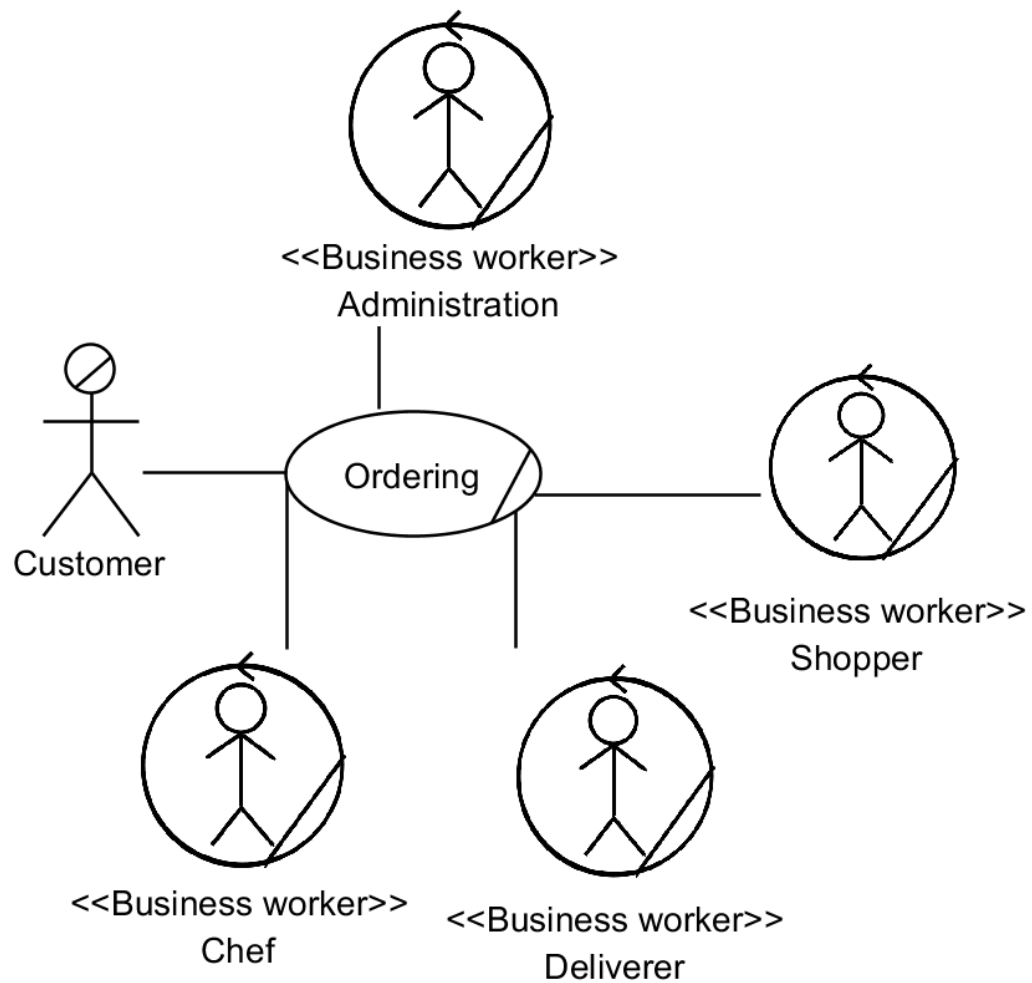
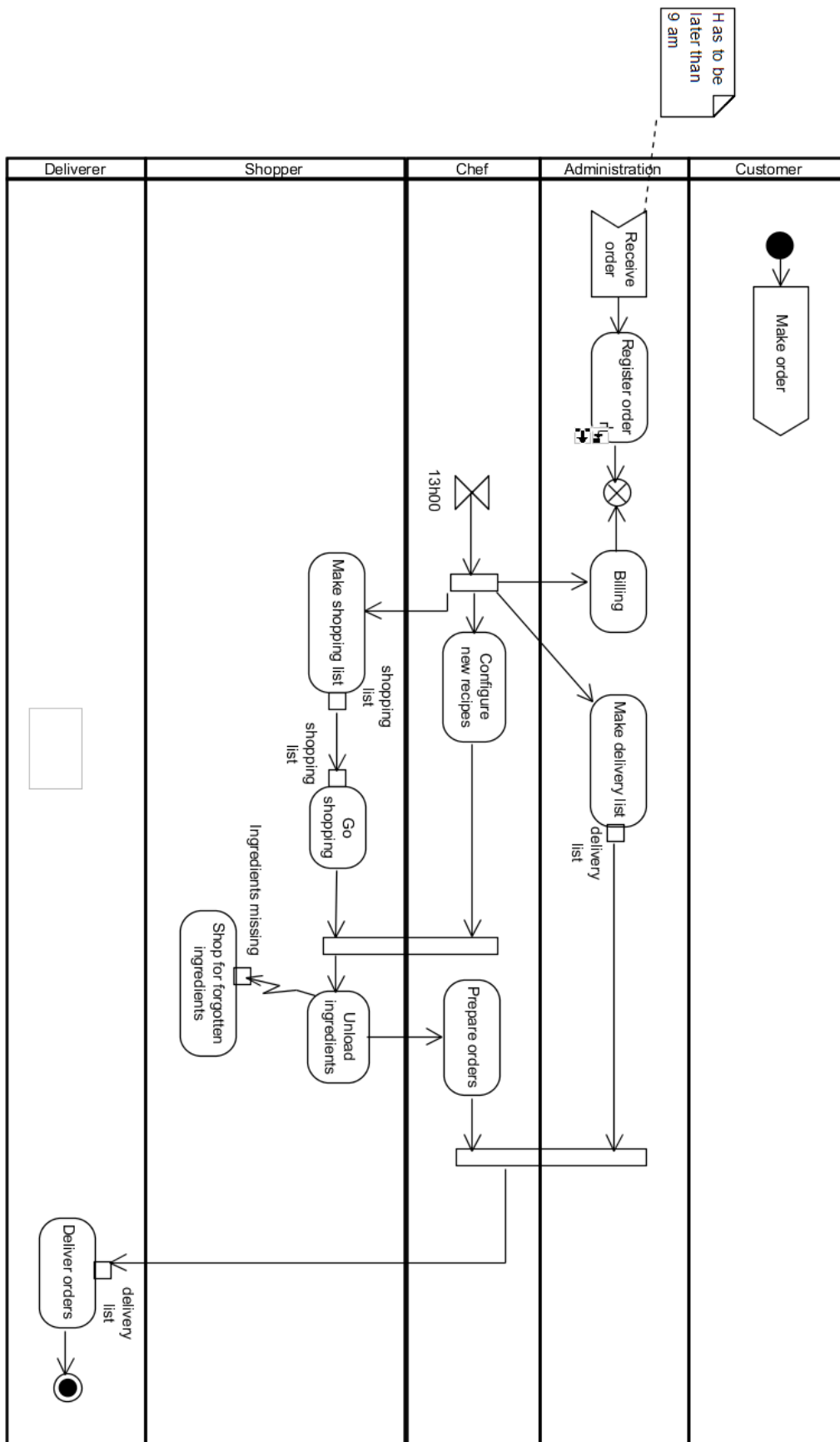


