**Analysis and Design Document**

Version 3.1

12/13/2013

*Restaurant Management System*

Object Oriented Software Engineering

SE 6301

***Prabhakar Ganesamurthy***

***Ajit Gogul Balakrishnan Kannan***

***Harika Jammula***

***Narla Sandeep Narla Saibaba***

***Sai Madhu Saranya Kachireddy***

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Inception draft | 11/4/2013 | First draft. To be refined primarily during elaboration. | … |
| Elaboration 1 Draft 1 | 11/16/2013 | First draft. To be refined primarily later | … |
| Elaboration 1 Draft 2 | 12/3/2013 | Second draft. To be refined primarily later | … |
| Elaboration 1 Draft 3 | 12/13/2013 | Revised draft. |  |

## Table of Contents

1. Use Case Diagram …4

2. Use Case Descriptions ...8

3. Domain Model ...17

4. System Sequence Diagram ...20

5. Operation Contracts ...24

6. Subsystem Diagram ...35

7. Use case Realization …38

8. Design Class Diagram …42

9. Glossary of terms …44

10. Screenshots …46

***1. Use Case Diagrams***

Version 2.1

12/3/2013

*Restaurant Management System*

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Inception draft | 11/16/2013 | Inception draft. To be refined primarily later | … |
| Elaboration 1 Draft 1 | 11/16/2013 | First draft. To be refined primarily later | … |
| Elaboration 2 Draft 1 | 12/3/2013 | Second draft. To be refined primarily later | … |
|  |  |  |  |

**Use Case Diagram - Restaurant Management System**



**Use Case Prioritization:**

1. Manage order
2. Process order
3. Manage table status
4. Handle payment
5. Assign waiter to table
6. Manage restaurant information

|  |  |  |
| --- | --- | --- |
| **Fully Dressed** | **Casual** | **Brief** |
| 1.Manage order | 3.Process order | 7.Manage restaurant information |
| 2.Login | 4.Manage table status |  |
|  | 5.Handle payment |  |
|  | 6.Assign waiter to table |  |
|  |  |  |
|  |  |  |
|  |  |  |

***2. Use Case Descriptions***

Version 2.0

11/16/2013

*Restaurant Management System*

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| 1.0 Inception draft | 11/16/2013 | Refined |  |
| 2.0 Elaboration 1 draft 1 | 11/16/2013 | First draft. To be refined primarily later |  |
| 3.0 Elaboration 2 draft 1 | 12/13/2013 | Refined draft. |  |

**Brief Use case descriptions:**

**1.Manage order:**

The waiter selects the table from the floor status screen and adds the item ordered by the user to the tables tab which gets added to the order queue which is viewed by the kitchen staff.

The waiter can cancel/modify the order. The waiter can check the status of the order whether it is ready or not. If the order status is ready waiter serves the order.

**2.Login:**

The user enters his/her credentials and they are welcomed by a welcome page which is user-role specific.

The welcome page is for:

Waiter : Floor status screen

Kitchen staff: Order queue

Busboy : Floor status screen

Host : Floor status screen

Manager : Restaurant management screen

**3.Process order:**

Kitchen staff selects order from the order queue in a first-come-first-serve basis and updates order status as 'Processing' when the order is being prepared.

Once the order is ready to be served, the kitchen staff updates the status of the order as ready.

**4.Manage table status:**

The host views table status screen and identifies available tables to which customers are assigned.

After the customer sits in an available table ,the waiter selects that table and updates its status as occupied(yellow).

After the order in a table is closed, the waiter assigned to that table, selects that table and updates its status as dirty(red).

The busboy identifies dirty tables from the table status screen.

After cleaning the table marked as dirty(red), Busboy selects that table and updates its status as available(green).

**5.Handle Payment:**

The waiter selects the mode of payment for the order. If the mode of payment is by credit card, then the waiter swipes the card in the card reader, the card is validated and payment is processed. if the customer pays by cash, the waiter registers the payment as 'by cash ' in the system and completes the payment.

**6.Assign waiter to a table:**

The host assigns a waiter to a set of tables. The waiter is responsible for this set of tables.

**7. Manage Restaurant Information**

Time triggers the system to store data in the database about employees, menu items, customers and their orders in day-by-day and hourly intervals. This data is used by the manager for analysis.

The manager can track the status of the various inventories available in the restaurant, analyze the sales of items such as identifying the most popular item, revenue generated by each item etc., tracks employee’s activities such as number of hours worked, performance by accessing the database.

The manager can also authorizes the activities which are restricted to the waiter by the selecting that waiter by ID and authorizing him/her.

The manager alters the floor plan by adding, removing and moving tables in the system. The manager can create/ modify employee profile in the system.

**Fully-dressed format:**

|  |
| --- |
| **Use Case: Manage order** |
| **ID:** UC-1 |
| **Brief description:** The waiter selects the table from the floor status screen and adds the item ordered by the user to the order which gets added to the order queue when order is placed. Waiter can cancel/modify/close order. Waiter can also know when order is ready by checking the order status for 'Ready'. |
| **Primary actors:** Wait Staff |
| **Secondary actors**: Kitchen staff |
| **Preconditions:** Update table status to occupied |
| **Main flow:**  1. Include : Login.  2. Select table from the floor status screen.  3. System provides waiter with an electronic menu  4. For each item ordered by the user  4.1 The waiter selects the item from electronic menu and adds it to the order  5. After all ordered items have been added, the waiter places the order.  6. The order gets added to the order queue  7. System notifies the kitchen staff that a new order has been placed in the order queue.  8.If order needs to be modified  8.1 Select order  8.2 if order status != 'Processing'  8.2.1 Add/Remove items in the order  8.2.2 Update order  else  8.2.1 Display 'Order cannot be modified'  9.If order needs to be cancelled  9.1 Select order  9.2 if order status != 'Processing'  9.2.1 Cancel order  9.2.2 Update order status as 'Cancelled'.  9.2.3 Order is removed from the queue.  else  9.2.1 Display order cannot be cancelled. |
| **Post conditions:**  If order is placed/modified:   1. The order is closed when order is served, payment for the order is complete and customer leaves the table.   If order is cancelled:  1.Order is removed from the order queue if it is not processed. |
| **Non-Functional requirements:**   * The waiter needs to place the order in minimal time, so the electronic menu should be organized and quick to use * The order placed must be placed in the order queue within 1 second. * Order modifications/cancellation should be reflected in the order queue within 1 second. |
| **Technology and Data Variation List:** None |
| **Open Issues:** None |

