Bakery-Group12

Requirements and Design

Group 12 – Object Oriented-Design Summer 2014

Jarred Lien – [jlien15@ccs.neu.edu](mailto:jlien15@ccs.neu.edu)

Hussein Abou Nassif Mourad – [hassuni9@ccs.neu.edu](mailto:hassuni9@ccs.neu.edu)

Table of Contents

*The Problem* 3

*User Stories* 4

*Use Cases* 5

*Functional and Non-functional Requirements* 6

*Design* 7

*UML* 8

*Contributions* 9

***The problem:***

You work at a software development company, and Dr. Schmidt is your boss. Your company has been hired by a bakery to develop software to track the bakery’s customers and orders. The system should be developed in Java and should be well-designed using object-oriented approaches.

The bakery would like to store customer information for contacting and billing the customers. The bakery also has a frequent customer loyalty card for which they can track purchases and give out rewards. The bakery also would like to store all of the order information. Through the software, the owner hopes to be able to understand his sales better.

The bakery owner should be able to use the software to see all of the available items, see all of the items ordered by a given customer, calculate order totals, and track loyalty card totals. The bakery is still deciding exactly what is expected of the system.

Given the basics of a bakery, your boss wants to hear your proposal for the project in the format of requirements and design documents. What features should the bakery software have? Given the type of information that will be stored and operations performed, what would your design be? What objects would it include and how would they relate to one another?

You will meet with your boss along with the other teams in class on Thursday, June 12, 2014 to finalize the expectations of the project. During this meeting, you should be prepared to discuss your documents and your ideas for the project, including operations or queries that the bakery owner and other potentially users will be able to perform with the software.

Your boss will be available for brief meetings on June 10 and June 11 to answer general questions you may have.

**User Story 1:**

Lionel is throwing a pre-World Cup party the night before the opening match. He has contacted the bakery, Champion's Bakery, with a large order to be picked up. He ordered 5 cakes, 5 pies, 100 doughnuts (50 glazed, 25 jelly, 15 powdered, 10 chocolate), 200 cupcakes (100 chocolate, 100 vanilla), and 200 cookies (100 sugar, 100 chocolate chip). He is a loyalty card member and this is his tenth order, so he is qualified for a 20% off discount. The order records his name and his phone number. His first and last name is Lionel Messi. His phone number is 800-GOL-ATHO. He has chosen to pay for the order with cash. Champion's Bakery Inventory before the order from Lionel was 20 cakes, 20 pies, 150 doughnuts, 200 cupcakes, and 500 cookies.

**User Story 2:**

Ronaldo wants to send his friend, Pepe, who is in the hospital some bakery goods. Ronaldo orders 1 bacon flavored cake, with 3 cucumber cupcakes. After having one bite of each, Pepe decides to return these items. He was unsatisfied with the cake and cupcakes, so he called Champion's Bakery and asked for a refund. However, the bakery has a refund policy where if you are not the buyer of the items, you cannot get a refund. Ronaldo gives his name, address, and phone number.

**Use Case 1:**

Owner of Champion's Bakery wants to view the orders of the current week.

He goes into the system and types in "view this week's orders."

The system prints out all of the orders within the last 7 days.

**Use Case 2:**

Owner of the bakery wants to add a new special flavor to the menu.

He goes into the system and types in "add new bakery item."

The system adds the given bakery item into the inventory with its description.

**Use Case 3:**

A customer places an order but wants it delivered a week from the order date. The system offers today’s date (order date), as well as the pick up date.

**FUNCTIONAL REQUIREMENTS**

- System computes the total amount of an order

- User gives contact information

- System outputs the estimated the order date, as well as the pick up date

- The owner can add and remove items from the menu

- Orders total cost is computed

- Menu is outputted into the screen with all the bakery items

**NON-FUNCTIONAL REQUIREMENTS**

- Secure - The owner is the only one that can add and remove from the menu

- Constraints

- Performance

- System will run in an efficient manner such that it will not keep the customer waiting (or blow up).