Assignment 1 : Business analysis

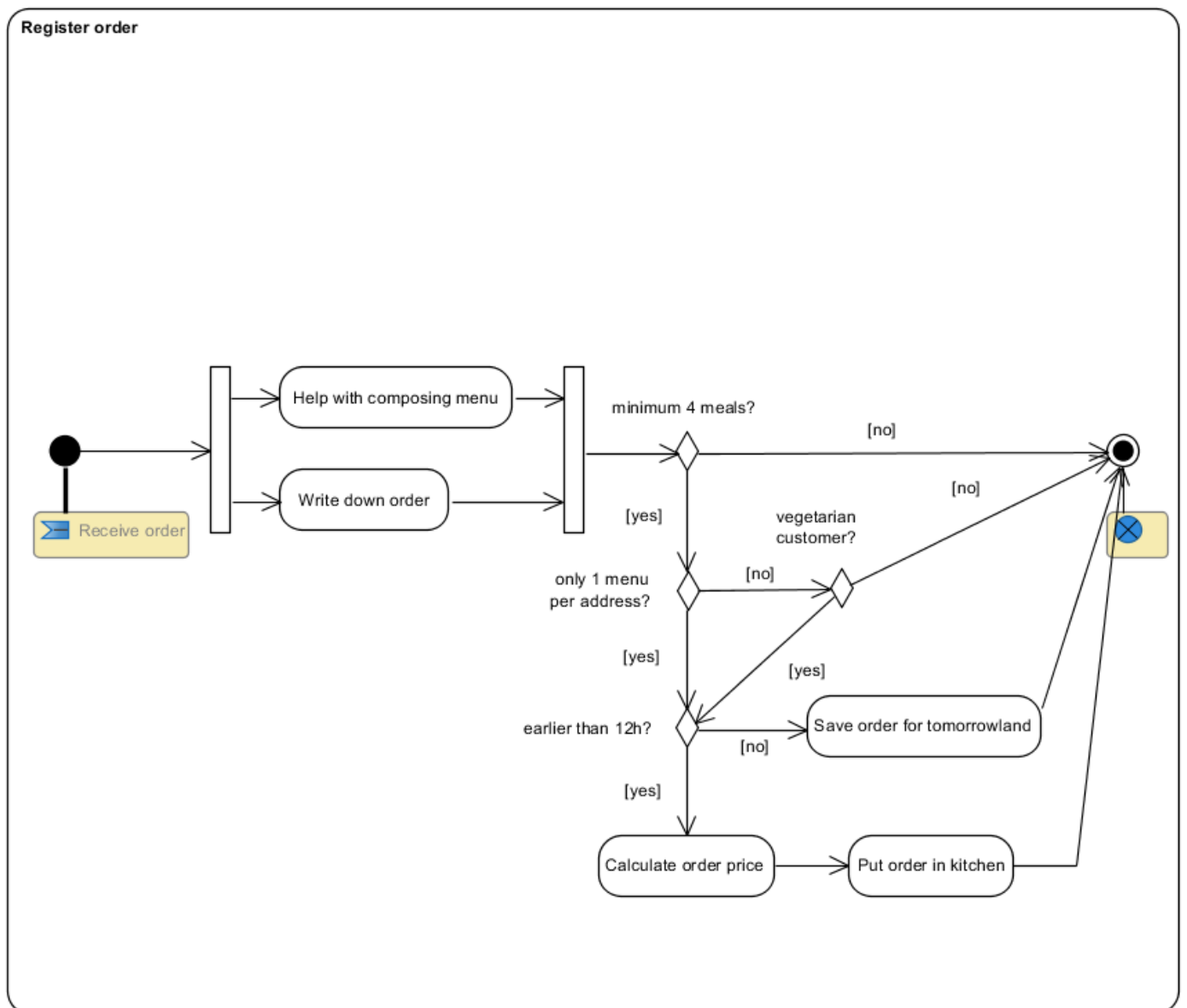
1.1 Use case diagram



1.2 Overall activity diagram

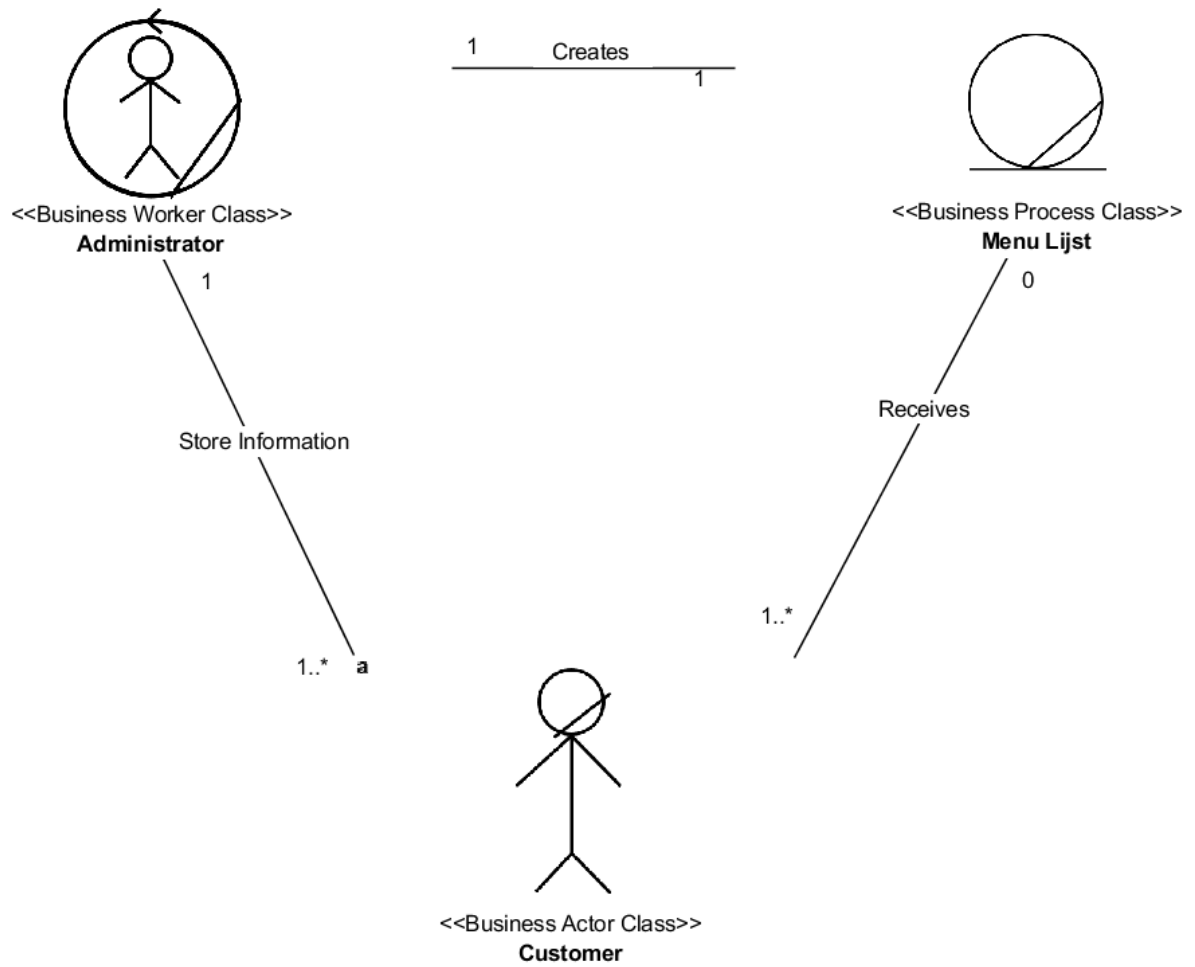


1.3 Detailed activity diagram (*register order*)

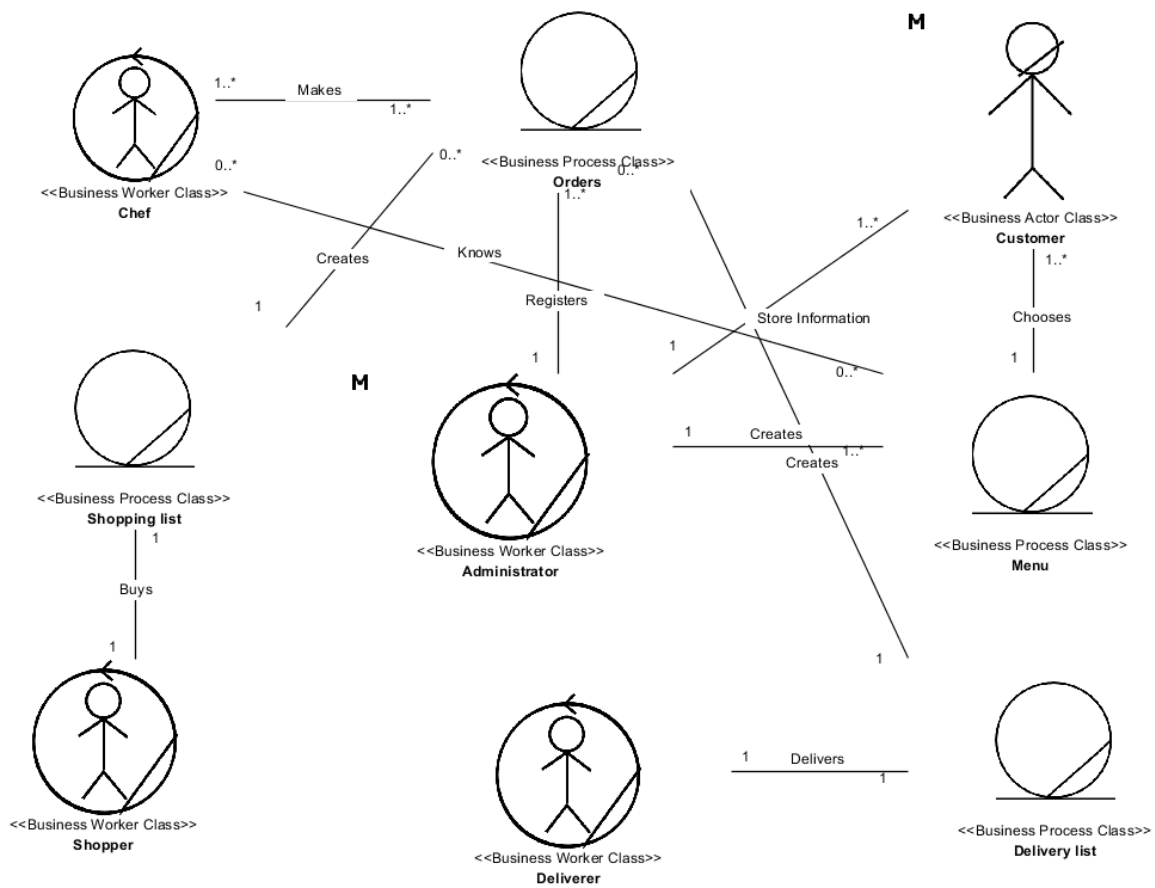


1.4 Business class diagram (*current situation*)

a



1.5 Business class diagram (*wanted situation*)



Assignment 2 : Problem analysis

PROBLEM STATEMENT MATRIX

Brief Statements of Problem, Opportunity, or Directive	Urgency	Visibility	Priority or Rank	Proposed Solution
Because there are 3 different cooks working in the kitchen, 1 dish is sometimes prepared by 3 different recipes. Clients don't like that.	ASAP	High	2	New development
Some clients are ordering twice on the same day so they can get different menus instead of just 1.	2 months	High	2	New development
Recipes suggested by clients aren't well kept up.	3 months	High	2	New development
There is no file with all the information of the client.	1 month	Med	1	New development
The shopper wants to work part time, but doing the groceries is a full time job at the moment.	3 months	Med	3	After the system is developed, doing groceries is going to be more effective so the shopper can work part time.
Administration has to do a lot of work, some of this work can be automated	1 month	High	1	New developments Automated system

Problem: 1 dish is sometimes prepared in 3 different ways by the 3 different cooks

PIECES Category(ies):

Cause: All chef prepare the meals from different recipes they have in their own mind.

Effect: The 3 cooks are making the same dish in a different way.

Opportunity: After the database is created we can track from which cook the different recipes are.

Possible Improvements: When we write a database for the different recipes all the cooks can use the same recipe for 1 particular dish.

Problem: clients are ordering twice or more for the same day

PIECES Category(ies):

Cause: you can only have one menu per order.

Effect: clients are trying to order more than once so they can have different menus

Possible Improvements: the system is going to track which orders are already made for one address so the clients can't order more.

Problem: recipes suggested by the clients aren't well kept up

PIECES Category(ies):

Cause: Cook Carlier has an amazing memory and knew all the suggested recipes of the client but when the company grew and there were more cooks, customers and recipes this wasn't possible anymore.

Effect: When a client asks for a recipe he suggested a while ago, Likkebaard has to ask for the suggested recipe again.

Possible Improvements: New recipes suggested by the client can be manually stored into the database so Likkebaard can reuse the suggested recipes.

Problem: There is no file with all the information of the client.

PIECES Category(ies):

Cause: Because it started as a little company, the client information was not registered but was written down when a client did an order. When an address got lost, Likkebaard couldn't find the address in their current system.

Effect: When Likkebaard tries to contact the clients they can't always do that because a lot of addresses are lost.

Opportunity: we can automate the mailing system. When there are new menus, it will be sent to all the clients

Benefit: Because the administration doesn't have to find all the email addresses, it has a lot less work.

Possible Improvements: We can store all the information about the client in the database.

Problem: The shopper wants to work part time, but doing the groceries is a full time job.

PIECES Category(ies):

Cause: Some ingredients are forgotten so the shopper has to visit suppliers for a second time

Effect: A lot of time is lost for both the shopper and the chefs, who can't start preparing some of the meals.

Possible Improvements: The new system can automatically generate a shopping list from all the orders received for that day.

problem: Administration has to do a lot of work, some of this work can be automated

PIECES Category(ies):


Cause: Everything has to be done by hand, or by one person. No digital systems are involved.

Effect: Sofie (the administrator) has to do a lot of manual work. This is very fatiguing, boring and not very efficient.

opportunity: We can automate a lot of aspects of the ordering system. Customer information and their order can be stored in a database. When there are new menus, they can also be stored.

Benefit: The prices, shopping list and delivery list can be automatically generated. Orders can be automatically checked if they are not for the same address. Also the client can use the application for making an order.

Assignment 3 : Product vision board

 VISION STATEMENT Facilitate the daily working process of “Likkebaard” catering			
TARGET GROUP	NEEDS	PRODUCT	VALUE
<p>Users: The staff. Mainly the administration who has to process a lot of manual paper work, like orders, new menu's and other data.</p> <p>Also the chefs who have to make the orders the kitchen receives and learn new recipes.</p> <p>The shopper who makes the shopping list manually.</p> <p>Lastly the deliverer who has to check if every order has been prepared.</p> <p>Customers: Existing customers of the “Likkebaard” catering service and new customers: adults who want a high quality warm meal but don't have the time, motivation or capability to cook themselves.</p>	<p>Inputting orders digitally so new clients are registered or existing clients can be selected from a database.</p> <p>Administration doesn't have to check if customer has made another order already.</p> <p>Orders are sent to the kitchen inbox automatically.</p> <p>Administration doesn't have to manually put forms there anymore.</p> <p>The standard recipe for the order is shown in the kitchen on a screen so all chefs prepare the order the same way.</p> <p>Price for every menu is calculated automatically.</p> <p>Administration and chefs don't have to do this anymore.</p> <p>Feedback is inputted in a database so this information is stored digitally and can be consulted.</p> <p>Shopping list and delivery list are automatically generated from orders received that day.</p>	<p>Digital platform to automate the data input and output of the catering process:</p> <ul style="list-style-type: none"> • Register and process orders • Store orders, customer information and feedback • Calculate order & menu prices automatically • Generate shopping & delivery list automatically 	<p>Goal: Digitize the data and information received from customers and orders to automate a lot of work that is being done manually at the moment.</p>

