditions

*Сервитьорът* се е идентифицирал в *Системата*

# Basic Flow of Events

1. *Системата* предоставя опции на *Сервитьора*
2. *Сервитьорът* избира, че ще управлява текущи поръчки.
3. *Системата* предоставя списък на направените засега *Поръчки*.
4. *Сервитьорът* избира *Поръчка*
5. Системата показва текущото състояние на *Поръчката* и информация за нея (поръчана *Храна*, дължима сума)
6. *Сервитьорът* променя състоянието на поръчката на *Успешно*
7. *Системата* потвърждава и записва промяната.

Потребителският случай е завършил успешно.

# Alternative Flows

**1.Избрана е друга опция, тогава:**

Потребителският случай завършва неуспешно.

## 2.Сервитьорът отменя поръчка

**Ако на стъпка 6** *Сервитьорът* избере да отмени дадена *Поръчка*, тогава:

1. *Сервитьорът* променя състоянието на поръчката на *Неуспешна.*
2. *Системата* потвърждава и записва промяната.

Потребителският случай е завършил успешно.

# Post-conditions

## Successful Completion

## *Променено* е състоянието на *Поръчка*.

## Failure Condition

*Сервитьорът* не е могъл да промени състоянието на определена *Поръчка.*

# Special Requirements

*Няма.*

**Use-Case: Prepare Food**

# Brief Description

*Системата* изпраща заявка до *Готвача* за приготвяне на *Храна*. *Готвачът* приготвя *Храната*.

# Actor Brief Descriptions

## Готвач – лицето, приготвящо *Храна* в *Ресторанта*.

# Preconditions

*Готвачът* е влязъл с потребителското си име в *Системата.*

# Basic Flow of Events

1. *Системата* показва на *Готвача* меню с опции
2. *Готвачът* избира опция да прегледа поръчки
3. *Системата* показва на *Готвача* списък с направени *Поръчки* и *Ресторантите*, към които са отправени
4. *Готвачът* избира конкретна *Поръчка*
5. *Системата* показва конкретна информация за *Поръчката* (поръчано *Ястие*) и опции
6. *Готвачът* променя състоянието на поръчката като *„Приготвяща се“*
7. *Готвачът* променя състоянието на поръчката като *„Приготвена“*
8. *Системата* запазва промените*.*

Потребителският случай е завършил успешно.

# Alternative Flows

## 1.Няма текущи поръчки

Ако след стъпка 2. *се окаже, че няма текущи Поръчки към Ресторанта, се правят следните стъпки:*

1. *Системата* информира *Готвача,* че в момента няма налични *Поръчки.*

Потребителският случай е завършил неуспешно.

# Post-conditions

## Successful Completion

*Поръчка* е обработена от системата и приготвена от *Готвача*, за което той е потвърдил.