|  |
| --- |
| **Use case: Login** |
| **ID:** UC-2 |
| **Brief description:** The user enters the login credentials to access their profile. |
| **Primary actors:** Kitchen staff, Waiter, Bus-boy, Manager, Host |
| **Secondary actors:** None |
| **Preconditions:** Access to a terminal which has access to the system network |
| **Main flow:**  1. User selects the login button on the screen  2. System prompts for username and password  3. The user enters the required information  4. The system validates the input of the user and the appropriate user interface is displayed. Waiter : Floor status screen  Kitchen staff: Order queue  Busboy : Floor status screen  Host : Floor status screen  Manager : Restaurant management screen |
| **Post conditions:** The user is successfully logged into the system and is greeted by a welcome screen. |
| **Non-Functional requirements:** Login authentication should take not more than 3 seconds. |
| **Technology and Data Variation List:** None |
| **Open Issues:** None |

***3.Domain Model***

Version 2.1

12/10/2013

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Elaboration 1 Draft 1 | 11/16/2013 | First draft. To be refined primarily later | … |
| Elaboration 2 Draft 1 | 12/10/2013 | Refined draft | … |
|  |  |  |  |

**Restaurant Management System:**

****

***4.System Sequence Diagram***

Version 2.0

12/3/2013

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Elaboration 1 Draft 1 | 11/16/2013 | First draft. To be refined primarily later | … |
| Elaboration 2 Draft 1 | 12/3/2013 | Refined draft | … |
| Elaboration 3 Draft 1 | 12/13/2013 | Revised draft |  |

**ManageOrder:**

****

**Login:**



***5.Operation Contracts***

Version 1

11/16/2013

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Elaboration 1 Draft 1 | 11/16/2013 | First draft. To be refined primarily later | … |
| Elaboration 2 Draft 1 | 12/3/2013 | Refined draft | … |
|  |  |  |  |

|  |
| --- |
| **Contract CO1**:Enter credentials |
| **Operation**: Enter Credentials() |
| **Cross references**:login |
| **Pre conditions:**The user should be an employee of the restaurant |
| **Post Condition:**The user credentials should be received by the system |

|  |
| --- |
| **Contract CO2**: Authenticate |
| **Operation**:Authenticate() |
| **Cross references**: login |
| **Pre conditions:** The user should enter the credentials |
| **Post Condition:** The authentication is acknowledged |

|  |
| --- |
| **Contract CO3**:Floor Status Screen |
| **Operation**: Floor Status Screen() |
| **Cross references**: login |
| **Pre conditions:** The employee should be a waiter or Busboy or Host and authenticated |
| **Post Condition:** Floor status screen is returned to the employee |

|  |
| --- |
| **Contract CO4:**Restaurant Management Screen |
| **Operation:** Restaurant Management Screen() |
| **Cross references**: login |
| **Pre conditions:** Employee should be a manager and authenticated |
| **Post condition:**Restaurant management screen is returned to the employee |

|  |
| --- |
| **Contract CO5:**Order Queue Screen |
| **Operation:** Order Queue Screen() |
| **Cross references**:login |
| **Pre conditions:** The employee should be a kitchen staff and authenticated |
| **Post condition:**Order queue screen should be returned |

|  |
| --- |
| **Contract CO6:** Authentication failed |
| **Operation:** Authentication failed() |
| **Cross references**:login |
| **Pre conditions:** Employee credentials should be incorrect |
| **Post condition:** Authentication failure is acknowledged |

|  |
| --- |
| **Contract CO7**: Select table |
| **Operation**: Select table() |
| **Cross references**: Manage order, Update table status, View table status, Assign waiter to table |
| **Pre conditions:** User must be logged in |
| **Post Condition:** The desired table object has been selected |

|  |
| --- |
| **Contract CO8:** Table selected |
| **Operation:** Display table status() |
| **Cross references**: View table status |
| **Pre conditions:** The desired table object has been selected |
| **Post condition:** The status attribute of the selected table object is displayed |

|  |
| --- |
| **Contract CO9:** Menu |
| **Operation**: Display menu() |
| **Cross references:** Manage order |
| **Pre conditions**: The status attribute of the selected table object is changed to ‘occupied’. |
| **Post conditions:** The menu object is invoked. |

|  |
| --- |
| **Contract CO10:** Select category |
| **Operation**: Select category () |
| **Cross reference:** Manage order |
| **Pre-condition:** The menu object is invoked. |
| **Post-condition:** The desired category attribute of the menu object is selected. |

|  |
| --- |
| **Contract CO11**: Category items list |
| **Operation:** Display category () |
| **Cross reference:** Manage order |
| **Pre-condition:** The desired category attribute of the menu object is selected. |
| **Post-condition:** The list of items belonging to the selected category attribute is displayed |

|  |
| --- |
| **Contract CO12:** Select item |
| **Operation:** Select item() |
| **Cross reference:** Manage order |
| **Pre-condition:** The list of items belonging to the selected category attribute is displayed |
| **Post-condition:** The desired item object is invoked. |

|  |
| --- |
| **Contract CO13:** Item added to order |
| **Operation:** Update order() |
| **Cross reference:** Manage order |
| **Pre-condition:** The desired item object is invoked |
| **Post-condition:** A order instance o is created. Attributes of o are initialized. |

|  |
| --- |
| **Contract CO14:** Place order |
| **Operation:** Place order() |
| **Cross reference:** Manage order |
| **Pre-condition:** A order instance o is created.  Attributes of o are initialized. |
| **Post-condition:** o is associated with kitchen staff  The attributes of order object are initialized. |

|  |
| --- |
| **Contract CO15:** Notify |
| **Operation:** Notify() |
| **Cross reference:** Manage order |
| **Pre-condition:** o is associated with kitchen staff  The attributes of order object are initialized. |
| **Post-condition:** The pending order attribute is set to true. |

|  |
| --- |
| **Contract CO16:** Update order in database |
| **Operation:** Update order DB() |
| **Cross reference:** Process order |
| **Pre-condition:** The pending order attribute is set to false |
| **Post-condition:** A order\_DB instance odb is created  Attributes of odb are initialized  odb is associated with kitchen staff |