Assignment 4 : Requirement analysis

Event response list

External events

Event	Trigger	Source	Use case	Response	Destination
Customer makes an order	New order	Customer	Create order	- Order confirmation - Bill	Customer
Customer changes order	Change order request	Customer	Change order	Order changed confirmation	Customer
Shopper consults shopping list	Shopping list request	Shopper	Generate shopping list	Shopping list	Shopper
Shopper inputs the bought ingredients	Arrival of shopper after going shopping	Shopper	Adjust ingredients stock	Ingredients stock is updated in database	System
New recipes are discussed	New recipe	Chefs Administration	Configure new recipes	- Price of new recipe is added to database	System Customer
Deliverer consults delivery list	Delivery list request	Deliverer	Generate delivery list	Delivery list	Deliverer
Deliverer confirms that all orders are prepared	Preparation confirmation	Deliverer	Confirm prepared orders	Delivery starts notification	Customer
Deliverer confirms order is delivered	Delivery completion	Deliverer	Deliver order	Order status change	System
Chef consults recipe for order	Recipe request	Chef	Prepare order	recipe	Chef

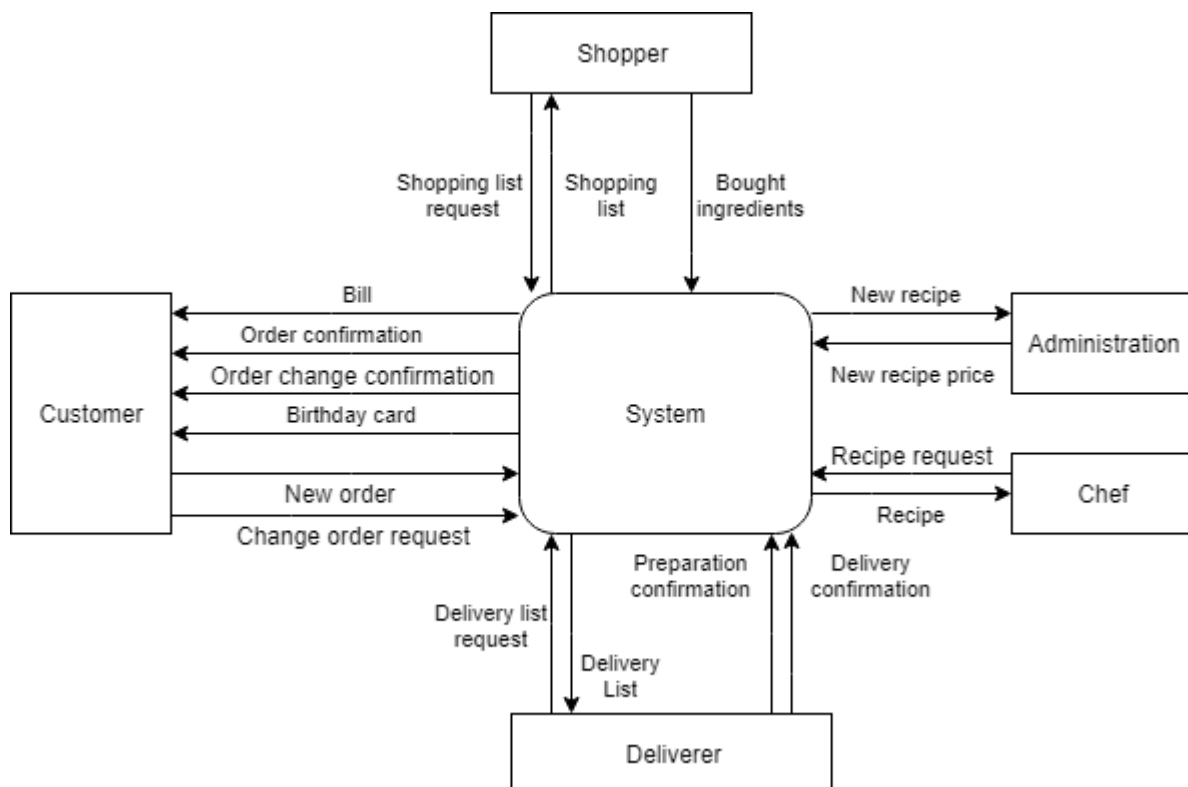
Internal events

Event	Trigger	Source	Use case	Response	Destination
Customer doesn't pick up order	Forgotten take away order	Customer	Process forgotten take away orders	Forgotten order notification	Customer
Administration sends birthday card	Customer birthday	Administration	Send birthday card	Birthday card	Customer

Temporal events

Event	Trigger	Source	Use case	Response	Destination
Price agreements with the suppliers is revised	Every 6 months	Administration	Revise prices	Ingredient prices are up to date	Internal system

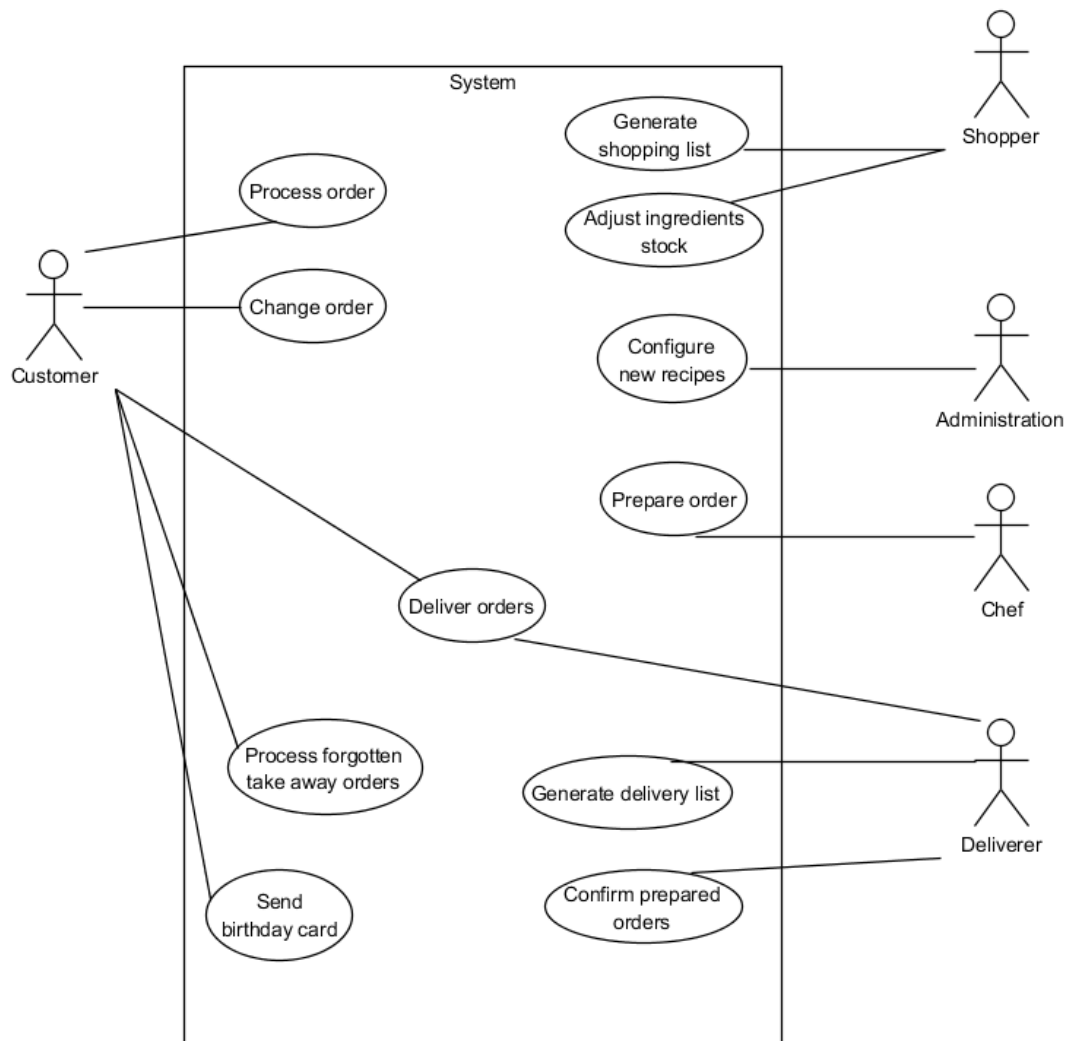
Context diagram



Use case priority

Use case	Priority	Release	Motivation priority
Create order	M	1	Without the ability to create an order, the customer can't make orders and there would be no profit
Change order	S	1	The customer should be able to change his order if he wants to, but the system works without this feature
Configure new recipes	S	2	New recipes should be added now and then to have a variety in menu's. These new recipes have to be viable to make and there prices need to be determined
Generate shopping list	M	1	Without a shopping list, ingredients can't be bought and orders can't be prepared
Adjust ingredients stock	S	2	When the shopper returns with all the bought ingredients, he can input them all into the system so it knows how large the stock is of every ingredient.
Prepare order	M	1	The chefs need to prepare the orders of the customers. This is the most important aspect of the catering service
Generate delivery list	M	1	The deliverer can get the delivery list of that day from the system.
Confirm prepared orders	S	1	The deliverer has to check if all orders are prepared before leaving for delivery
Deliver orders	M	1	It is necessary that orders are delivered to the customers
Process forgotten take away order	C	3	When a customer forgets to take away his order, it needs to be handled.
Revise prices	C	3	It would be a good idea to revise the agreed prices with the suppliers every now and then but it isn't necessary
Send birthday card	C	3	It's a thoughtful gift for customers but also not necessary

Use case diagram



Assignment 5 : Business rules – non functional requirements

Business rules:

ID	Description domain rule	Type	Static/dynamic	Source
1	Meals are delivered in the evening.	Constraint	Static	Delivery
2	Orders for the same day have to be placed before 12 o'clock in the morning en can't be changed after 12 o'clock in the afternoon.	Constraint	Static	Administration/S ystem
3	One order has to have a minimum of 4 meals.	Constraint	Static	Administration/S ystem
4	Only one menu can be delivered on one day on one address.	Constraint	Static	Administration/S ystem
5	If there are vegetarian, a menu can be combined with a vegetarian menu.	Diversion	Static	Administration/S ystem
6	When a meal consists out of 3 courses or more, a discount of 10% is given.	Event	Static	System
7	Clients can give feedback about their previous order.	Event	Static	Administration/S ystem
8	The price of a dish is the standard price of the ingredients + 40% of that price for the preparation costs and the margin of profit	Event	Dynamic	System
9	Every six months the price arrangement with the suppliers is being revised.	Event	Dynamic	Administration
10	At 5:45 in the evening the deliverer is going to deliver all the dishes in the correct order	Event	Static	Delivery
11	New menu's are placed on the bulletin board and are indicated with a complexity indicator (High, Medium, Low).	Event	Dynamic	Administration/S ystem

non functional requirements:

Reliability

- **Fault tolerance:** When there are technical problems with the system all the client information must still be accessible through a document so Likkebaard can still keep doing business.
- **Recoverability:** If the system fails, it has to be operational again the next day so that Likkebaard can keep doing business.

Usability

- **Understandability:** The administration staff has to find all the client information in a couple of seconds based on a name or an address.
- **Operability:** The system has to give a clear message/reason when something went wrong or when the user makes a mistake.

Maintainability

- **Changeability:** It should be easy to add or change new recipes or clients in the system
- **Stability:** 95% of the implemented requirements can't show unsolicited behavior when using the system

Efficiency

- **Time behaviour:** The system should show the most efficient route based on the addresses of the clients so the deliverer can deliver all the orders in time.
- **Recourse behaviour:** When there are too many orders for one day, the system shouldn't allow adding any more orders.

Functionality

- **Accurateness:** The system has to calculate the price of an order correctly, there can be no mistakes.
- **Security:** When a client tries to make a second order for the same day, the system should block that order.

Portability

- **Adaptability:** The software should be easily installed on a new device when the current one is replaced
- **Replaceability:** The software should be easily be compatible with the email application for sending mails to the clients.

