29.10.2021 Zealand – Sjællands Erhvervsakademi

**UML 1 – Big Mamma Gastronomia**



Group 2:

Digna Bringule [dign0009@edu.zealand.dk](http://dign0009@edu.zealand.dk)

Tea Koscina [teax0669@edu.zealand.dk](http://teax0669@edu.zealand.dk)

Szymon Mateusz Zwara [szym0277@edu.zealand.dk](http://szym0277@edu.zealand.dk)

Bartol Cagalj [bart0707@edu.zealand.dk](mailto:bart0707@edu.zealand.dk)

TABLE OF **Contents**

[1.](#_heading=h.gjdgxs) Abstract 2

[2.](#_heading=h.30j0zll) User Stories 2

[3.](#_heading=h.2et92p0) Product backlog 5

[4.](#_heading=h.tyjcwt) Domain model 6

[5.](#_heading=h.3dy6vkm) Class Diagram 7

[6.](#_heading=h.1t3h5sf) Sequence diagram 8

[7.](#_heading=h.4d34og8) Conclusion 9

## Abstract

For the first version of Big Mama Pizzeria the goal is to make a system where the customers can order a pizza. To successfully implement this code proper documentation is needed. In this case that will be: user stories, domain model, class diagram and sequence diagram. All previous mentioned forms of documentation represent the structure and functionality of application that will help the implementation of the code.

## User Stories

A user story is a general, informal explanation of a particular software feature from the perspective of the user. It is used to describe the end functionality that the user should receive[[1]](#footnote-0). For this report 10 user stories about ordering a pizza have been made, in order to describe the most important features of the solution. They are as follows:

1. As an existing user I want to log-in so that I can access the specific part of the system that is meant for me.
   1. AC - Given I’m a registered customer when I log-in then I want to access the menu and order system
   2. AC - Given I’m a staff member when I log-in then I want to access all managerial and sales tools
   3. AC - Check if the password and email are correct
   4. AC - Check if the password recovery option is working correctly

Non-functional requirements:

1. If user inputs correct information show a login-in success message
2. If a user inputs incorrect information display which input is wrong
3. Have a response time be less than 1 second
4. As a new user I want to be able to register so that I can use the application.
   1. AC - User is prompted to register once they open the application
   2. AC - A registration form is opened with: email, password, name and phone number
   3. AC - The credentials should be entered in the correct format
   4. AC - A confirmation email should be sent to the specified address

Non-functional requirements:

* 1. Make the user input the password twice, if it does not match display a message
  2. After the user confirms their email take them to the account confirmed webpage
  3. Have the response time be less than 1 second

1. As a customer I want to access the menu so that I can choose what to order.
   1. AC - All menu items should have prices listed
   2. AC - The menu items should be grouped and enumerated
   3. AC - Each menu item should have specified ingredients, allergens and toppings

Non-functional requirements:

1. All the menu items should have picture next to their listing
2. The menu should load all the assets under 3 seconds
3. As a customer I want to be able to filter menu items so that I can see only the preferred groups.
   1. AC - Option to sort by food types (pizza, sandwich, drinks etc.)
   2. AC - Option to sort by dietary preferences (vegan, vegetarian, gluten free etc.)
   3. AC - Option to sort by price range

Non-functional requirements:

1. Different types of sorting should be allowed (ascending, descending...)
2. When changing sorting options, the re-listing should not take more than 3 seconds
3. As a customer I want to be able to select menu items and put them in my cart so that I can make an order.
4. AC - The chosen menu items should be placed in the “cart”
5. AC - The customer should be able to edit items already in the “cart”
6. AC - The customer should be able to individually edit items with toppings and other ingredients when adding them to the “cart”

Non-functional requirements:

1. The cart icon with number of items in the cart should be shown in website corner
2. The cart should be stored in the cookies and saved if the user leaves/closes the website
3. As a customer I want to be able to choose between different payment options so I can pay for the order.
4. AC - Different options for payment should be displayed at checkout
5. AC - The payment information should be entered in the correct format and verified upon completion
6. AC - Payment success/failure message should be displayed

Non-functional requirements:

1. All the payment methods should be listed before the purchase so the user can know if they have an appropriate payment method
2. As a staff member I want to receive notifications about new orders so that I can prepare the orders in time.
   1. AC - When a new order is placed the notification should pop up on the staff member screen
   2. AC - The order should contain all the customer preferences along with their name
   3. AC - When the order is ready a notification should be sent to the customer
   4. AC - If the ordered can’t be prepared for some reason the staff member should be able to contact the customer or cancel the order

Non-functional requirements:

1. The notifications show up as toast messages that can be dismissed
2. The notification should be visible not later than 10 seconds after an order was placed
3. As a staff member I want to be able to edit the menu so I can edit the menu items like their price, availability, and information
   1. AC - For a staff member an edit icon should be displayed next to the menu items
   2. AC - When editing an item, a save button should be at the bottom at the bottom if something was changed
   3. AC - Extra information is shown to the staff member like internal ID number and similar

Non-functional requirements

1. The change should be saved within 1 second after clicking the button
2. The change should be visible instantly after it’s done
3. As a customer I want to receive a receipt by email after paying for my order so I can see how much I paid.
   1. AC - The email should be sent to the email address specified by the customer.
   2. AC - The email should contain all the information about the order, e.g. pizzeria information, menu items, prices, total price.