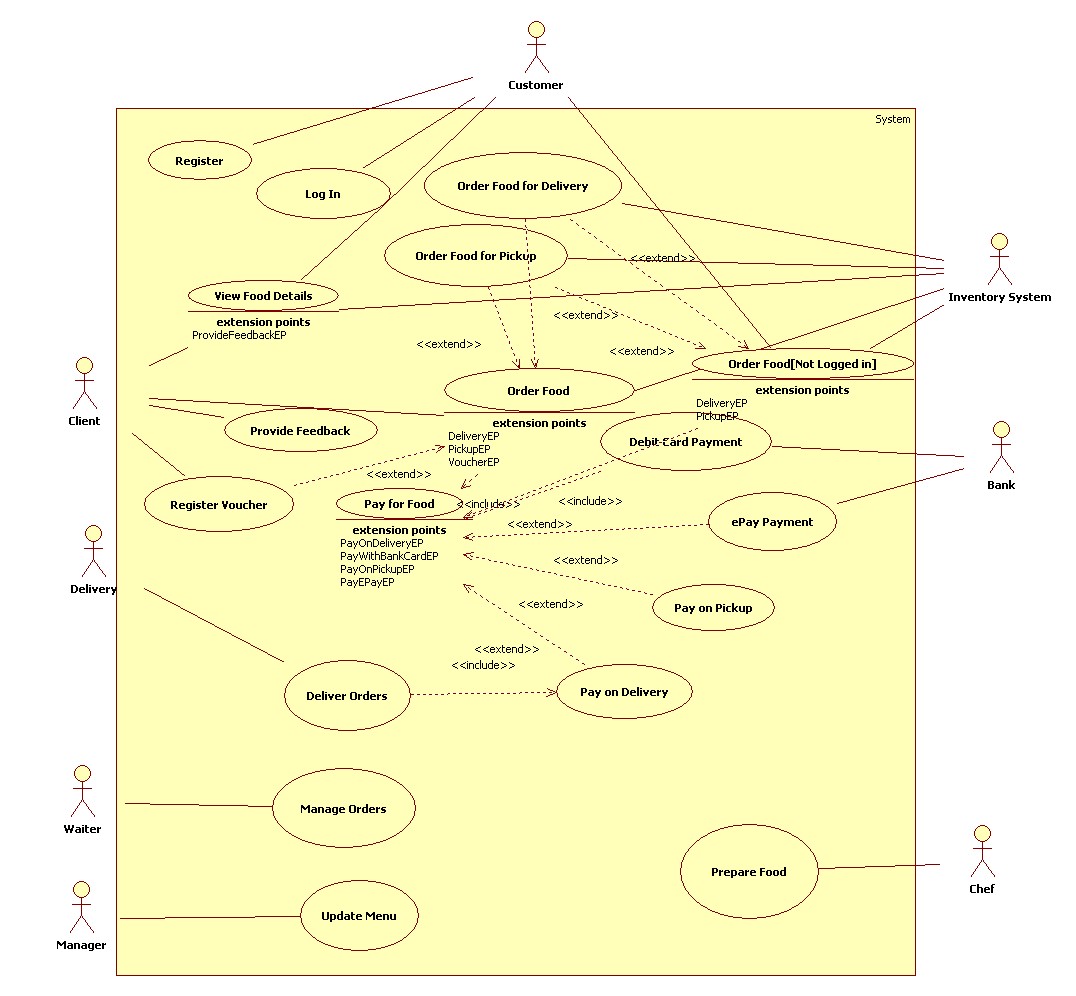
## Failure Condition

Не е приготвена *Поръчка.*

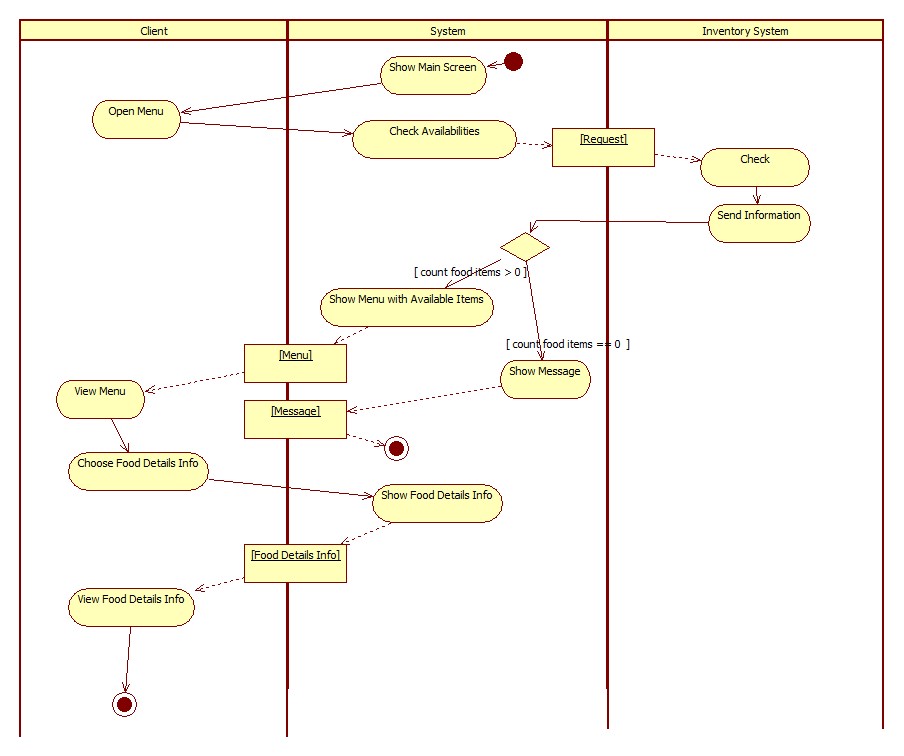