Assignment 6 : Logical analysis – flow of events

Flow of events

Use case ID + Use case name: 1 Create new order
Primary actors: Customer
Secondary actors: External payment service
Short description: this use case allows a registered customer to make an order.
Preconditions: Customer is registered and has filled in his profile.
<p>Basic flow:</p> <ol style="list-style-type: none">1. Customer wants to order from website.2. System shows the list of all different menu's (Standard , Themed, Veggie and Create own).3. Customer selects standard, themed or veggie menu. A1: Customer selects create own menu. A4: Customer wants another category.4. System shows overview of all menu's and their price in selected category.5. Customer selects desired menu.6. System shows the different courses in selected menu and their ingredients A5: Customer doesn't want that menu.7. Customer orders that menu for an amount of people, delivery date and hour and chooses Take-Away or delivery. A6: Desired delivery date' order capacity is full. A14: Customer place order after 13:00 on the day of the delivery A7: The amount of people is greater than 10 A9: Customer chooses Take-Away. A12: Customer already has an order8. Customer chooses payment type9. System forwards customer to external payment system. E1: payment system unavailable.10. Customer pays for the order. A10: Account not found A11: Insufficient funds.11. Systems registers order. E2: Database unavailable.12. Systems shows order confirmation and details, and sends billing report to the customer's email.13. The use case ends.

Alternative flows:

A1: Customer select create own menu.

1. System shows overview of all different dishes and their price and an option to create an own course.

A2: Customer selects create own course.

A4: Customer wants another category.

2. Customer creates a menu from 1 or more courses (1 dish/course).
3. The flow returns to the basic flow, step 7.

A2: Customer selects create own course.

1. Systems shows a form, for ingredients and the recipe, to create a dish.
2. Customer fills the form in.
3. The chefs and administrator determine the price and if it's cookable.

A3: Not cookable.

4. The dish is added to the list of dishes.
5. The flow returns to the alternative flow A1, step 1.

A3: Not cookable dish.

1. The customer gets a notification.
2. The flow returns to the alternative flow A1, step 1.

A4: Customer wants another category.

1. The flow returns to the basic flow, step 2.

A5: Customer doesn't want that menu

1. The flow returns to the basic flow, step 4

A6: Desired delivery date's order capacity is full.

1. The system asks if another delivery date is possible.
2. The flow returns to basic flow, step 7.

A7: Amount of people is greater than 10.

1. The system checks if the order is 7 days or more after the order's date.

A8: The date is in less than 7 days.

2. The half of the total price is asked in advance, the other half will be asked the day before delivery.
3. The flow returns to basic flow, step 8.

A8: The date is in less than 7 days and the amount of people is bigger than 10.

1. The system shows an error message

2. The flow return to basic flow, step 7

A9: Customer chooses Take-Away.

1. A discount is given
2. the flow returns to basic flow, step 8.

A10: Account not found.

1. The system displays a message that the credit account was not found.
2. The flow returns to basic flow, step 8.

A11: Insufficient funds.

1. The system displays a message that there were not enough funds on the card to complete the transaction.
2. The flow returns to basic flow, step 8.

A12: Customer has more than 1 order.

1. System checks if 1 of the orders is a vegetarian menu.
A13: 2 not Veggie orders.
2. The flow returns to basic flow, step 8.

A13: 2 not Veggie orders.

1. The system displays a message that there is already an order.
2. The use case ends.

A14: Order placed after 13:00

1. System sets the order for the next day
2. Flow return to basic flow, step 8

Error flows:

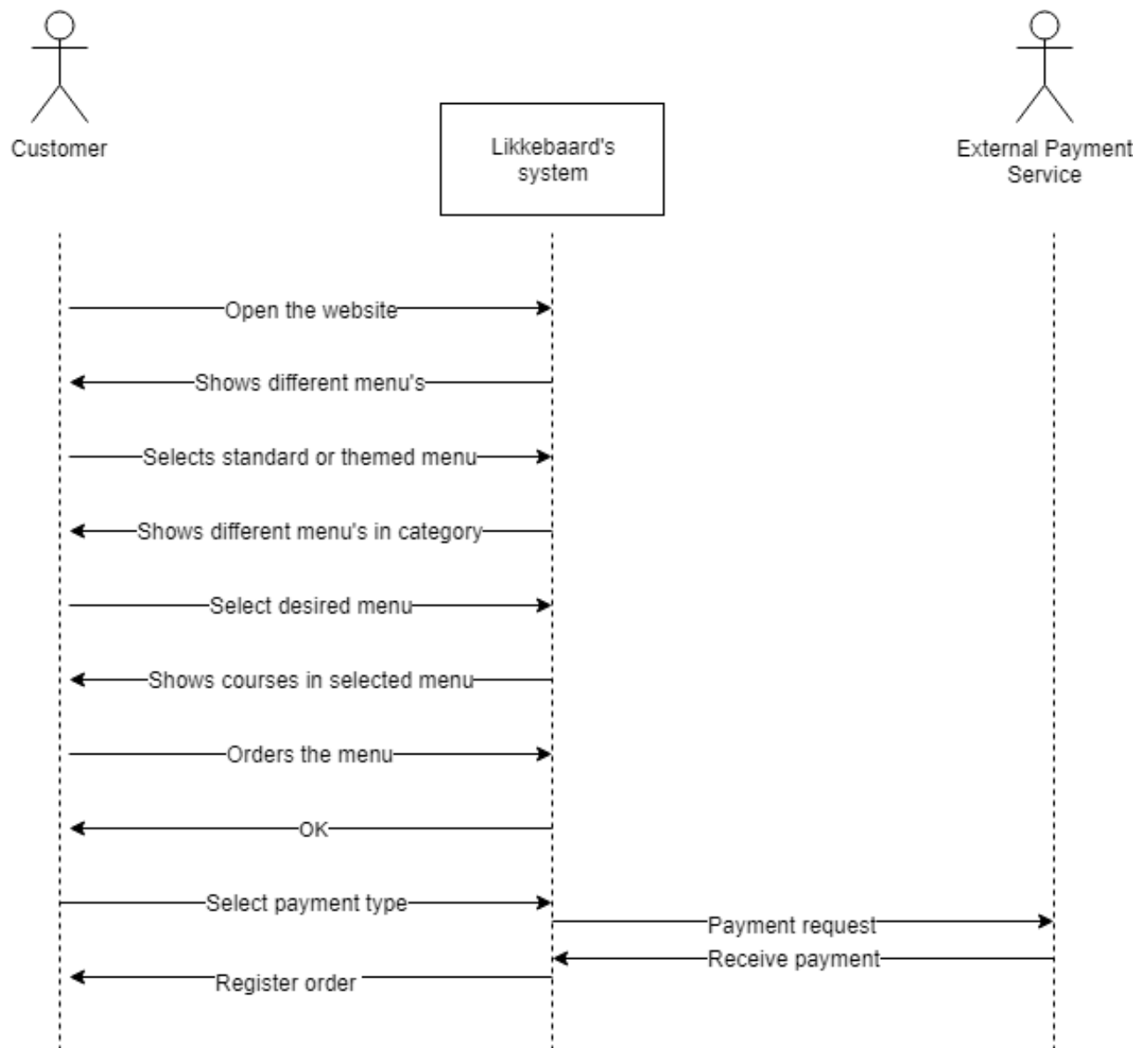
E1: Selected payment system is unavailable.

