```
          ,------.                ,--.                  ,------.                        ,--.
|  .-.  \  ,---. ,--,--, |  |,-. ,---. ,--. ,--. |  .-.  \  ,---. ,--,--, ,--.,--.,-'  '-. ,---.
 |  |  \   :| .-. ||   \   |     / | .-. : \  '  /   |  |  \  :|  .-.  ||   \   |  ||  |'-.  .-'(  .-'
 |  '--'  /'   '-' '|   ||  ||  \  \\  --.    \  '    |   '--'  /'  '-'  '|  ||  |'  "  '  |  |  .-'  `)
`-------' `---' `--''--'`--'`--'`----'.-'  /    `-------'  `---' `--''--' `----'    `--'  `----'
                                       `---'
```

Bakery13
Nicholas Scheuring
scheuring.n@husky.neu.edu
Justin Bosma
bosma.j@husky.neu.edu
Team # 13
Bakery Management System - Donkey Donuts

BakeryManagementSystem13

1. **Introduction**

Bakery Management System

Console based interactive Bakery Management System for management of inventory, ordering, and customer information.

BMS tracks customer information for contacting and billing customers.
BMS tracks Loyalty Card purchases and rewards.
BMS tracks and stores all order information and presents reports
to Bakery Manager to understand sales better.
BMS tracks all of available menu items.
BMS tracks all items ordered by a given customer.
BMS calculates order totals.
BMS calculates Loyalty Card totals.

Main class must be run either from console or command line with no arguments to access program main menu.

2. **Use Cases**

```
a) Bakery Manager logs into system - Default password “Donkey”
        Bakery Manager starts system
>[Prompt] “Enter the password to access system”
        >[System compares password to Bakery Password Field]
        >[If system grants access] Main Menu Displays
        >[If system does not grant access] >[Prompt] “Enter the password to access system”

b) Bakery Manager creates a new item
        BM at Main Menu
>[Enter Item Options Menu] “Press X for Item Options Menu”
>[If validate input] Item Options Menu Displays
>[If not validate input] Main Menu Displays

>[“Enter Item Creation Menu”] “Press X to add a menu item”
>[If validate input] Add a menu item method displays Prompts
>[If not validate input] Item Options Menu Displays

c) Bakery Manager creates a new customer
        BM at Main Menu
>[Enter Item Options Menu] “Press X for Customer Options Menu”
>[If validate input] Customer Options Menu Displays
>[If not validate input] Main Menu Displays

>[“Enter Item Creation Menu”] “Press X to add a Customer”
>[If validate input] Add a customer method displays Prompts
>[If not validate input] Customer Options Menu Displays

d) Bakery Manager creates a new order
        BM at Main Menu
>[Enter Item Options Menu] “Press X for Order Options Menu”
>[If validate input] Order Options Menu Displays
>[If not validate input] Main Menu Displays

>[“Enter Item Creation Menu”] “Press X to add a Order”
>[If validate input] Add an order method displays Prompts
>[If not validate input] Order Options Menu Displays

e) Bakery Manager generates a Loyalty Card totals report
                BM at Main Menu
>[Enter Item Options Menu] “Press X for customer options menu”
```

>[If valid input] Display customer options menu
>[If not valid input] Main Menu displays

>[In customer options menu] “Press X for loyalty card totals”
>[If valid input] Return totals of loyalty cards
>[If not valid] Customer options menu displays

f) Bakery Manager generates an order totals report (Annual/Monthly/Weekly)
BM at Main Menu
>[Enter Item Options Menu] “Press X for order options menu”
>[If valid input] Display order options menu
>[If not valid input] Main Menu displays

>[In customer options menu] “Press X for order totals”
>[If valid input] “Press X for Annual, Y for Monthly, Z for Weekly
>[If not valid] order options menu displays

g) Bakery Manager generates an Inventory Report for menu items
BM at Main Menu
>[Enter Item Options Menu] “Press X for Item Options Menu”
>[If validate input] Item Options Menu Displays
>[If not validate input] Main Menu Displays

>[“Enter Item Creation Menu”] “Press X to Generate an Inventory Report”
>[If validate input] Inventory Report Displays
>[If not validate input] Item Options Menu Displays

3. **Non-Functional Requirements**

The Bakery Management System should have these qualities:
Modular
The ability to add new data structures and make links between existing structures.
Secure
Only Bakery Manager and system administrators should be able to access the system.
Reliable
Functions that are run by both Manager and administrators must have predictable behaviors.
Maintainable
The guiding maintenance principle will be to "Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live." *John F. Woods*
Performance Oriented
Should run fast.

