

System tests 3/4
SD 6.5/9
CD 5/5

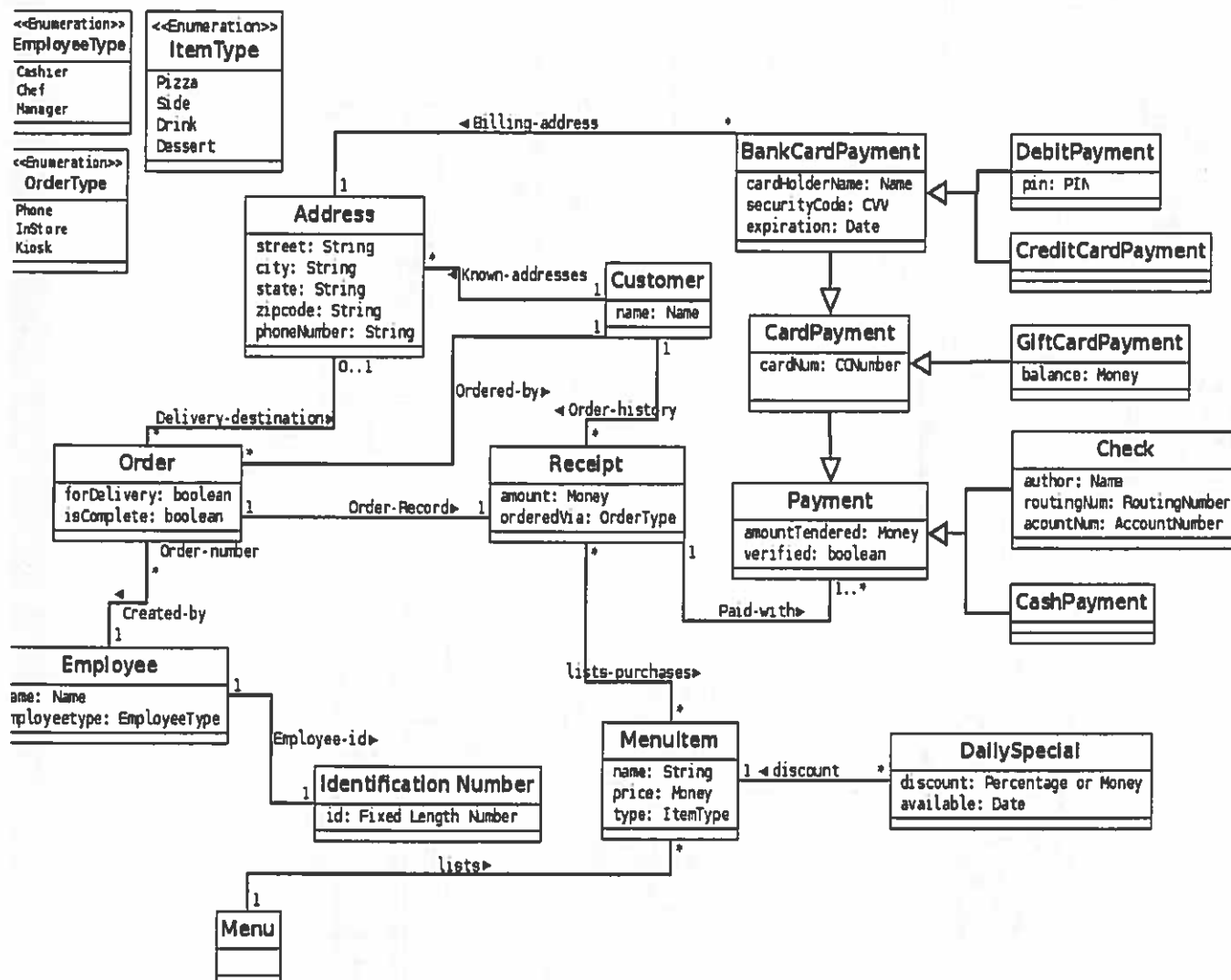
CS414 Group-H
Bryce Myers
Troy Cogburn
Sky Egelhoff
Nate Gillard

Assignment 4: Design and Implement Iteration I

Assumptions:

1. This pizza ordering system is designed from a in-store register perspective. Meaning the application is designed to be used by a Employee of the pizza restaurant.
2. The ordering system does not keep record of customers to maintain a database of orders or customer record. Each order has a one to one association to a customer and each customer has at most one address, depending on whether the order is for delivery or not.