1. The system displays a message that the payment system is not available.
2. The flow returns to basic flow, step 8.

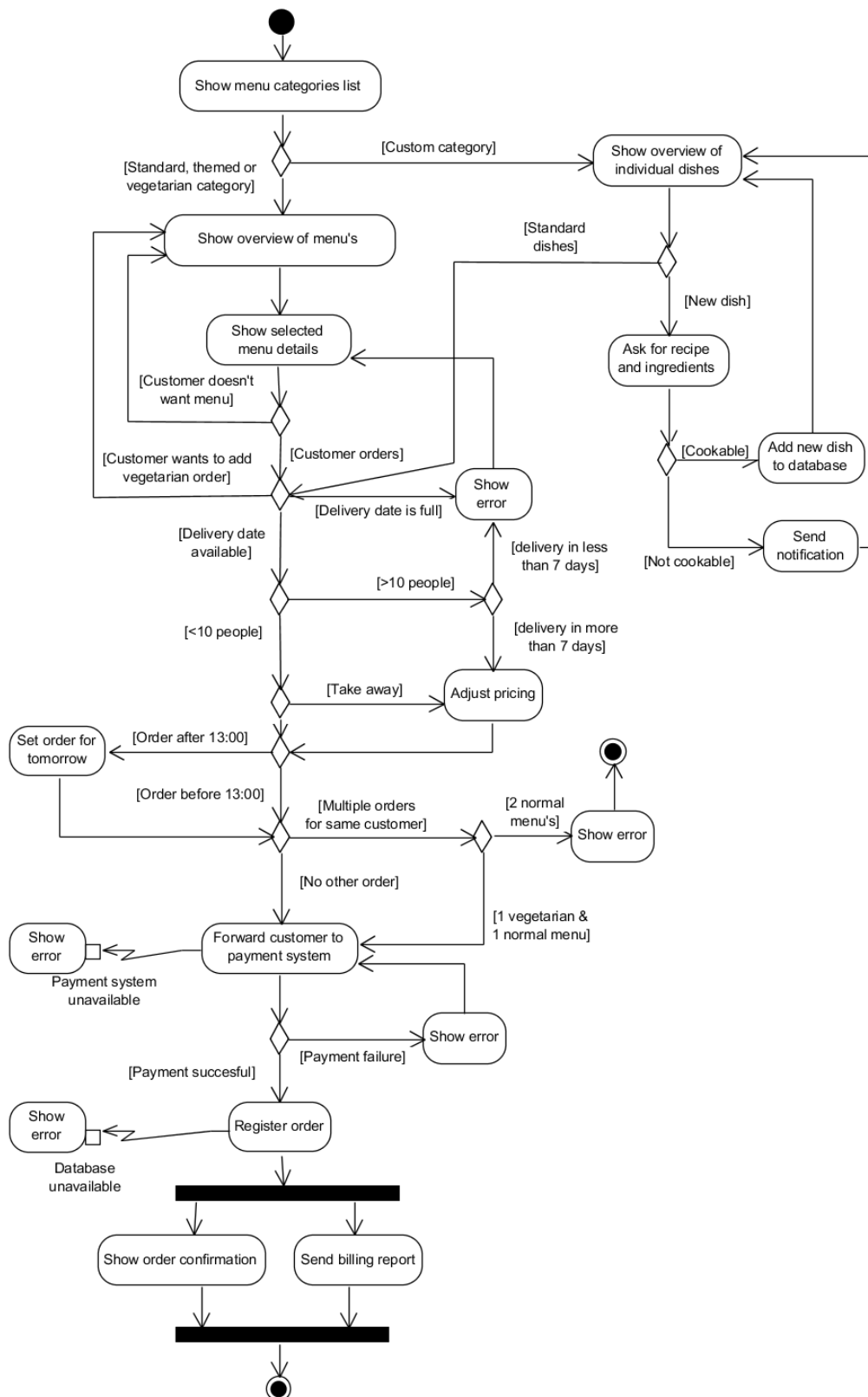
E2: Database unavailable

1. The system displays a message that the database is unavailable.
2. The flow returns to basic flow, step 11.

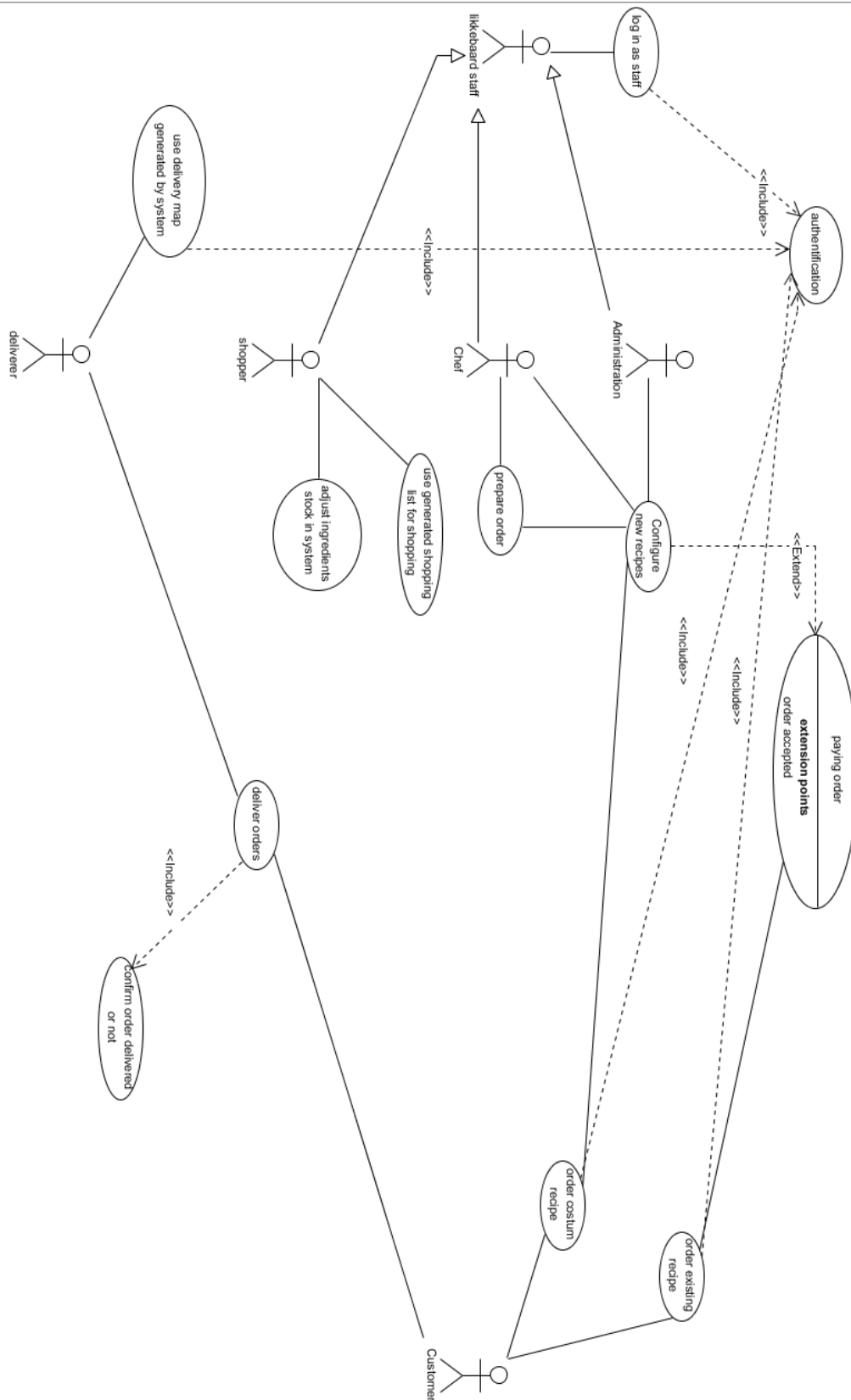
System sequence diagram



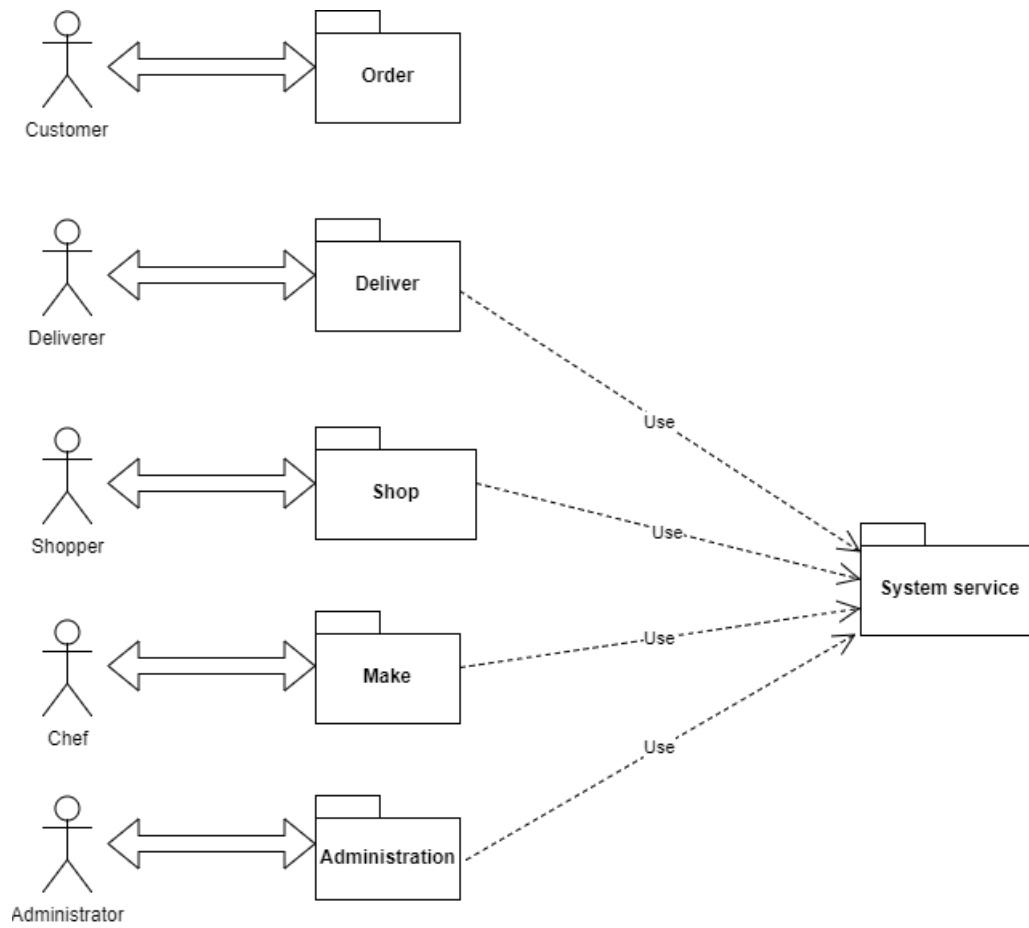
Activity diagram (*create new order*)