|  |
| --- |
| **Contract CO17:** Order placed |
| **Operation:** Order placed() |
| **Cross reference:** Process order |
| **Pre-condition:** A order\_DB instance odb is created  Attributes of odb are initialized  odb is associated with kitchen staff |
| **Post-condition**: The order status attribute of o is updated as ‘ready’ |

|  |
| --- |
| **Contract C018:** Select order |
| **Operation**: Select order |
| **Cross reference:** Manage order |
| **Pre-condition**: The user should login  The orderID attribute of o exists |
| **Post-condition:** The desired orderID attribute is selected |

|  |
| --- |
| **Contract C019:** Order details |
| **Operation:** Display order() |
| **Cross reference**: Manage order |
| **Pre-condition:** The desired orderID attribute is selected |
| **Post-condition:** The attributes of o are displayed |

|  |
| --- |
| **Contract C020:** Add Item |
| **Operation**: Add item() |
| **Cross reference**: Manage order |
| **Pre-condition:** The desired orderID attribute is selected  The attributes of o are displayed |
| **Post-condition**: o is associated with the kitchen staff  The attributes of order object are updated. |

|  |
| --- |
| **Contract C021**: Add Item |
| **Operation:** Remove Item |
| **Cross reference**: Manage order |
| **Pre-condition:** The desired orderID attribute is selected |
| **Post condition:** The order item attribute of the selected orderID is removed  o is associated with the kitchen staff |

|  |
| --- |
| **Contract C022:** Add Item |
| **Operation:** Update order |
| **Cross reference**: Manage order |
| **Pre-condition**: The desired orderID attribute is selected  The attributes of o are displayed |
| **Post condition:** The order item attributes of the selected orderID is updated  o is associated with the kitchen staff |

|  |
| --- |
| **Contract C023**: Cancel order |
| **Operation**: Cancel order() |
| **Cross reference:** ManageOrder |
| **Pre-condition**: The desired orderID attribute is selected and order status attribute is not processed |
| **Post condition** The order object of the selected order ID is destroyed  o is associated with the kitchen staff |

|  |
| --- |
| **Contract C024**: Order cannot be modified |
| **Operation**: Order cannot be modified |
| **Cross reference**: Manage order |
| **Pre-condition**: The desired orderID is attribute is selected and order status attribute is processed |
| **Post condition**: None |

|  |
| --- |
| **Contract C025**: Order cannot be cancelled |
| **Operation**: Order cannot cancelled |
| **Cross reference**: Manage order |
| **Pre-condition**: The desired orderID is attribute is selected and order status attribute is processed |
| **Post condition**: None |

***6.Sub-system Diagram***

Version 2

12/3/2013

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Elaboration 1 Draft 1 | 11/16/2013 | First draft. To be refined primarily later | … |
| Elaboration 2 Draft 1 | 12/3/2013 | Refined draft | … |
|  |  |  |  |

**Restaurant Automation System**

**Login\_Subsystem**

**Inventory\_Subsystem**

**UserInterface\_Subsystem**

**Waiter**

**Database\_Subsystem**

**OrderManagement\_Subsystem**

OrderDB

InventoryDB

EmployeeDB

**Manager**

**KitchenStaff**

**Host**

**Busboy**

TableStatus

screen

Login Screen

OrderQueue

Screen

Menu

OrderQueue

System

OrderStatus

System

AddEmployeeControl

RemoveEmployeeControl

PayrollControl

SelectItemControl

QueueControl

TakeOrder

Control

CompleteOrder

Control

ManageOrder

Control

TableStatusControl

TableStatusControl

Authentication

Control

InventoryStatistics

Control

***7.Use Case Realization***

Version 2

12/10/2013

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Elaboration 2 Draft 1 | 12/3/2013 | First draft. To be refined later. | … |
| Elaboration 3 Draft 1 | 12/10/2013 | Refined draft |  |

**Manage Order:**

****

**Login:**



***8.Design Class Diagram***

Version 1

12/11/2013

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Elaboration 2 Draft 1 | 12/3/2013 | First draft. To be refined later. | … |
| Elaboration 3 Draft 1 | 12/11/2013 | Refined |  |

**Design Class Diagram:**



**9.Glossary of terms**

**1. Employee – Any individual who is involved in the day to day activities of the**

**restaurant, including cooks, bus boys, kitchen staff, waiters, managers and hosts.**

**2. Manager – Type of employee with the highest administrative authority, with the power to change the status of all other employee types, and access to restaurant statistics and floor plan.**

**3. Customer – Any individual who is benefitted by the restaurant automation system and the employees working within its framework.**

**4. Kitchen Staff – Type of employee who is responsible for processing the orders submitted by the waiter in the order queue.**

**5. Waiter – Type of employee who is responsible to interact with the customer by creating, modifying and cancelling orders and to keep the order queue up to date**

**6. Host – Type of employee who welcomes the customer and directs them to an available table. The role of host can be performed by both manager and waiter.**

**7. Bus Boy – Type of employee who is responsible to keep the tables clean and to change the status of the table from dirty to clean upon completion.**

**8. Table – A physical object in which the prepared food is presented to the customer, for which a waiter is assigned.**

**9. Table Status – Shows the status of a table and the waiter who is assigned to that table.**

**10. Dirty – A type of table status indicating the condition which implies that it is not being currently occupied by any customers and needs to be cleaned to make it available to a new customer.**

**11. Busy – A type of table status indicating the condition where it cannot be assigned to a new customer and also that it cannot be cleaned.**

**12. Ready – A type of table status indicating the condition that it is ready for service and that it can be assigned to a new customer.**

**13. Order – A description submitted by the waiter, which is to be cooked and eventually**

**delivered to the table from which it has been ordered.**

**14. Order Status – Used to know the status of the order placed by the customer based on which the waiter decides what needs to be done.**

**15. Processing – A type of order status which means that the order is currently being cooked and the order cannot be modified after this.**

**16. Ready – A type of order status which means that the order has been cooked and ready to be served to the customer.**

**17. Payroll – Used for calculating the salary of an employee based on the number of hours worked. It**

**differs based on the designation of the employee.**

**18. Menu – List of dishes available in the restaurant from which the waiter chooses from based on the request from the customer.**