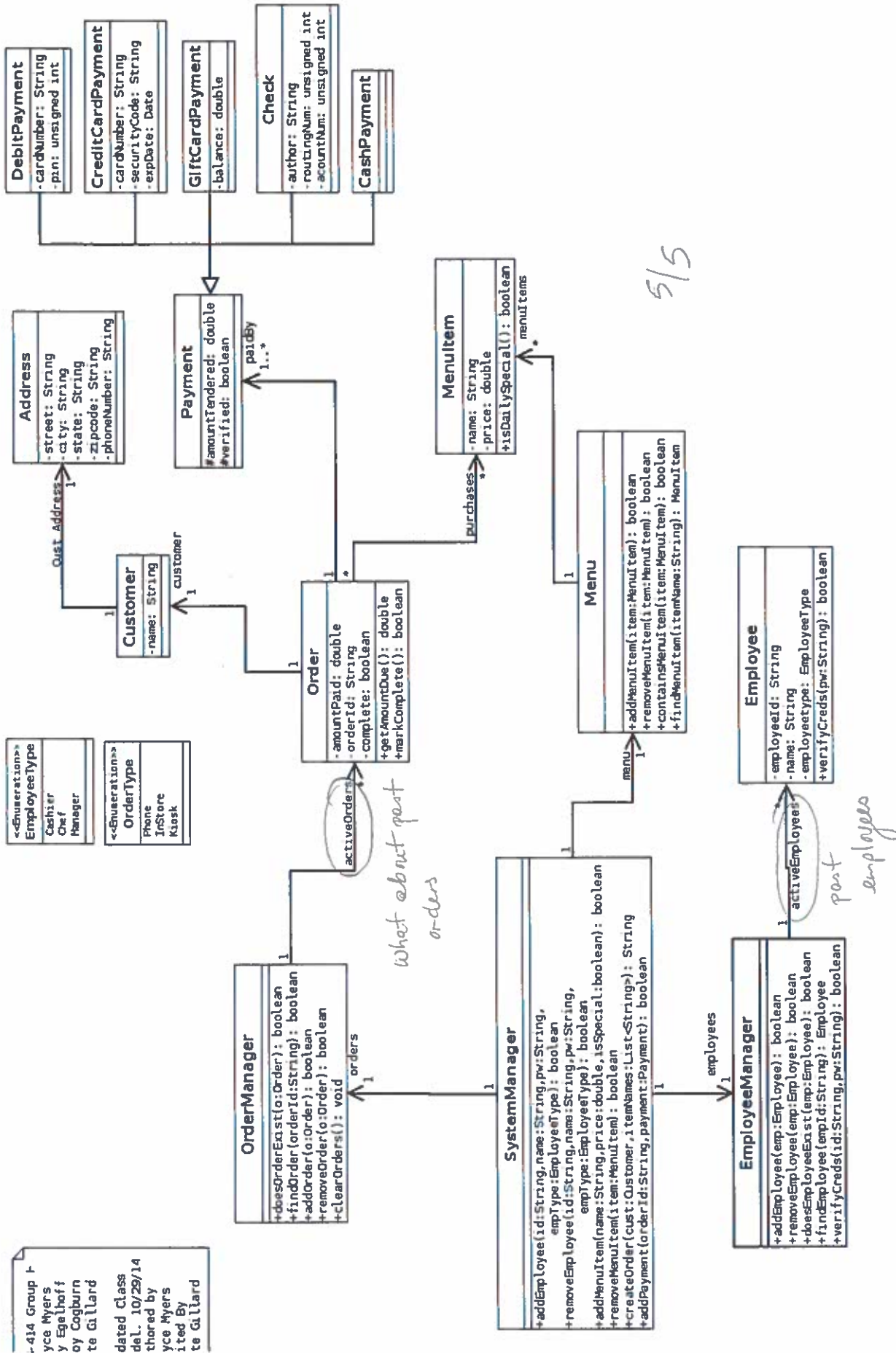
Updated Domain Model:



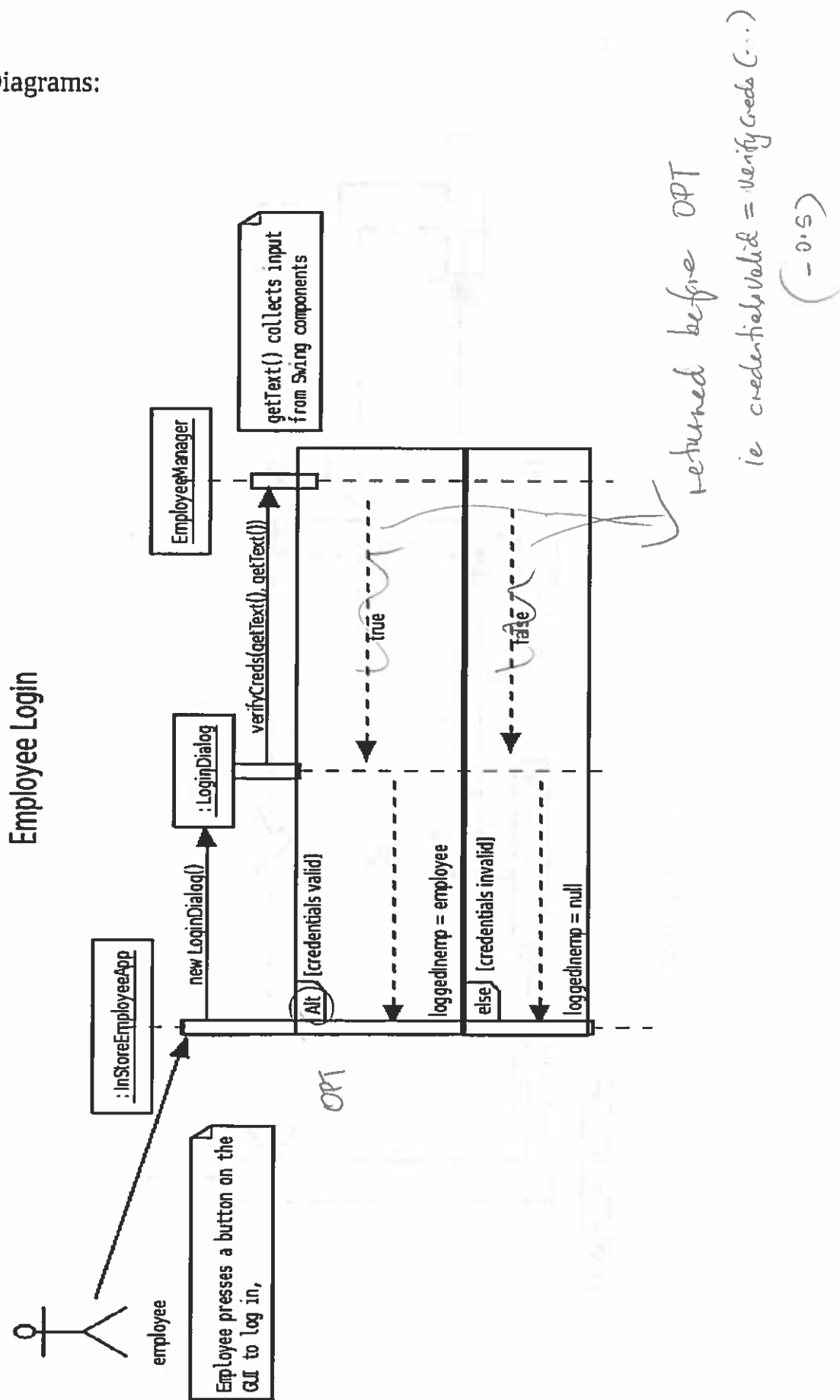
CS-414 Group I
Bryce Myers
Sky Egelhoff
Troy Cogburn
Nate Gillard