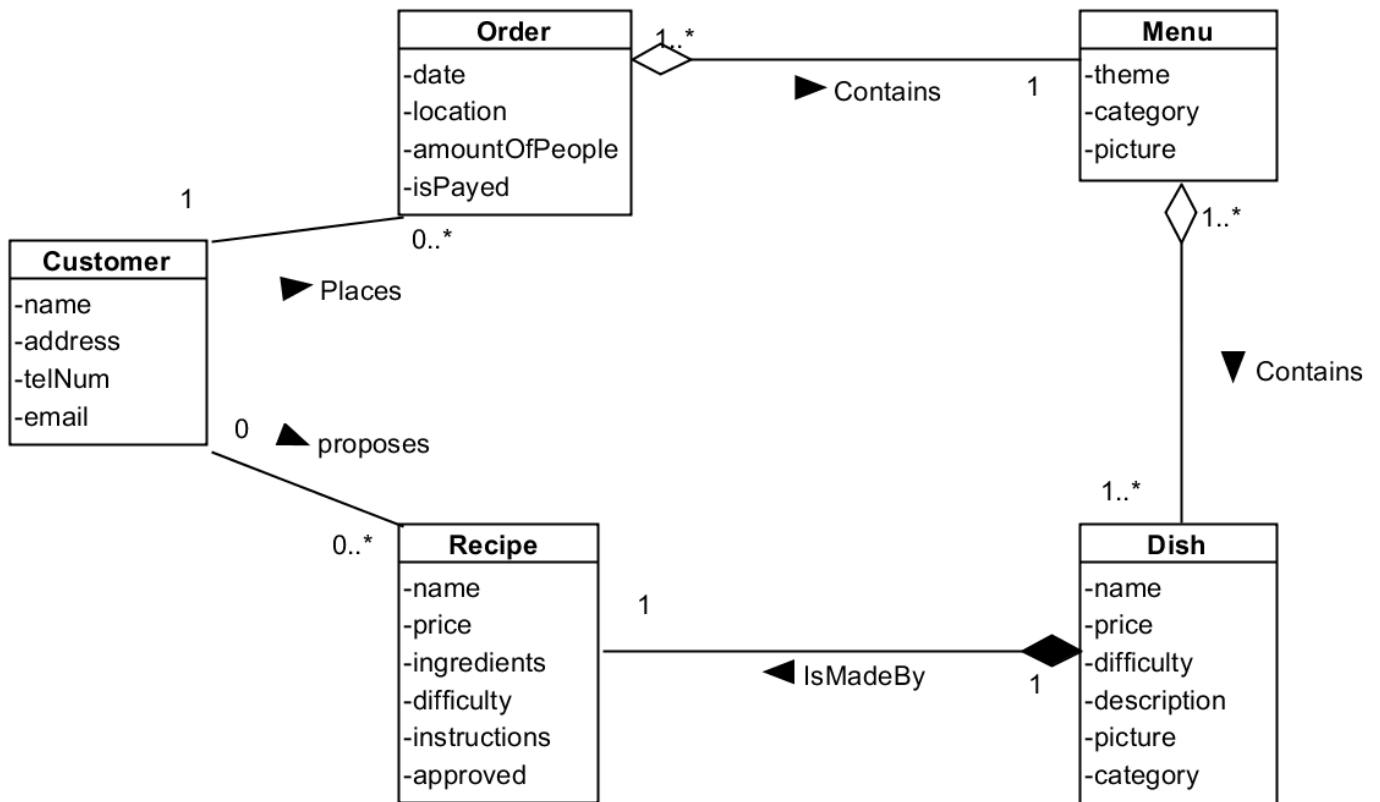
Refactored use case diagram



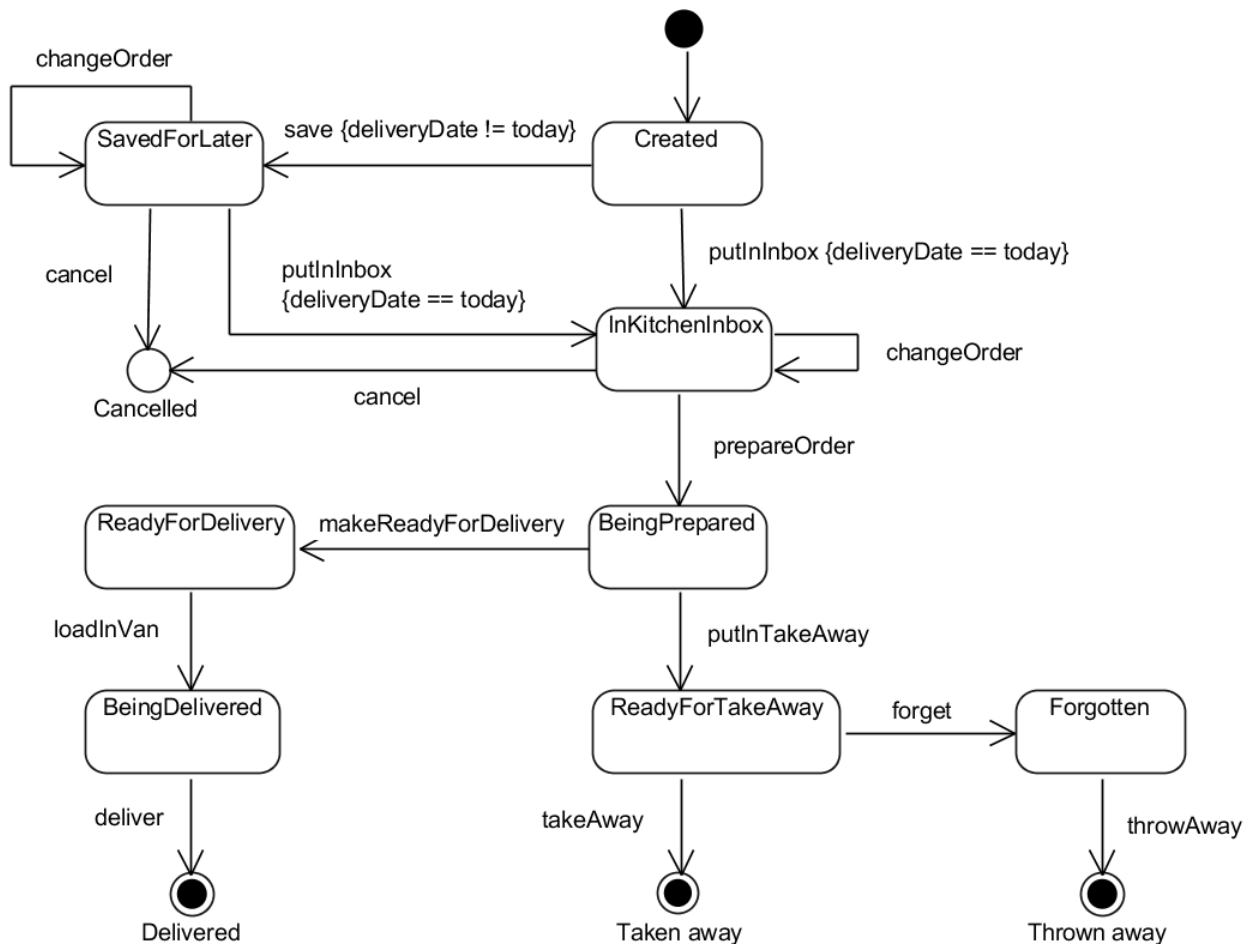
Package diagram



Domain class diagram

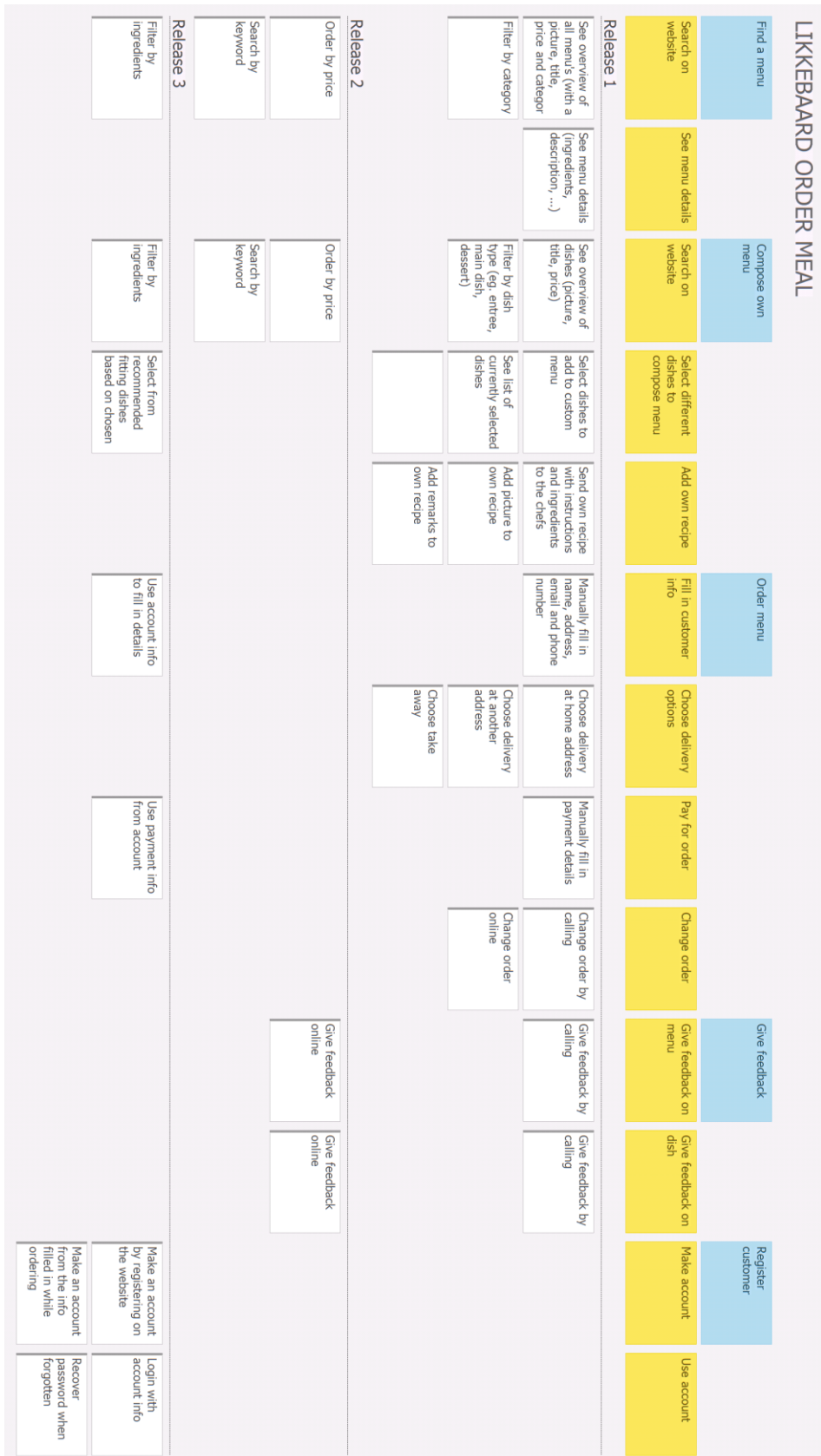


Statechart diagram



Assignment 7 : Requirement&logical analysis/agile approach

User story map



User story title: Menu overview

As a customer

I want to be able to see an overview of all menu's

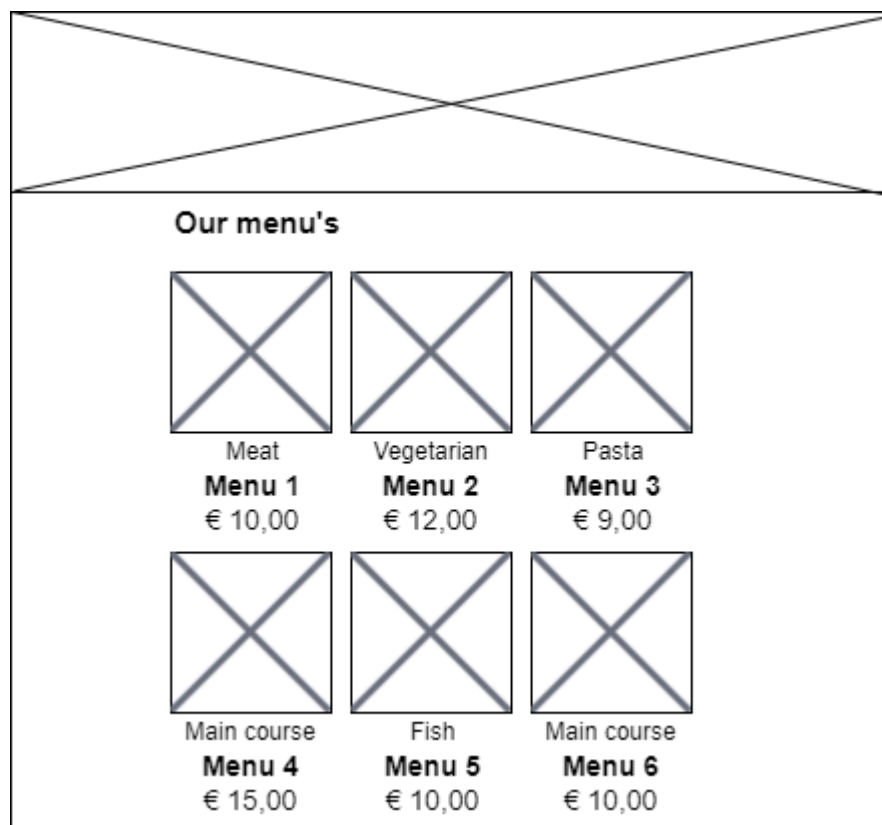
So that I can choose what menu I want to order.

Conversations

- The overview should show a grid of all the available menu's
- For every menu a picture, title, price and category is shown

Confirmations

- Click "Order" button/link
- Menu overview is shown



User story title: Select dishes for custom menu

As a customer

I want to be able to select different dishes

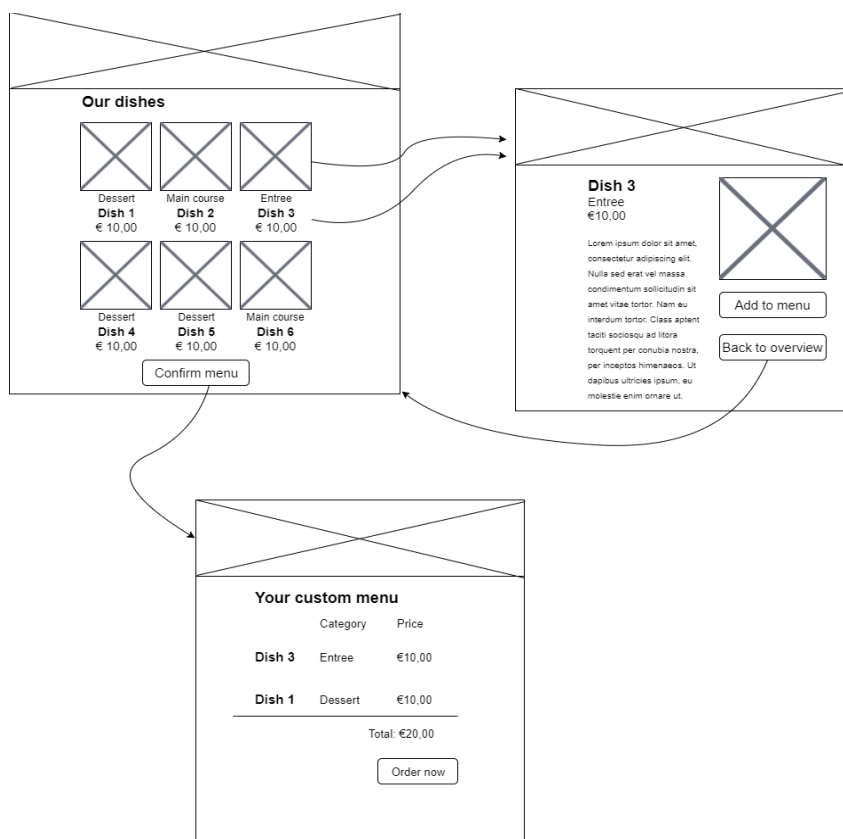
So that I can compile my own custom menu.