4. **Functional Requirements**

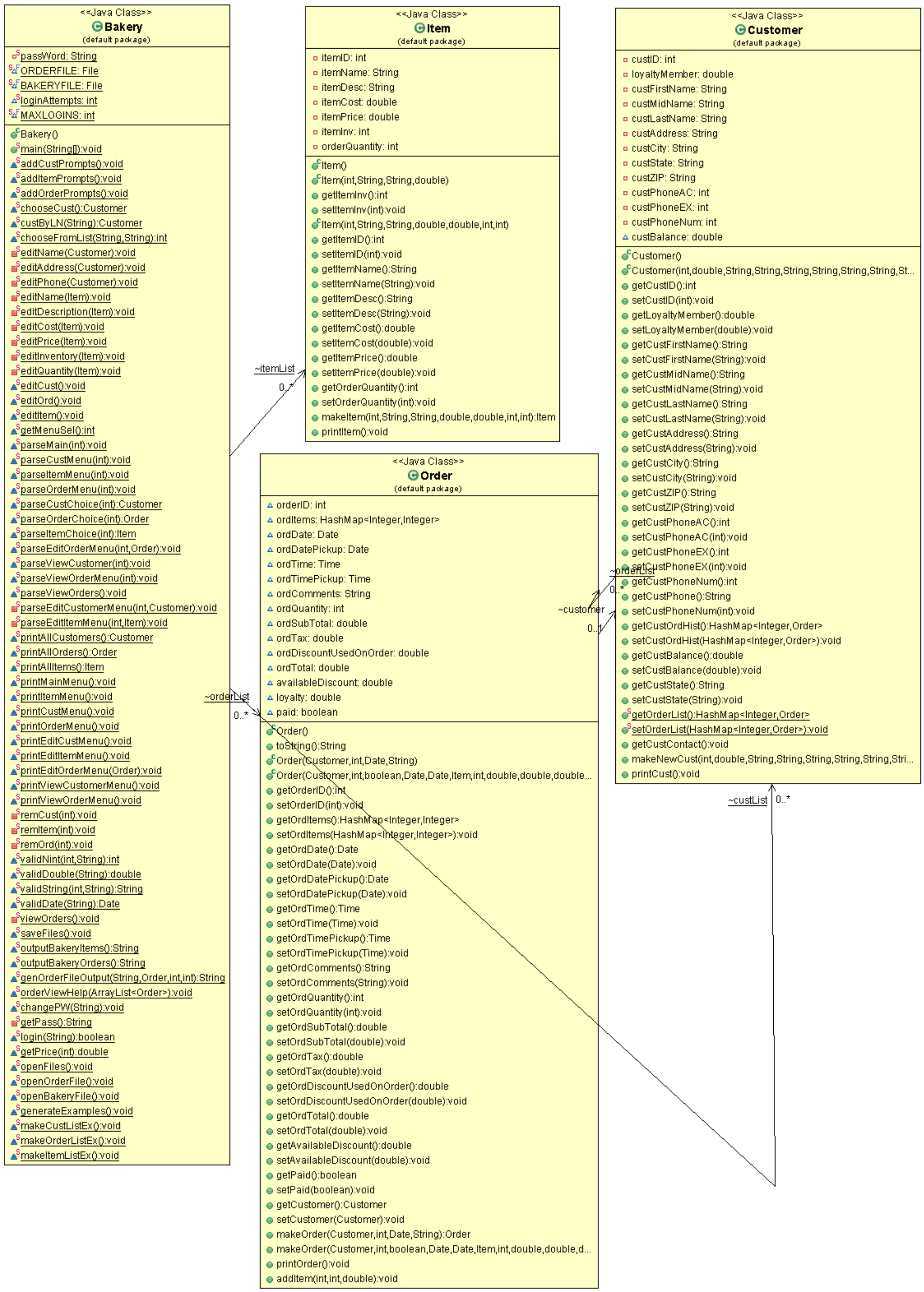
A. Customer -
1. Add new customer to system.
2. Retrieve customer info: name, number, billing amount, etc
3.Modify customer info.
4.Modify frequent customer loyalty

B. Stock/Orders -
1. Display current inventory
2.Show items ordered by customer
3.Calculate the order totals
4.Track loyalty card totals

5. **Platforms**

Developed using Java 1.6 with Eclipse. Java can be run cross-platform.

6. **UML Diagram**



More may be added/altered as client wishes.

7. **Module Dependency Diagram**

```
Bakery Manager → Bakery Data ← User Interface
                   |
                   |
                   V
        Subclass Data (Item, Order, Customer)
                   |    ^____^______^
                   V     |
            Algorithms/Operations
```

8. **Team Member Contributions**

Nicholas Scheuring

Introduction

Use Cases

Functional Requirements

Module Dependency Diagram

UML

Code Contributions:

Menus in bakery class (Main, Item Options, Order Options, Customer Options)

User entry validation (validNint, validString, validDouble, validDate)

Open Files - Save Files - Password Options

Justin Bosma

Introduction

UseCases

Non-Functional Requirements

Functional Requirements

Platforms

Code Contributions:

Menus in bakery class (Edit Menus, Remove Menus for Item, Order, Customer)

Edit Operations - Remove Operations (Item, Order, Customer)

Class operations and helper methods (addItem etc…)

Pair Programmed - Fixed bugs and completed unfinished individual tasks. Refactoring the Bakery class to use HashMap structures instead of originally planned ArrayList structures. Completed functionality and added menu options, menus, interactions, and operations as needed.

9. **Possible Extensions**

1. GUI for a more modern user experience
2. Mult-User Extensibility with security enhancements
    3. Online extendability for ecommerce purposes
    4. More detailed reporting capabilities
5. Automation/Scheduling of ordering and discounting procedures