Updated Domain
Model, 10/22/1
Authorized by
Nate Gillard

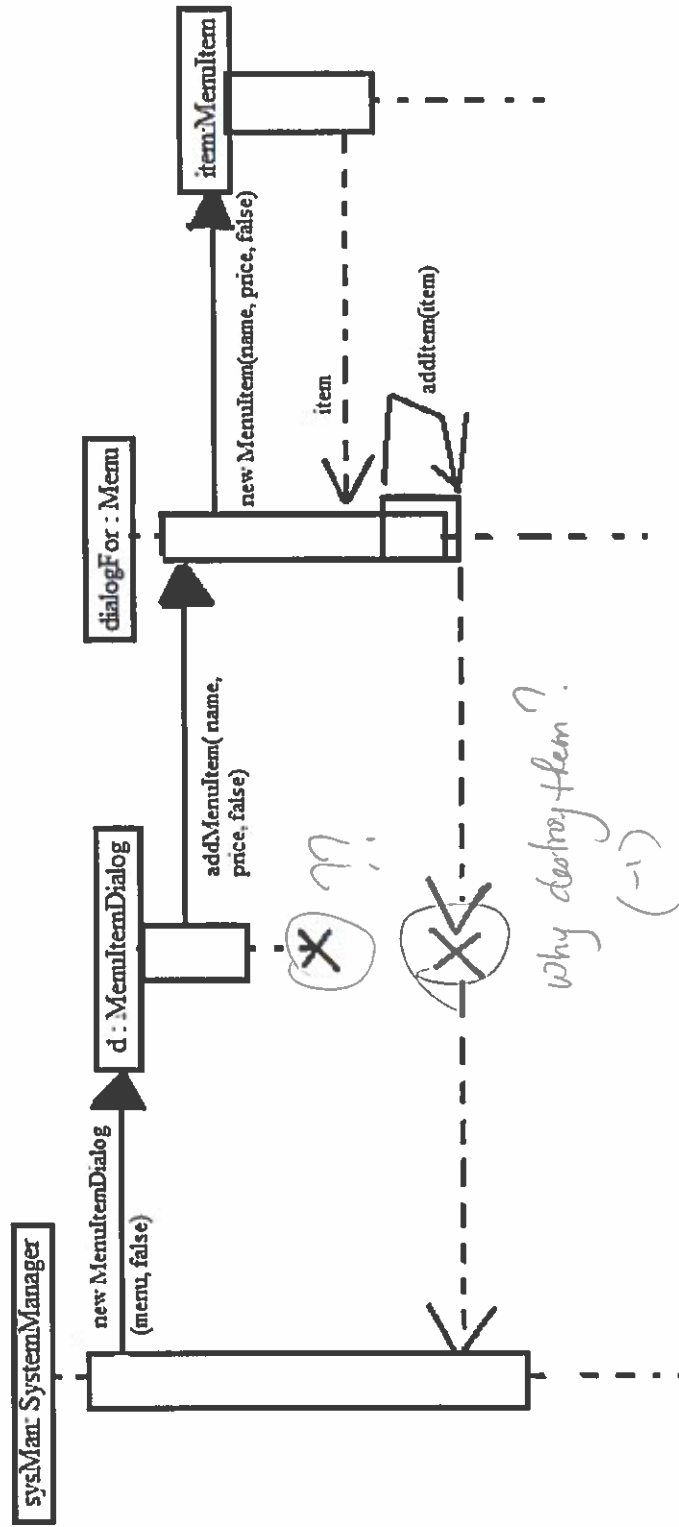
Design Class Model:



Sequence Diagrams:

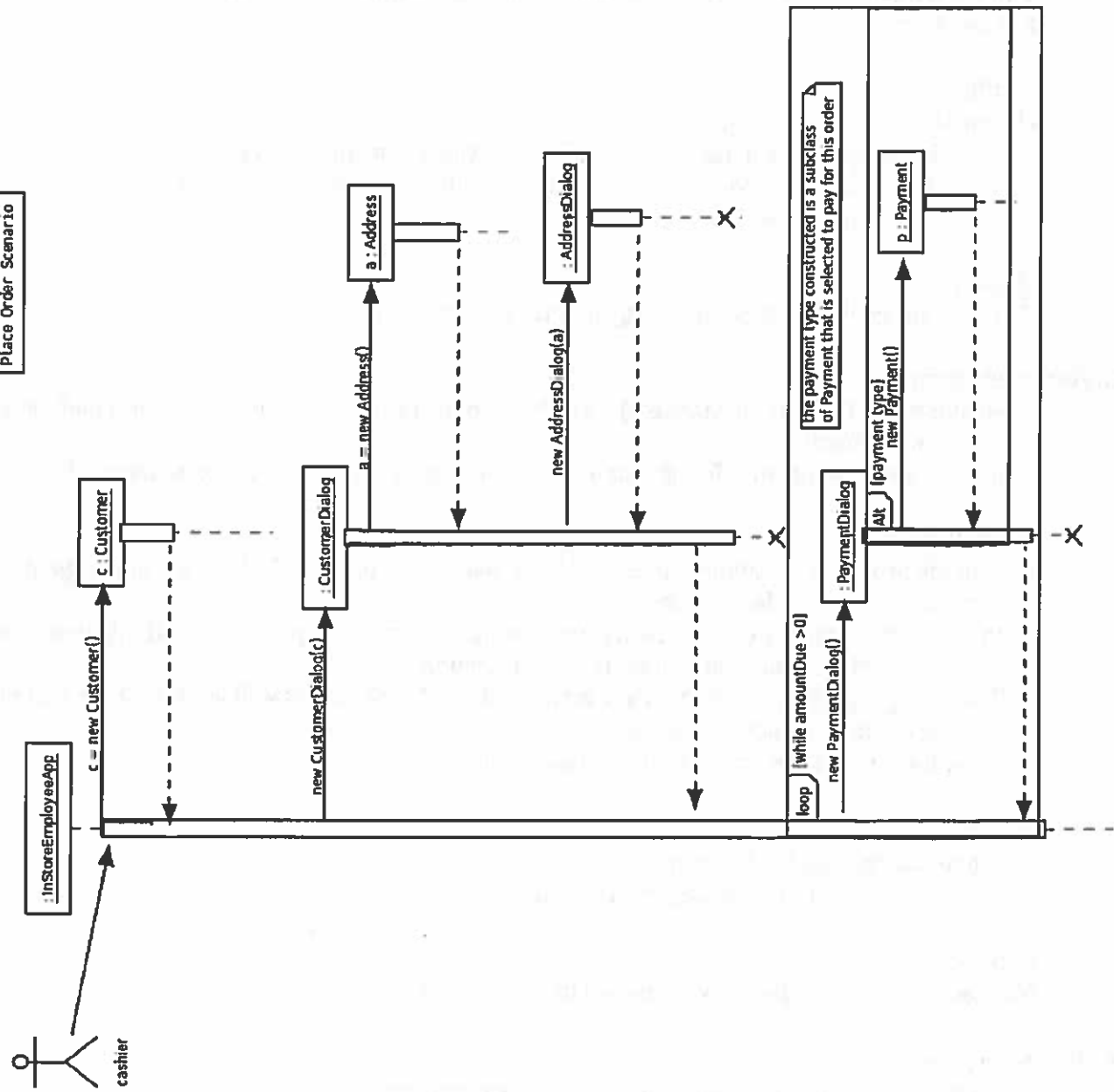


Add Menu Item Scenario



User pressed a button on the GUI to place a new order

Place Order Scenario



Where do you add the items to the order?
(-1)

Test Plan:

1. Login Employee:

- Tests the EmployeeManager module.
- The system must allow employees to login to the system.

Instructions:

1. Run the InStoreEmployeeApp by running the provided .jar (resize the window if needed).
2. Press the Login Employee button.
3. In the dialog that appears enter '003' as ID and 'password' as Password.
4. Press Accept.

Result:

The employee has been logged in.

- The employee's name, Jo Smith, is displayed as the logged in employee.
- Since this employee is a manager more buttons are available (Add/Edit Menu Item, View Orders).

Cleanup:

No clean up required, please stay logged in for next test case.

2. Add Menu Item:

- Tests the Menu (MenuItemManager) module, by performing an add item scenario and view menu scenario.
- If the restaurant wants to offer new items there must be a way to add offers to the menu.

Instruction:

1. Run the provided .jar and login as user '003' password 'password' if not already logged in.
2. Press the Add menu Item button.
3. In dialog that appears create a name for your item, then give a price, select if this item is a daily special, and finally press the accept button.
4. The dialog will exit, press the View menu button. A new dialog will appear where you will see the item you created listed.
5. Press the back button to return to the main menu.

Result:

The item has been added to the menu.

- The item can be viewed on the menu.

how will specials be identified on the menu?

Cleanup:

No clean up required, please stay logged in for the next test case.

3. Place New Order

- tests the OrderManager module, and Place order scenario.

Instruction:

1. Run the provided .jar and login as user '003' password 'password' if not already logged in.
2. Press the "Place Order" button.

3. A new dialog will appear for entering the customers address. Please fill these fields with an address real or imaginary, (eg: Street: 123 Sesame street, City: New York, State: NY, Zip: 08665, Phone: 908-211-4456) then press accept.
4. A new dialog will appear for the customer's name. Please enter a name (eg Steve MillerBand), then press accept
5. A new dialog will appear. This dialog is for adding items to the order. The top left cell contains the menu. Select items with the mouse (one item at a time), and press the 'Add selected items to order' button. (eg add a Pepperoni Pizza and Breadsticks(6)).
6. Once the desired items are added to the order press the 'accept' button.
7. A new Dialog will appear for adding payments to the order. Note the Amount due displayed in this dialog.
8. Press the button for cash and enter the amount 10.00 in the new dialogue's text field labeled amount. Then press accept.
9. Press the button for credit payment and enter the amount 3.99 and press accept.
10. A new dialogue will appear for entering the credit payment information Fill this information with the name provided in step 4 (or any String). Give a card number (any string), Card Security Code (any string) and Card Expiration date (any string). Press accpet.
11. The program will return to the main screen. Select view orders to see the order has been created.