**DESIGN**

Module dependency diagram:

--------------------------

+-------------------------+ +----------------------+ +--------------------------+

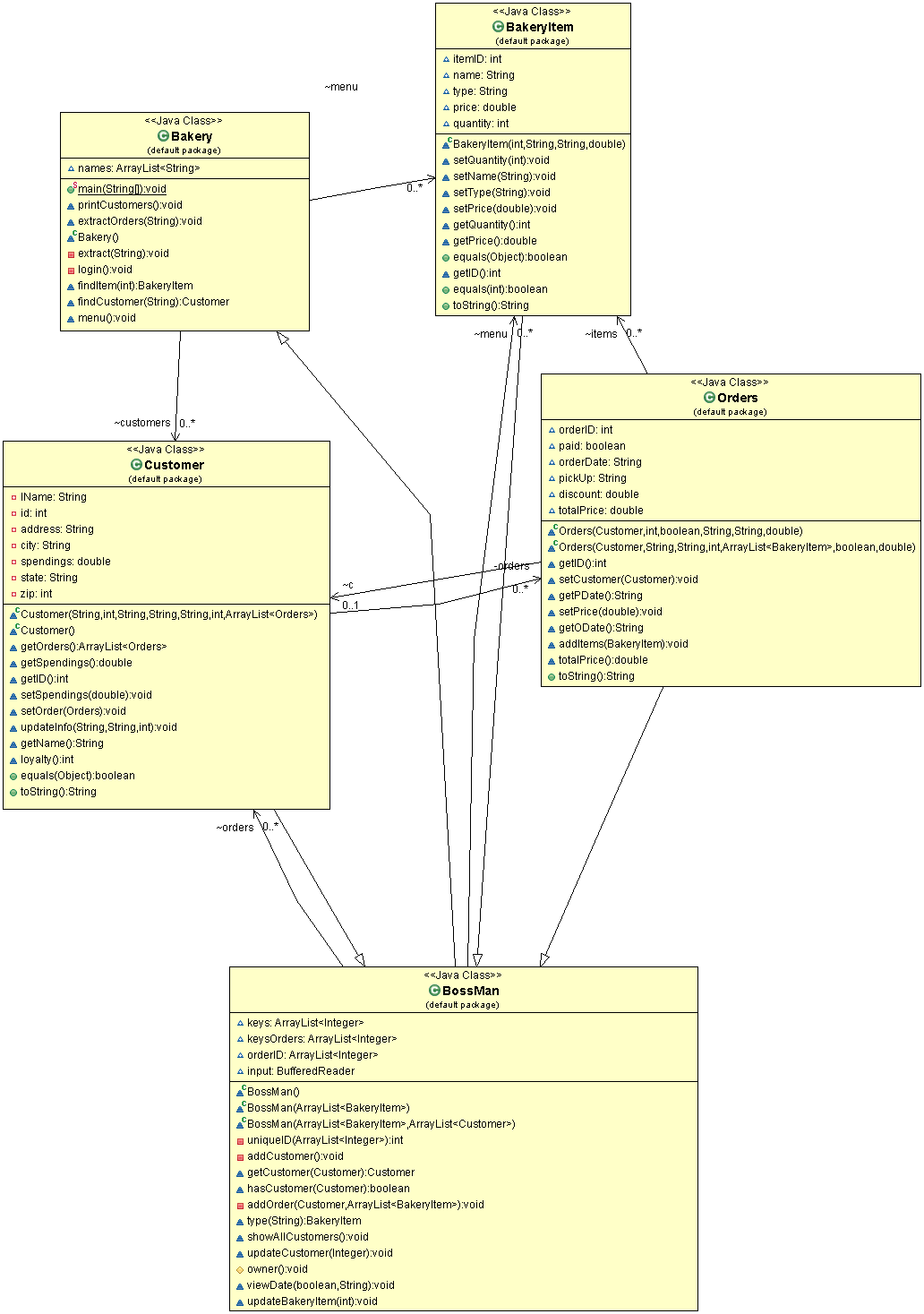
| Customer Interface | <------> | Bakery Manager | <------> | Employee Interface |

+-------------------------+ +----------------------+ +--------------------------+

UML diagram:

--------------------------

**UPDATED UML:** On the next page, we have updated the UML to include the additional methods that we didn’t have before.

***C***

***Contributions:***

Jarred Lien

- Came up with User Story 1

- Came up with User Case 1 & 2

- Worked on the non-functional requirements

- Created the Module Dependency Diagram

Hussein Abou Nassif Mourad

- Came up with User Story 2

- Came up with User Case 3

- Worked on the functional requirements

- Created the UML

- Created the Cover Page