Conversations

- From the overview of all dishes, a dish can be added directly to the custom menu
- From the details page of a specific dish, the dish can be added to the custom menu
- When the selection is confirmed, a list is shown with the selected dishes

Confirmations

- Add dish from overview
 - Click the “Add dish” button of a dish
 - The dish is added to the custom menu
- Add dish from details page
 - Click the dish image or title
 - The details page is shown of the dish
 - Click the “Add dish” button
 - The dish is added to the custom menu
- After confirmation a list is shown
 - Click the “Confirm custom menu” button
 - A new page is shown with a list of the selected dishes



User story title: Filter the menu by category

As a customer

I want to be able to filter the menu's by a category

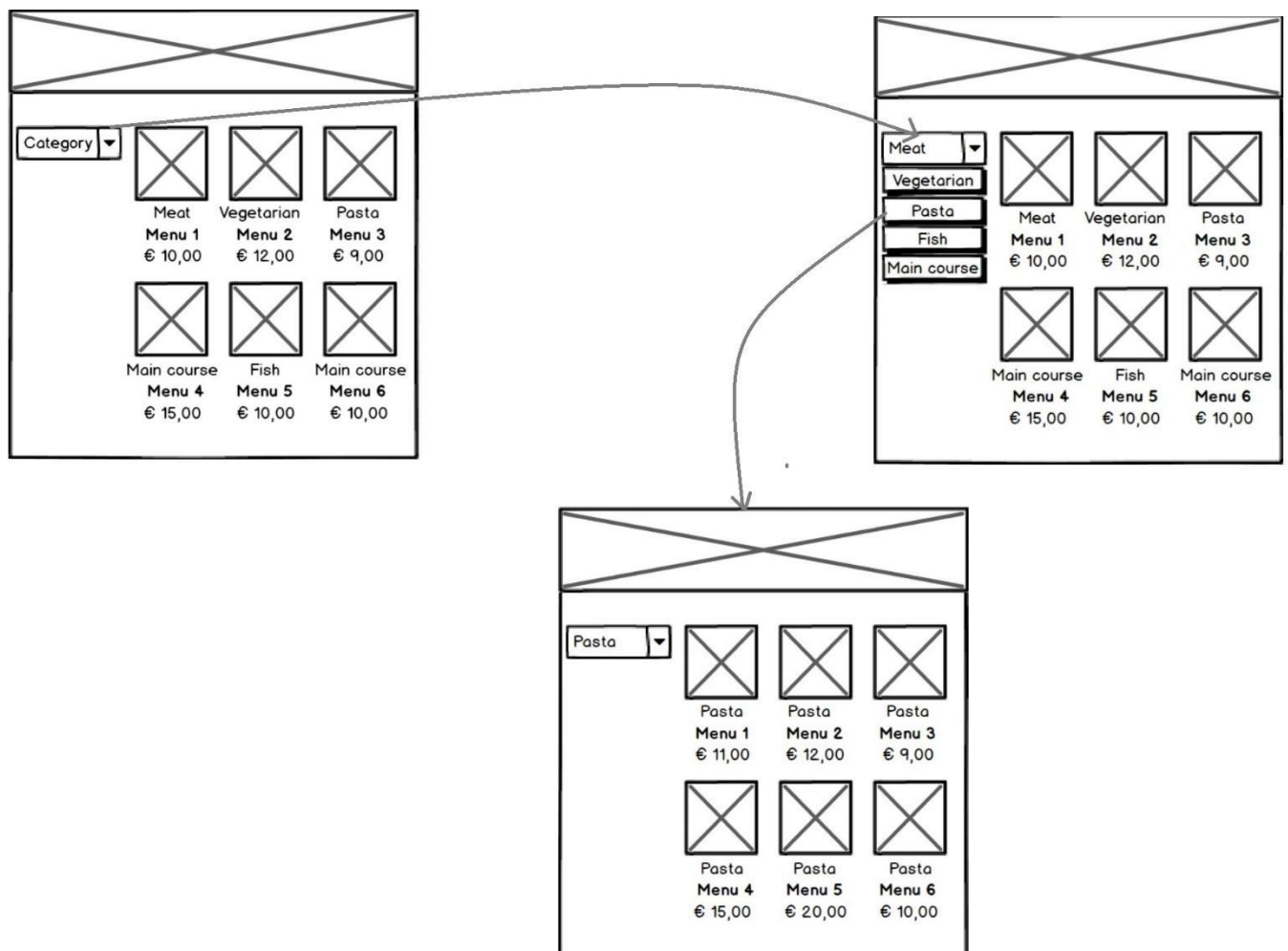
So that I can choose what kind of menu I want to order

Conversations

- From the overview all the menu's, the menu's can be sorted by a category
- When a category is selected, only the menu's with that kind of category are shown.

Confirmations

- Click "Category" button from overview
- Click the specific kind of menu
- Overview is shown with only specific kind of menu's.



User story title: Manually fill in payment details

As a customer

I want to fill in my payment details

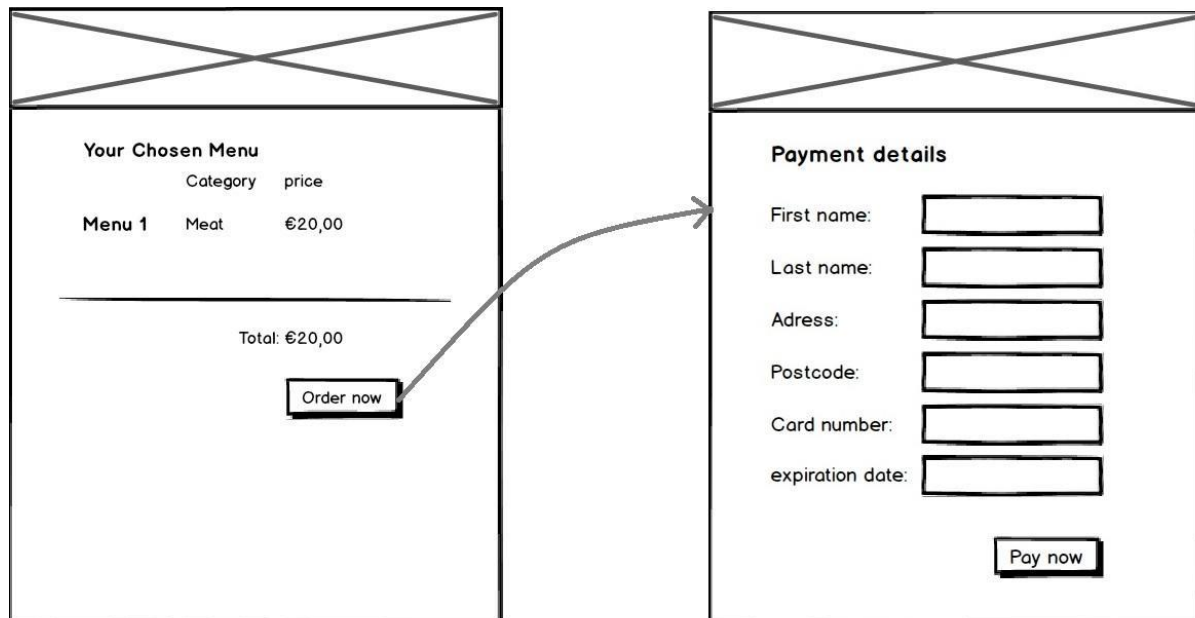
So that I can pay the order I made

Conversations

- From the shopping basket page I can confirm my order when I have all the dishes I need.
- When I confirm my order there will be a form where I can fill in my payment details

Confirmations

- Confirm order from shopping basket page
 - Click the "Order now" button
 - You redirected to payment details page
- After confirming the payment details page is shown
 - Fill in your payment details



User story title: Filter by dish type

As a customer

I want to be able to filter the type of dish I want to order in a custom menu.

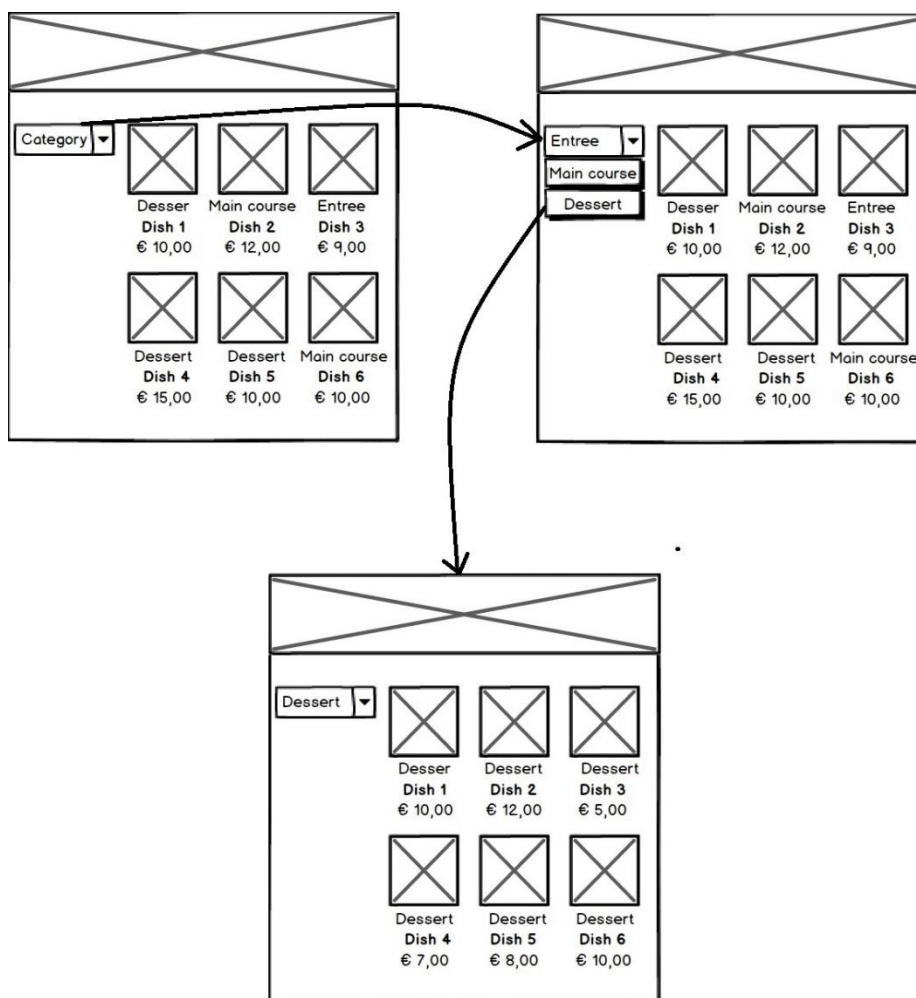
So that I can choose only the types of dishes from that category.

Conversations

- From the overview page of all dishes, a dish can be sorted by a category
- When a category is selected, only the dishes with that kind of category are shown.

Confirmations

- Click “Category” button from Overview
- Click the specific kind of dish you want
- The overview is shown with only the kind of dishes you selected



User story title: Send own recipe with instructions and ingredients to the chef

As a customer

I want to be able to send instructions and ingredients to the chef

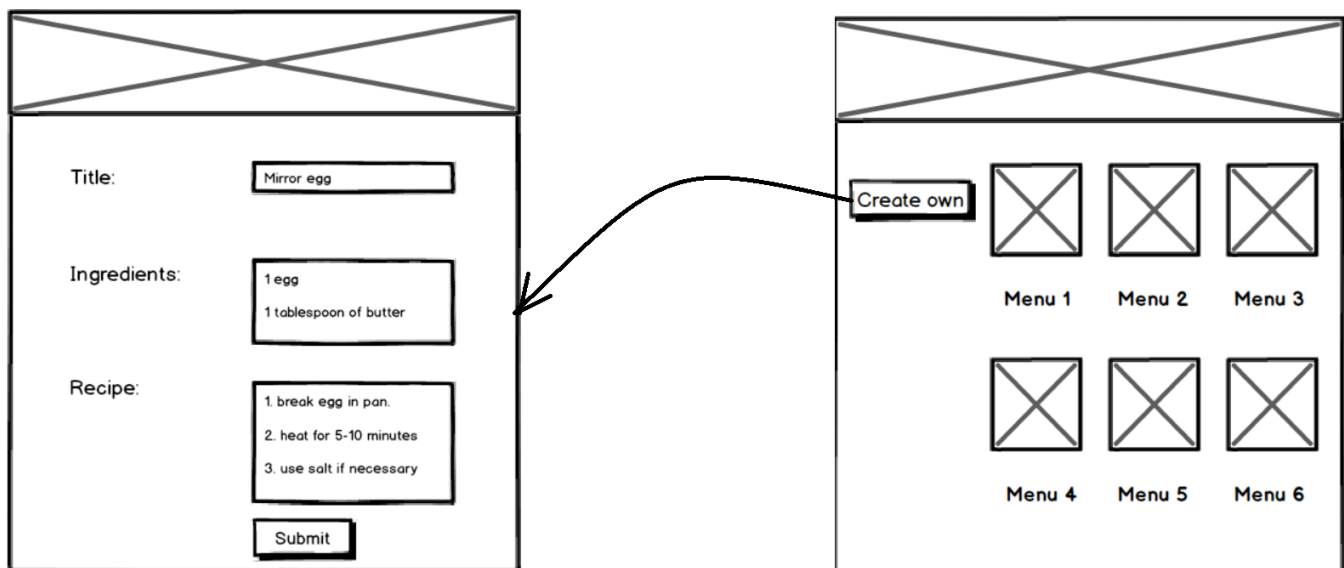
So that I can create my own recipe

Conversations

- From the overview all the menu's, there is an option to create your own recipe
- When that option is selected, a form is shown.

