Requirements and Design Documents

Robert C Jensen

rcj@ccs.neu.edu

&

Anh Tran

panhtran249@gmail.com

Team #3

Table of Contents

Requirements

Design

Appendix

Requirements

User Stories

1. Anya calls the bakery and asks if they have some cheesecakes. Rob (the bakery owner), consults the Bakery Software to examine the inventory. He goes to the "Cakes" section and finds that there are 5 cheesecakes remaining. He passes this along to Anya, who gleefully purchases all of them for a cheesecake party.

2. Basil had placed an order a month in advance for his wedding. He ordered three cheesecakes, two mincemeat pies, 10 tiramisu and one Boston crème pie. Two days before his wedding, he calls the bakery asking Rob to confirm that he ordered the right amount of tiramisu. His fiancé loves tiramisu and she wanted to make sure that he ordered a lot. Rob consults the Bakery Software to review Basil's order. He asks Basil for his order number and enters it into the software's search function. Basil's most recent order appears, along with some old orders for bread. It turns out that he meant to order 20 tiramisu for his hungry fiancé. Basil is so grateful that his mistake was corrected that he invites Rob to the wedding to drink champagne.

3. Bear loves red velvet cupcakes. He loves them so much that he visits Rob's bakery every day around 4:00pm to purchase ten. Bear signed up for Rob's loyalty card program. Each dollar spent is worth one point. Once Bear accumulates 50 points, he gets one free cupcake and all the points are deducted. Today is Bear's birthday. As per usual, he goes to Rob's bakery to purchase his cupcakes. Rob enters the order into the Bakery Software, additionally entering the loyalty card number. Bear's card entry indicates that he has 50 points. A message appears on the screen: "Bear gets one free cupcake!" The points are reset to zero. Rob relays the good news to Bear who is ecstatic. Bear was so happy that he bought 2 more cupcakes.

Functional Requirements

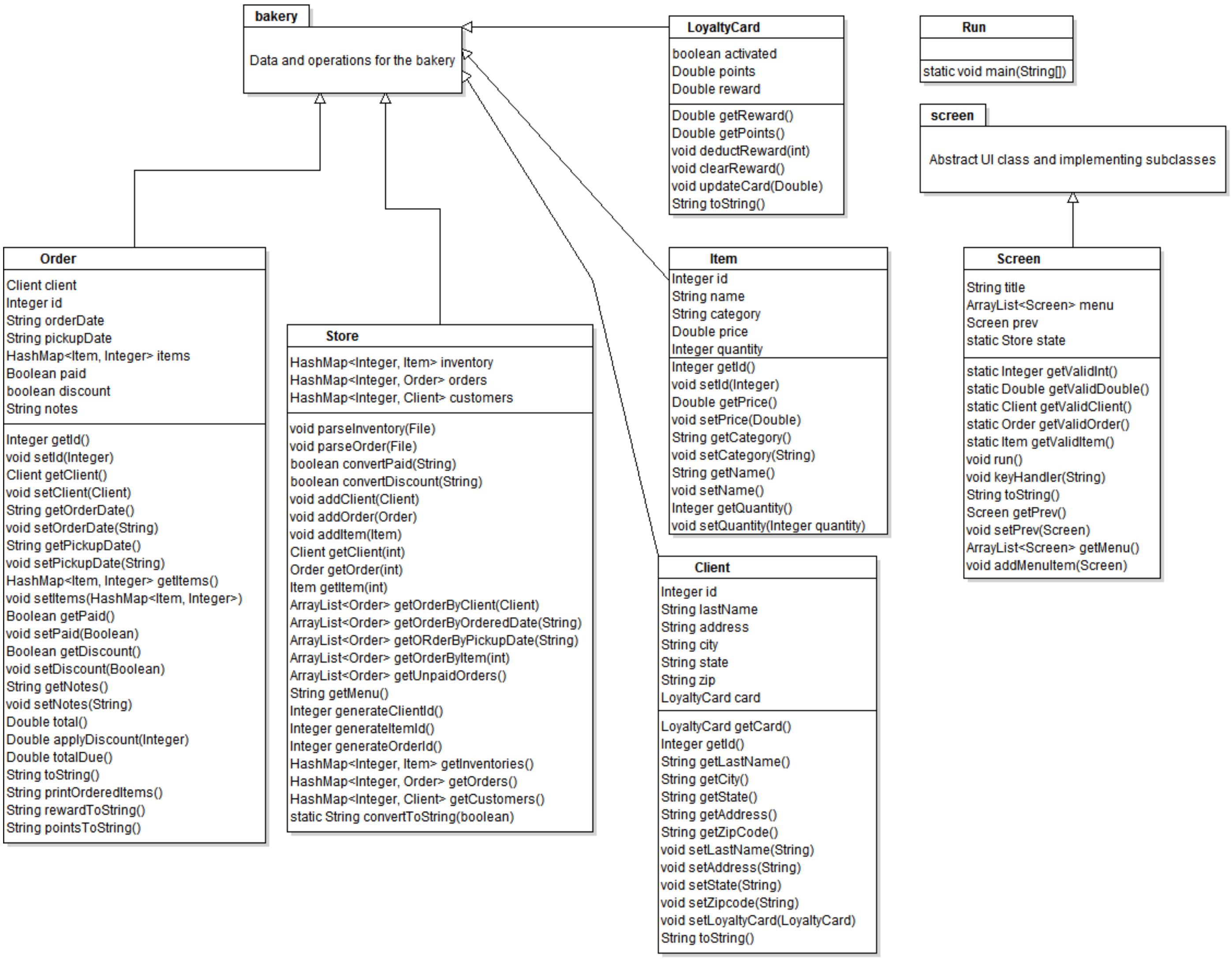
The bakery will be able to perform these operations: add and update a customer/item/order into the database, get a receipt after each new order is added, view the menu, client’s information, loyalty card status, and retrieve orders by searching for a client, order date, pickup date, item type or payment status.

Non-Functional Requirements

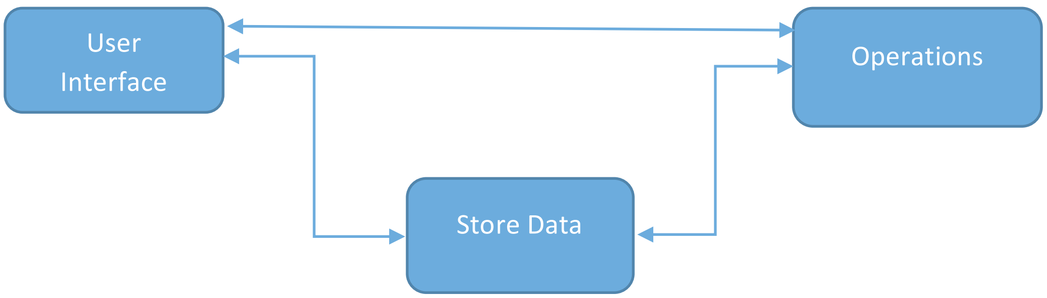
The bakery will be able to store information, relating to the customers, their orders and what is available in the store, such as customer’s last name, address, city, state, zip code, loyalty card(current points and reward balance, and whether the card is activated or not). The store manager will be able to interact with the system using a menu-driven console UI.

Design

UML



Module Dependency Diagram



Appendix

Student Contributions

Robert C Jensen – UML diagram & user stories

Anh Tran – Requirements list, concept & design