# Special Requirements

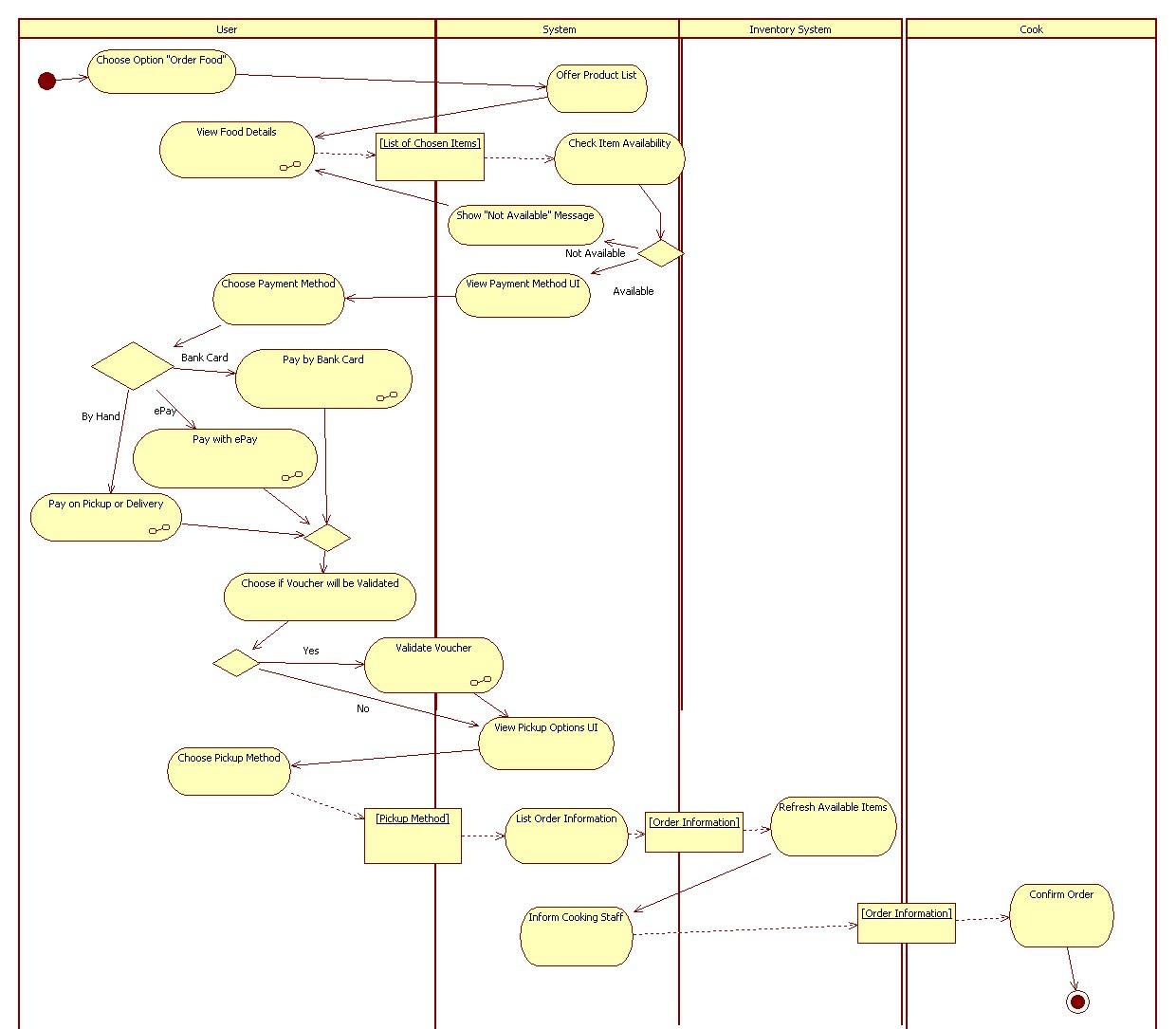
[SpReq:PF-1] Един и същи *Готвач* може да работи по най-много три *Поръчки* едновременно.