Confirmations

- Click "Create own" button from overview.
- Form is shown.



Assignment 8 : Technical analysis/use case realisation

Operation contracts “create new order”

Open the website

- **Responsibilities**
Request the Likkebaard catering website from the server
- **References**
Get overview use case
- **Preconditions**
 - The customer is connected to the internet
 - The Likkebaard server is online
 - A controller object exists
- **Postconditions**
 - A connection to the webserver is established

Show different menus

- **Responsibilities**
Show an overview of all available menu's the customer can order
- **References**
Get overview use case
- **Preconditions**
 - A database of menu's exists
 - The database contains at least 1 menu
 - A MenuService object exists
- **Postconditions**
 - A list of Menu objects is returned for all available menu's in the database
 - A webpage with an overview of the menu's is shown to the customer

Select standard or themed menu

- **Responsibilities**
Show only the menu's from the database that are in the given category
- **References**
Get overview use case
- **Preconditions**
 - A desired category is given
 - There is a connection to the database
 - A MenuService object exists
- **Postconditions**
 - A menu category is given to the system

- **Exceptions**

A message is returned when there are no menu's for the given category

Show different menu's in category

- **Responsibilities**

shows all the different courses from 1 particular category.

- **References**

get overview use case

- **Preconditions**

- there is a category selected
- there is a connection to the database
- A MenuService object exists

- **Postconditions**

- A list of all menus with only the chosen category is returned
- A webpage with an overview of the menu's is shown to the customer

Select desired menu

- **Responsibilities**

Choose a specific menu to see the different dishes it consists of

- **References**

Order menu use case

- **Preconditions**

- At least one menu option is shown
- The menu is available to order

- **Postconditions**

- The selected menu id is given to the system

Show courses in selected menu

- **Responsibilities**

Show the details of a given menu

- **References**

Order menu use case

- **Preconditions**

- A menu id is given
- The given menu id is a valid menu
- The menu consists out of at least one dish

- **Postconditions**

- The description, title, picture and price of the menu is shown
- The dishes out of which the menu consists are shown

Order the menu

- **Responsibilities**

The customer orders the menu

- **References**
Order menu use case
- **Preconditions**
 - A menu id is given
 - A customer id, address, email and phone number are given
- **Postconditions**
 - An order request is sent to the system.

Select payment type

- **Responsibilities**
Select a desired payment type to pay for the order
- **References**
Order menu use case
- **Preconditions**
 - An order is given
 - A customer id is given
- **Postconditions**
 - The desired payment type is given to the system

Payment request

- **Responsibilities**
Make a payment request to the payment system for the chosen payment type so the customer can pay for his order.
- **References**
Order menu use case
- **Preconditions**
 - Customer info is given
 - The payment system is available
 - The customer has enough money to pay for the order
- **Postconditions**
 - The payment service has received the request so it can process the payment

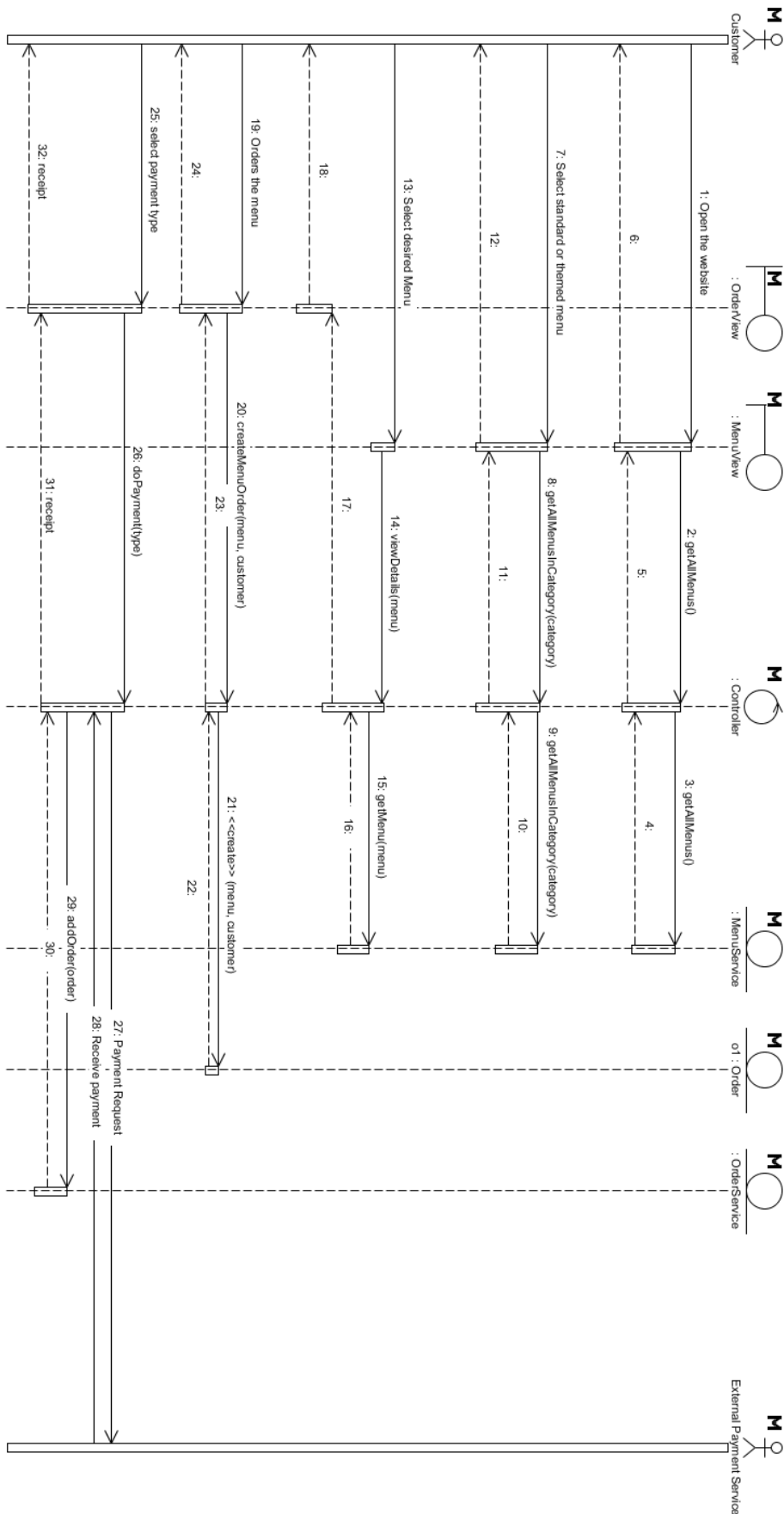
Receive payment

- **Responsibilities**
Likkebaard gets a notification that the payment was successful
- **References**
Register order use case
- **Preconditions**
 - A payment request has been sent
- **Postconditions**
 - An order confirmation with the details of the menu and payment is returned
 - An email is sent to the customer with the bill and order details

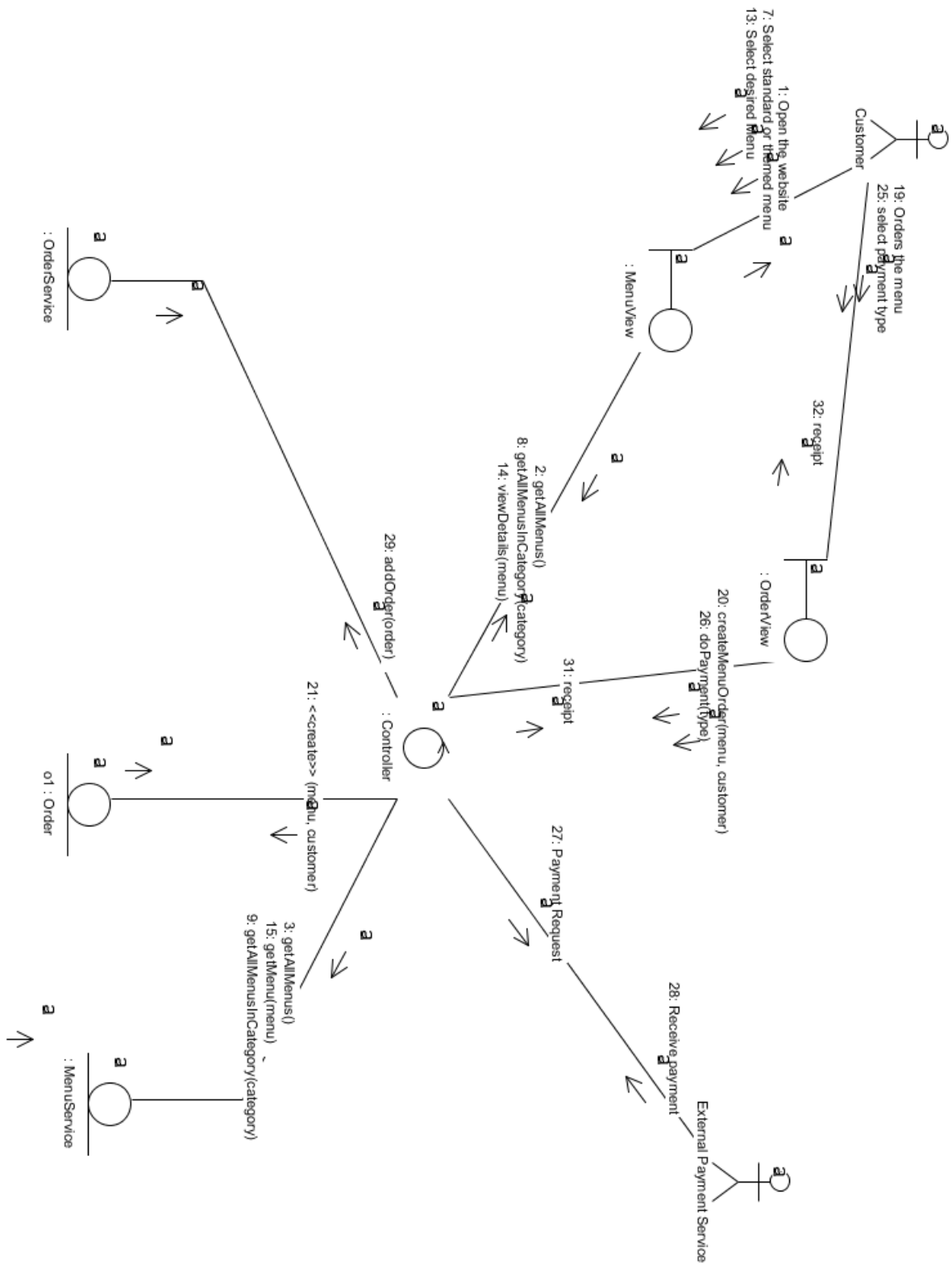
Register order

- **Responsibilities**
Notify that the payment was successful and that the order is registered
- **References**
Register order use case
- **Preconditions**
 - The payment was successful
- **Postconditions**
 - The order is registered

MVC Sequence diagram “create new order”



Communication diagram “create new order”



Design class diagram “create new order”