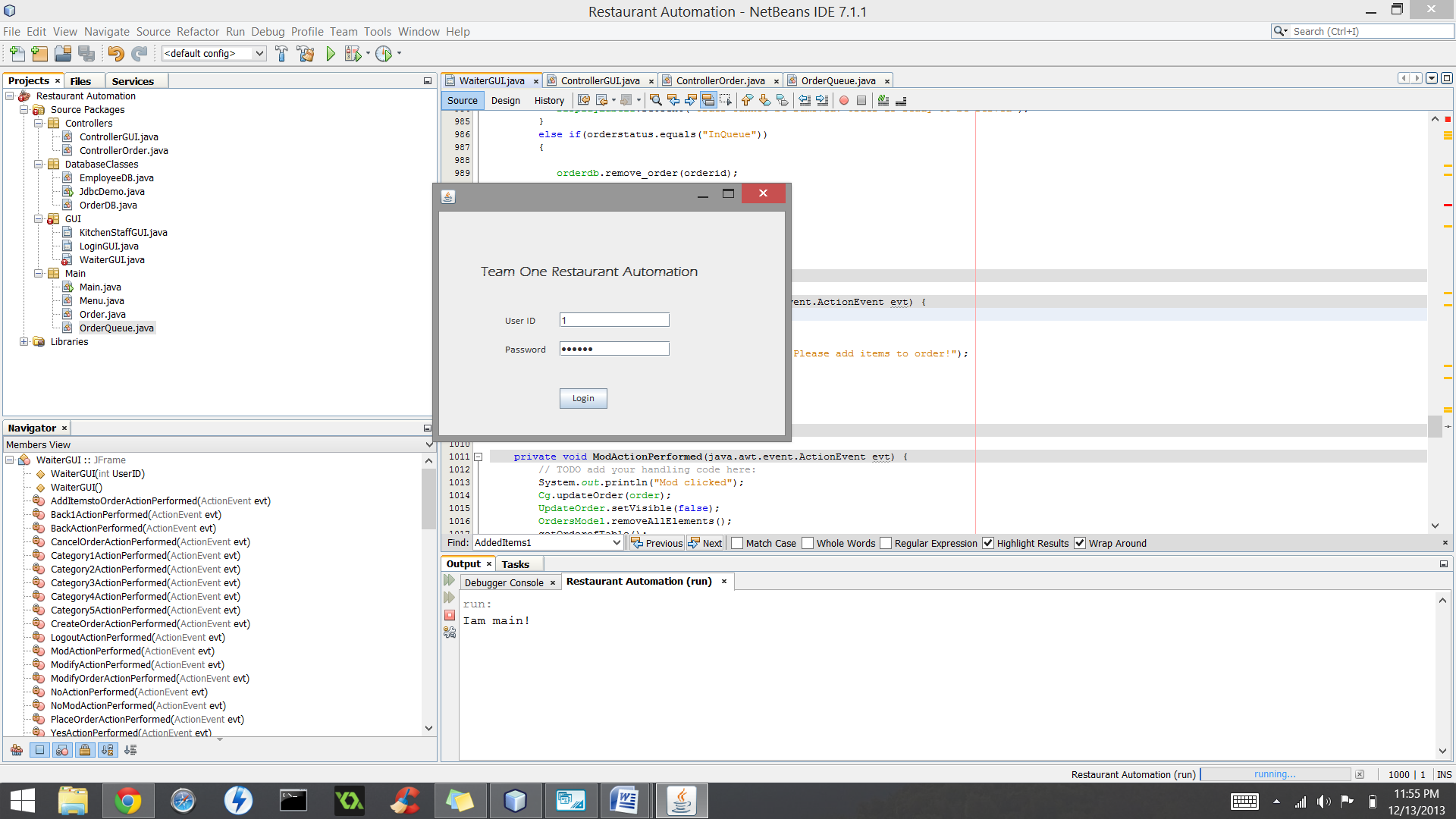
**19. Floor Plan – Shows the arrangement of all the tables in the restaurant which can be modified only by the manager when required.**