Results:

The item may now be viewed in the View Order screen.

The total was calculated correctly and payments were applied correctly (minor issues with rounding doubles)

Cleanup:

No clean up required, please stay logged in for the next test case.

Insufficient system tests 3/4

Commit log:

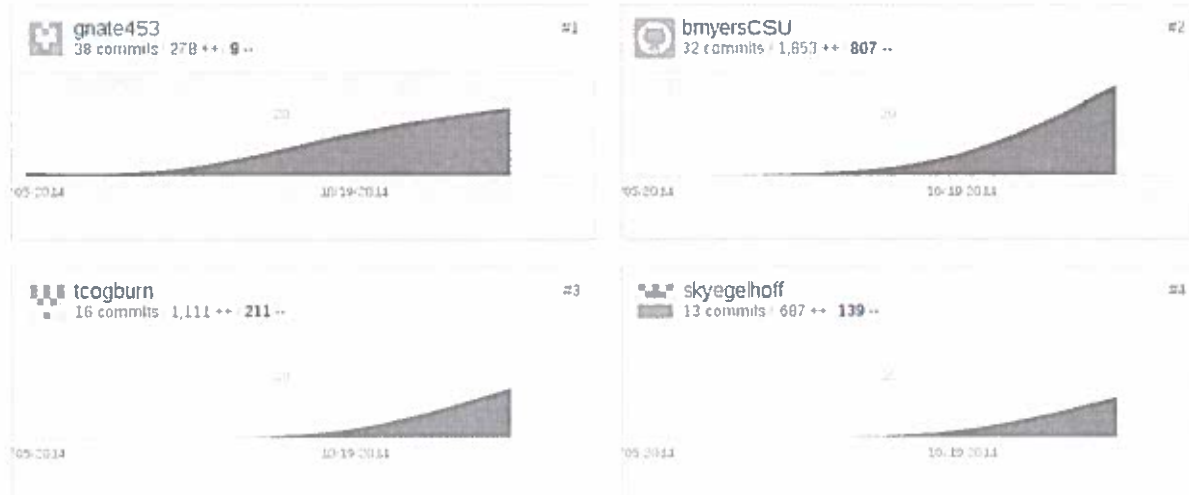
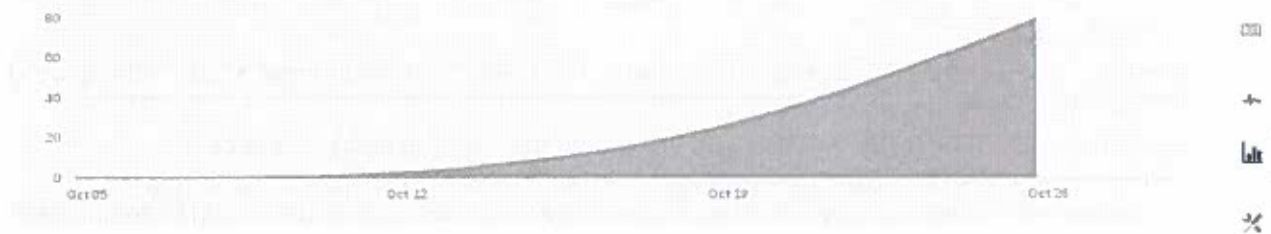
gnate453 / CS414h-Pizza_Ordering_System PRIVATE Unwatch 4 ★ Star 0 Fork 0

Contributors Traffic Commits Code frequency Punch card Network Members

Oct 5, 2014 – Oct 29, 2014

Contributions to master, excluding merge commits

Contributions Commits



commit 6be78f8b25233841e739dfcd128927338fe7b027

Author: tcogburn <tcogburn@rams.colostate.edu>

Date: Wed Oct 29 19:17:12 2014 -0600

Added CashBack and Also Changed accept to finish

commit b8cc8c08b75879d670408265e2db960ee445dc49

Merge: 61ca7f9 aebe091

Author: tcogburn <tcogburn@rams.colostate.edu>

Date: Wed Oct 29 18:53:59 2014 -0600

Merge branch 'master' of git@github.com:gnate453/CS414h-Pizza_Ordering_System.git

commit 61ca7f996e6d5a519a5595e2771962c5002c1e2f