Non-functional requirements:

1. The email should be sent within 60 seconds from paying for the order.
2. The email should contain a message with a phone number to customer service in case of any errors.
3. As a user I want to be able to see how much time my order will take so that I can plan the pick-up time
   1. AC - After order confirmation, message with approximate pick-up time should appear on website
   2. AC - The time should update based on the input from staff
   3. AC - When the order is ready the user should be notified

Non-functional requirements:

1. The update of data should happen every 2 minutes
2. The notification should come with a sound

## Product backlog

The user stories can be placed in a product backlog, which helps to catalogue them, add priority etc. From there a few stories can be selected and put into a sprint backlog, which is then developed during a specific sprint. T-Shirt estimation was used for defining the scope of the stories. The product backlog for the Big Mama Pizzeria solution is as follows:

| PRODUCT BACKLOG | | | |
| --- | --- | --- | --- |
|  |  |  |  |
| User Story ID | User Story | Estimate (size) | Priority |
| US003 | As a customer I want to access the menu so that I can choose what to order. | Medium | 1 |
| US002 | As a new user I want to be able to register so that I can use the application. | Large | 2 |
| US001 | As an existing user I want to log-in so that I can access the specific part of the system that is meant for me. | Medium | 3 |
| US005 | As a customer I want to be able to select menu items and put them in my cart so that I can make an order. | Large | 4 |
| US006 | As a customer I want to be able to choose between different payment options so I can pay for the order. | X-Large | 5 |
| US007 | As a staff member I want to receive notifications about new orders so that I can prepare the orders in time. | Small | 6 |
| US008 | As a staff member I want to be able to edit the menu so I can edit the menu items like their price, availability and information. | Medium | 7 |
| US009 | As a customer I want to receive a receipt by email after paying for my order so I can see how much I paid. | Small | 8 |
| US004 | As a customer I want to be able to filter menu items so that I can see only the preferred groups. | Small | 9 |
| US010 | As a user I want to be able to see how much time my order will take so that I can plan the pick-up time. | Large | 10 |

**Table 1 - Product backlog with user stories**

## Domain model

A domain model is a conceptual model of the domain, that includes both behavior and data. It gives a general overview of the class logic in the solution. The domain model that was made for the Big Mama Pizzeria ordering system is as follows:

In this version of the program only the 3 classes that are highlighted in the box are going to be implemented. The two remaining ones are concepts that might be implemented in the future updates of the software.

## Class Diagram

A class diagram is similar to the domain model, but it adds additional information about the methods used in the classes, as well as access types for the instance fields and methods. The class diagram for the Big Mama Pizzeria ordering system is as follows:



The order class is the main class of the program, which is going to contain both the Pizza and Customer objects. The methods implemented in that class are AddPizza and AddCustomer, that are both designed to create two objects respectably. The CalculateTotal method is designed to calculate the total price for the order.

The Customer, Pizza and Topping are simple classes, that only contain instance fields with the necessary information. The Menu class contains a list of Pizza objects, that could later be used to display the menu to the user.

The relationship between the Customer and the Order class is association with the direction arrow towards the Order, because the Customer has an Order. The relationship is one to many in the direction of the Order and one in the direction of the Customer, because the customer can have either one or many orders, and the order can be only assigned to one customer.

The relationship between the Order and Pizza class is composition with the direction arrow towards the Pizza, because if you delete the Order class, the Pizza class will also be deleted. The relationship is many towards the Order, because the Pizza object can be assigned to many

and it’s one to many in the direction of the Pizza, because you can have one or many pizzas on an order.

The relationship between Pizza and Topping is association with the direction arrow towards the Topping, because the pizza has a topping. The relationship is zero to many in the direction of the Topping, because the pizza can have either zero or many toppings and it’s many towards the pizza, because the pizza can have many toppings.

The relationship between Pizza and Menu is aggregation with the direction arrow towards the Menu, because the Menu is a collection that aggregates Pizza objects. The relationship is one towards the Menu because the pizza can only be in one menu and it’s many towards the Pizza because the menu contains many pizzas.

## Sequence diagram

A sequence diagram or a system sequence diagram (SSD) is a type of interaction diagram because it describes how and in what order a group of objects work together. It shows the interaction logic between the objects in the system and the time it takes,



This sequence diagram is a simple visual representation of the time and data flow during the execution of the program. The main program wasn’t depicted since only the store start method is called from there. Since the central class of the program is the Order class the first thing that is created. This class contains method AddCustomer which is called next.

With this method a new customer object is created and then stored in an internal variable. Similar process is repeated for the pizza object with the AddPizza method. Both of those methods pass their respective information to the class constructor.

After all required objects are created and stored in the order class the Calculate total method is called. This method takes all the prices of individual pizza objects and applies the delivery and tax cost returning the value as double. After printing the program waits for the user to press a key a then terminates.

## Conclusion

To maximize effectiveness and completely streamline the project workflow good documentation is needed. This comes in the form of user stories, which are a very simple method to describe a sometimes-complex task concisely. From those user stores additional diagrams can be created. The domain model and the class diagram show relationships between parts of the program. Domain model chart emphasizes the general layout while the sequence diagram is shows with more detail all the variable types and methods, the actual implementation. Finally, the sequence diagram is used to show the passing of time during program execution. All these visual representations are used to help everyone evolved in the project to easily see implement the requested features. Also, this can be used to spot any potential problems of oversights early in the development cycle which in turn reduces potential losses.

A relatively simple program is required for the Big Mamma Gastronomia prototype and in turn the documentation is simple and easy to understand. Only a few classes are present in the program. The interaction between classes is not complex and consists of only a few functions. Since this is only a prototype it is important to limit the scope of the project to only that which is required. Also, not departing from what is stated in the documentation is an important factor.

Finally, this documentation is a great foundation to base future additions. It also facilitates further development and implementation of more complex systems. This is exactly the use case of this prototype and what is depicted in the documentation. This is of benefit to both the projects owners and the project team.

1. Max Rehkopf, *‘User Stories with examples and a template’*, <https://www.atlassian.com/agile/project-management/user-stories> [↑](#footnote-ref-0)