***2.Use-Case Model:***

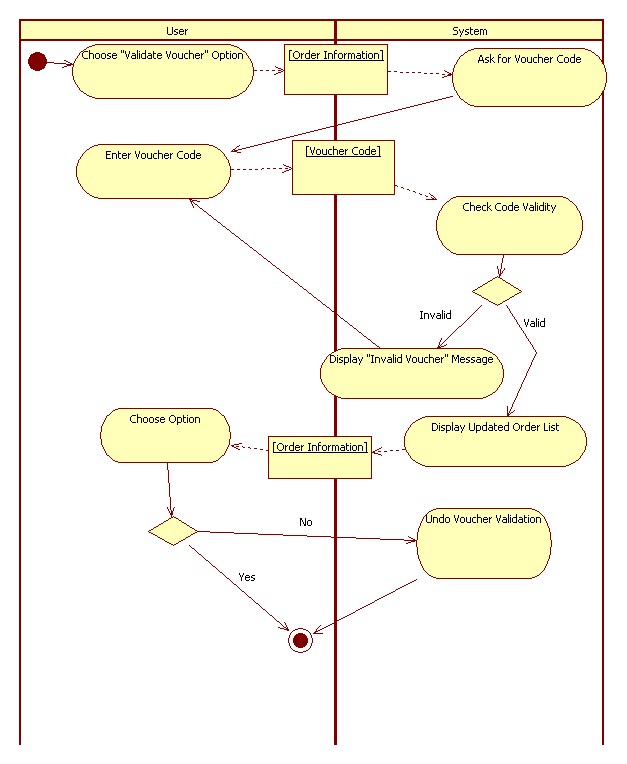


***3.Activity Diagrams***

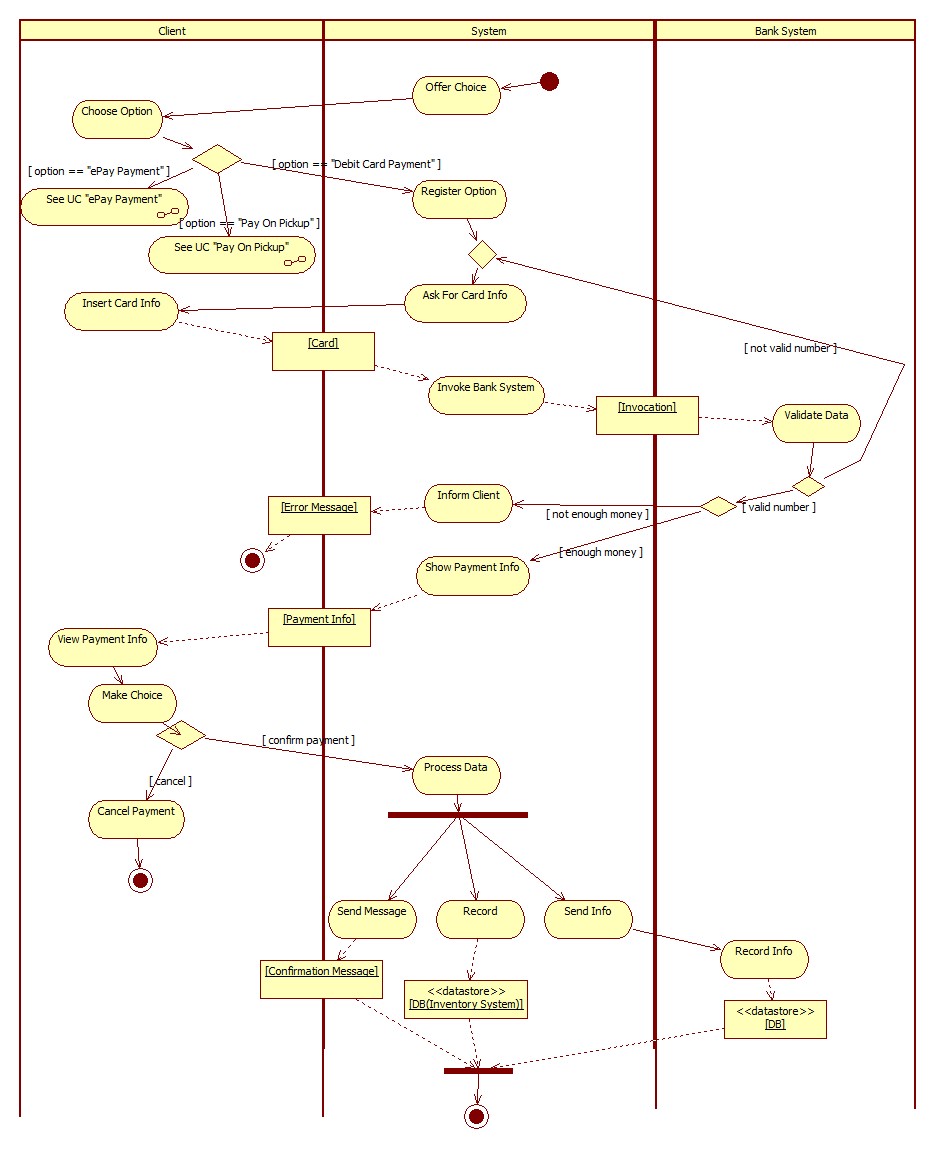
***3.1.View Food Details***

***3.2.Order Food [for Delivery]***

***3.3. Register Voucher***

******

***3.4.Debit Card Payment***

******

***4 .Речник***  
*Храна, Ястие:* Избраната от клиента храна от ресторанта.

*Меню:* Списък с предлагани *Храни* в ресторанта.

*Поръчка:*Опакованата храна за вкъщи, която клиентът купува от ресторанта - има три състояния: *Поръчана, Приготвяща се, Приготвена*

*Сервитьор:*Обслужващо клиенти лице в ресторанта.

*Ресторант:*Мястото, където се приготвя храната.

*Клиент:*Човекът, който иска да си купи храна от ресторанта.

*Доставчик:*Лице, което доставя храната по адреси в града на ресторанта.

*Готвач:* Лицето, което приготвя храната в ресторанта

*Ваучър:* Талон за отстъпка в ресторанта

*Оплакване:* Негативно мнение за храна в ресторанта, което не е систематизирано в доклад

*Хвалба:* Позитивно мнение за храна в ресторанта, което не е систематизирано в доклад.