***10.Screenshots***

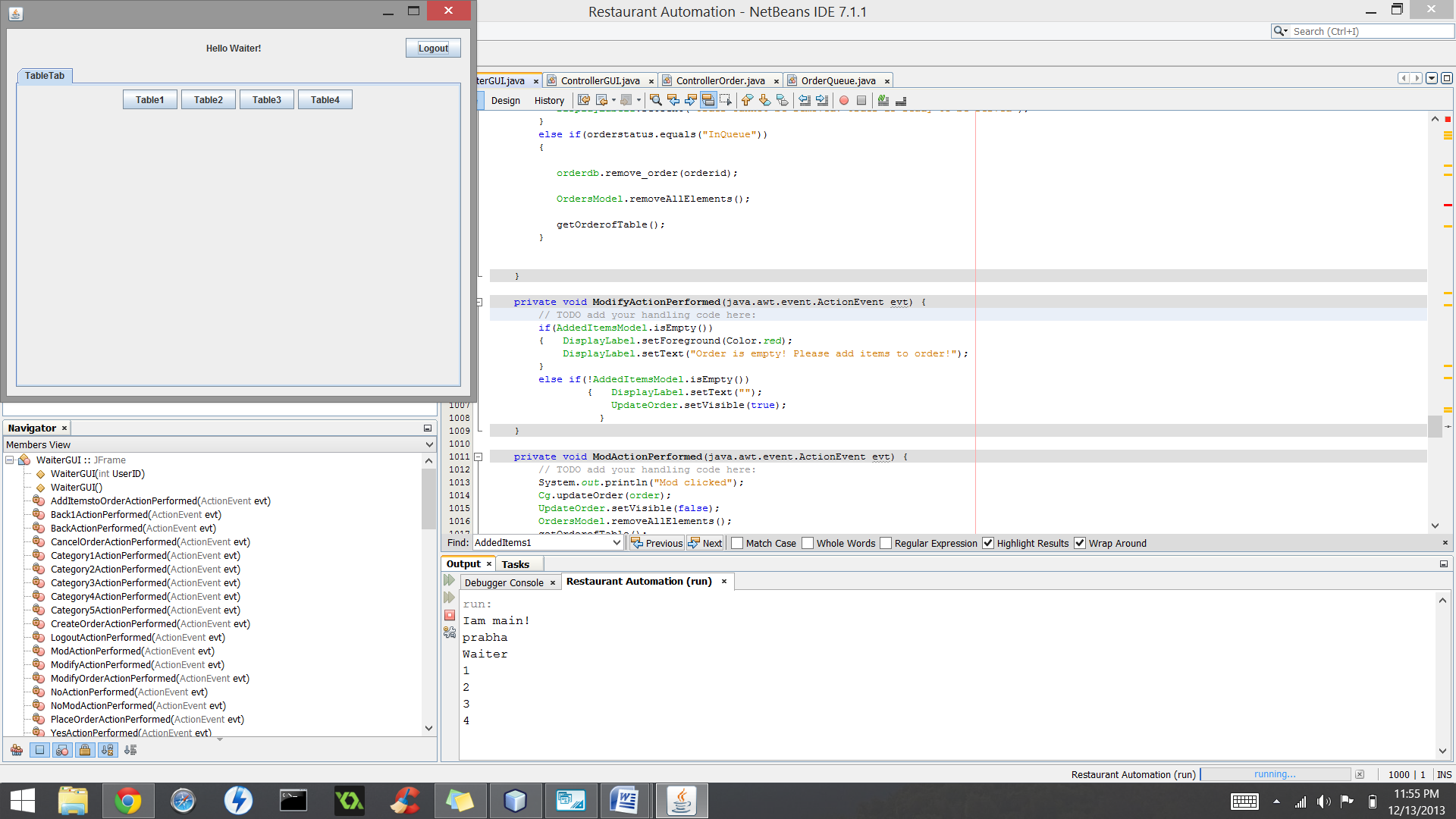
Version 1

12/13/2013

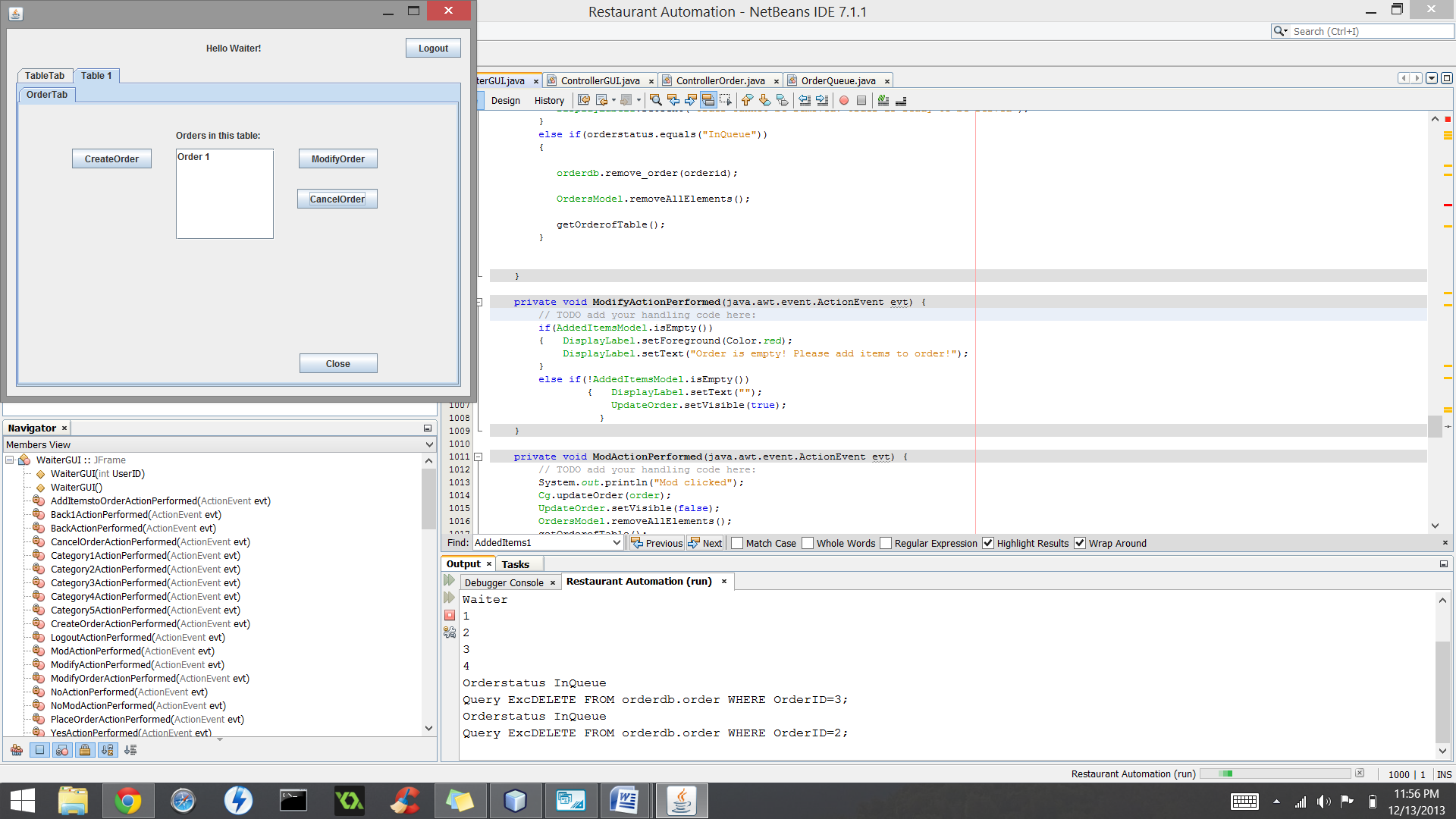
Login Screen:



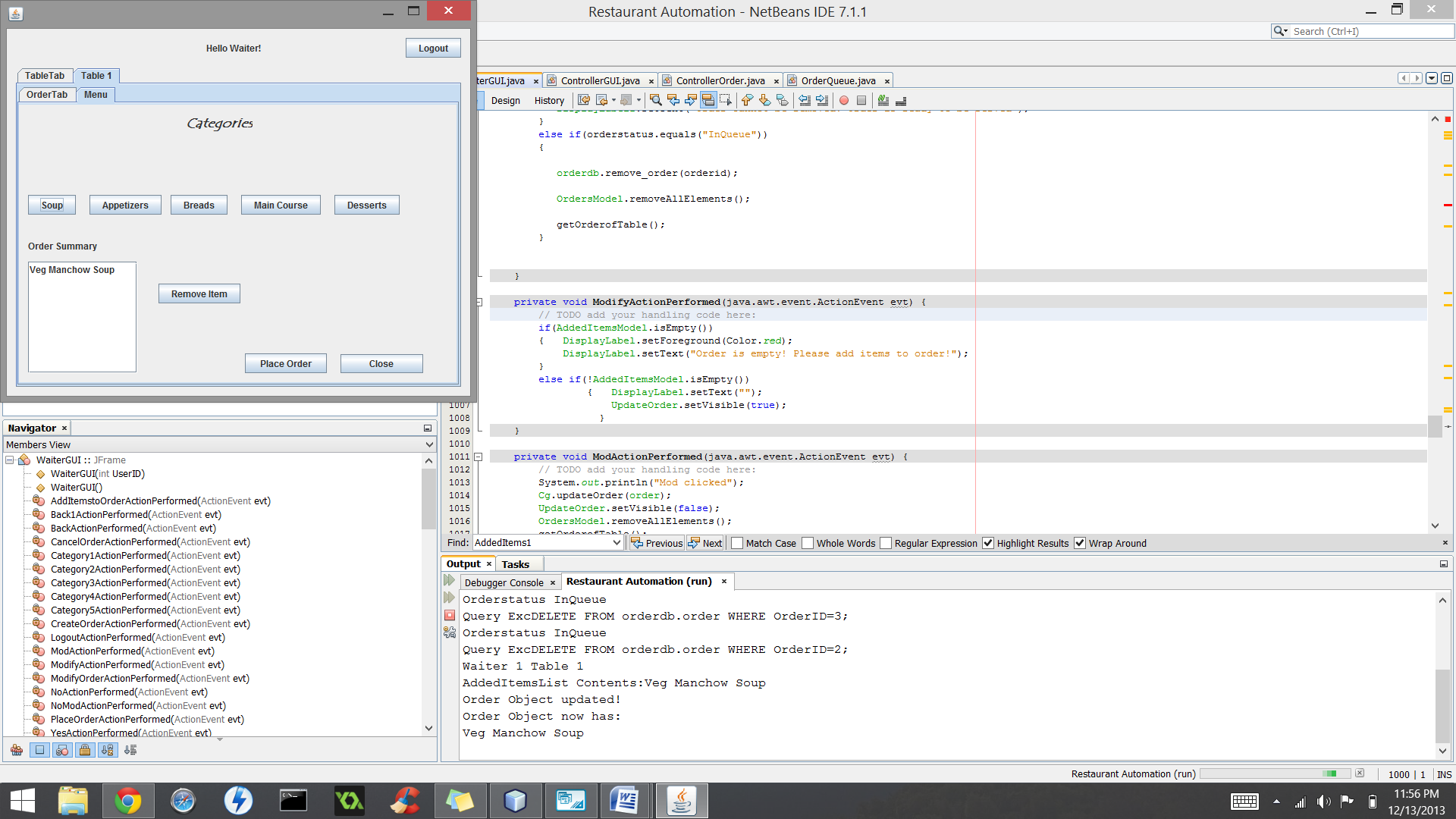
Waiter Table Screen:



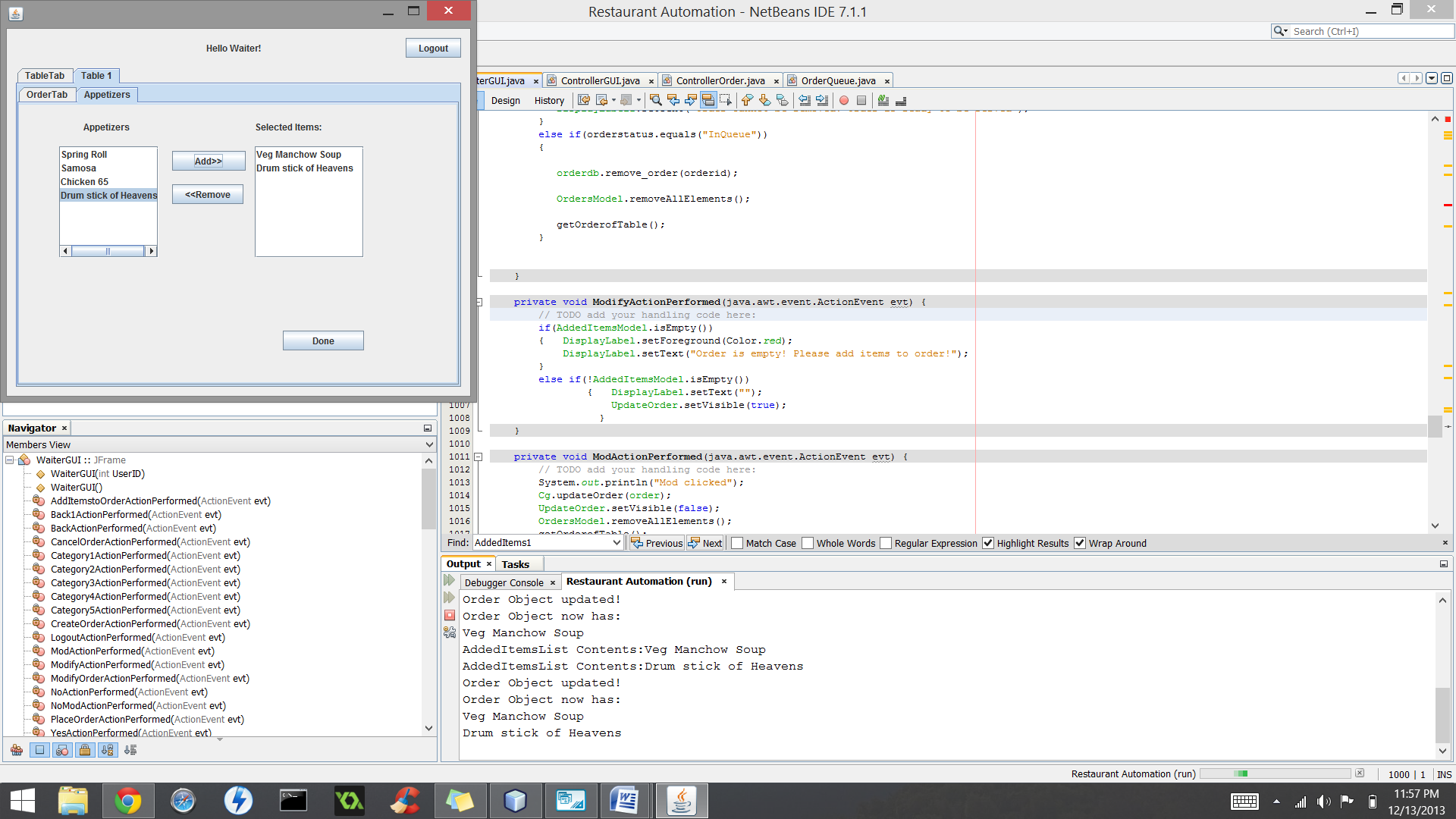
Waiter Table Tab:



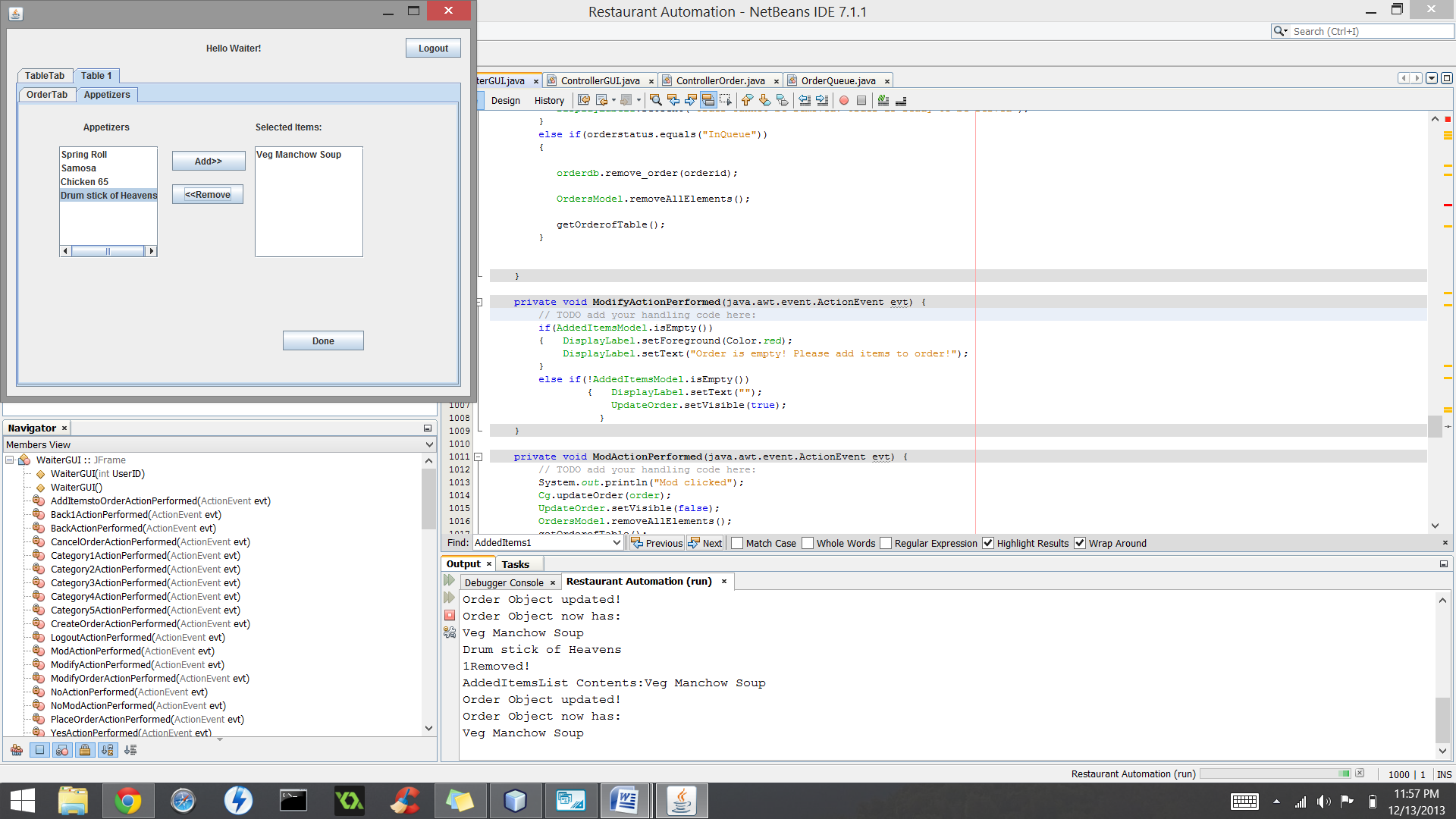
Create Order Screen:



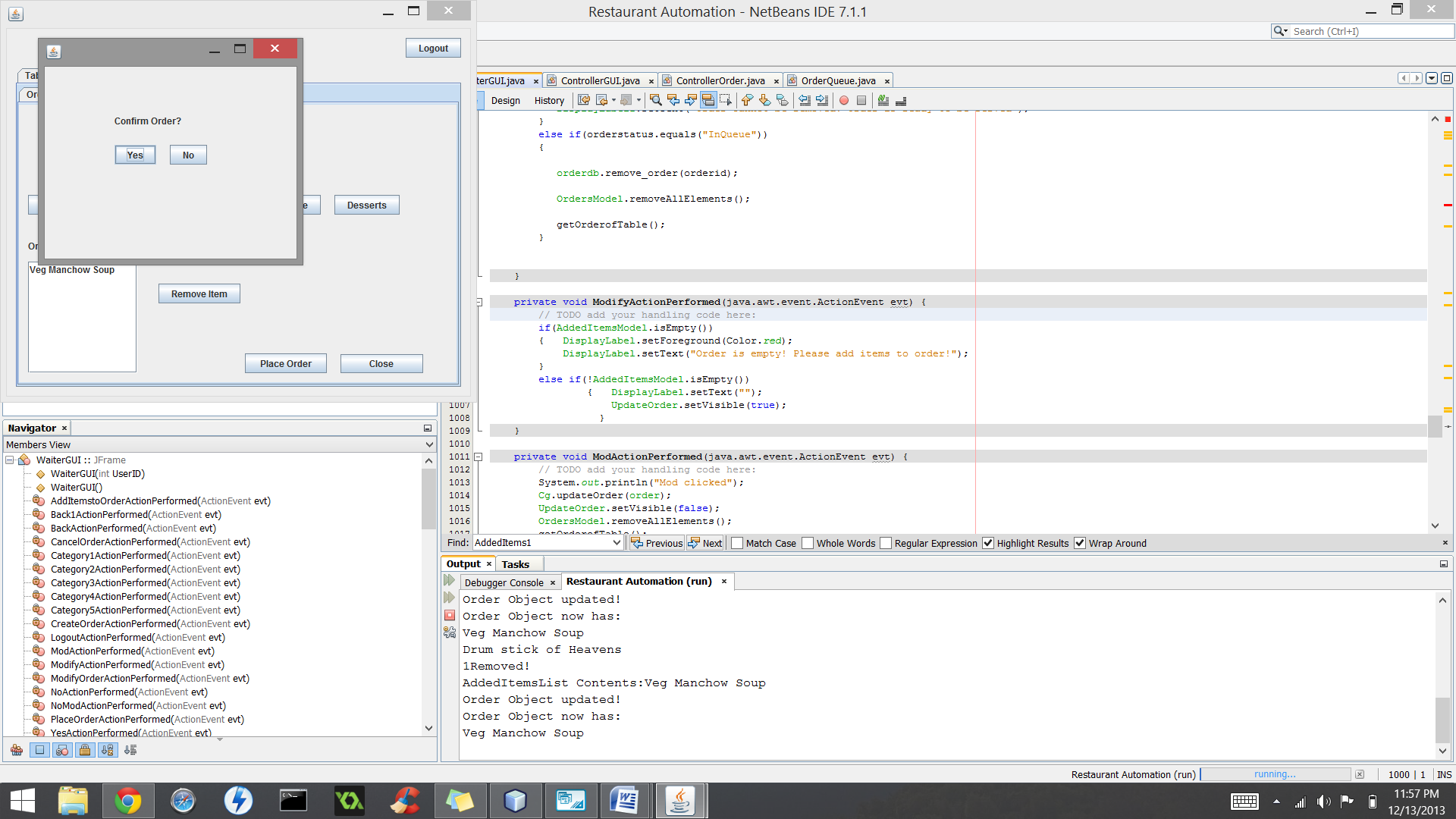
Adding Items to Order:



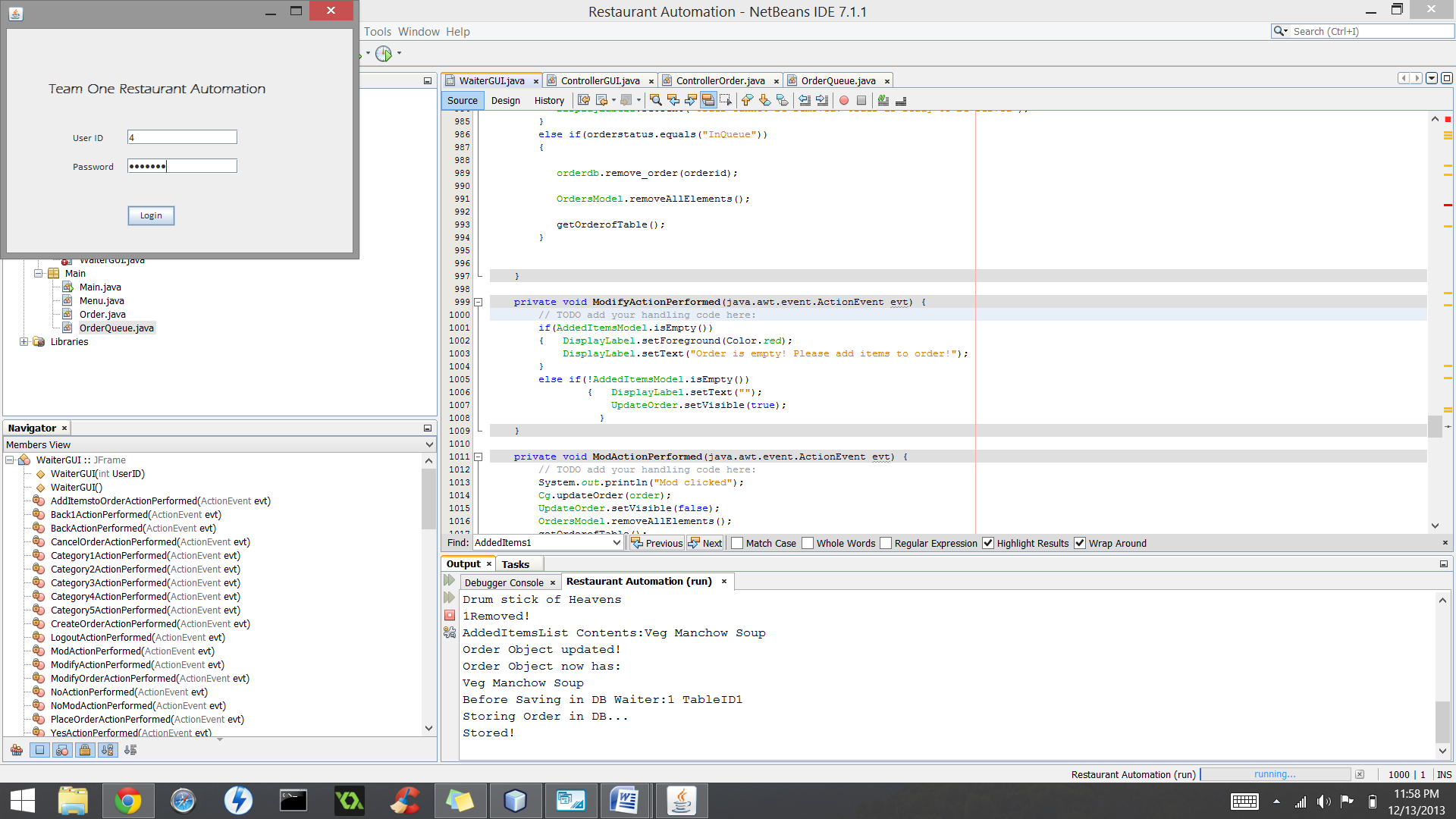
Removing Items to Order:



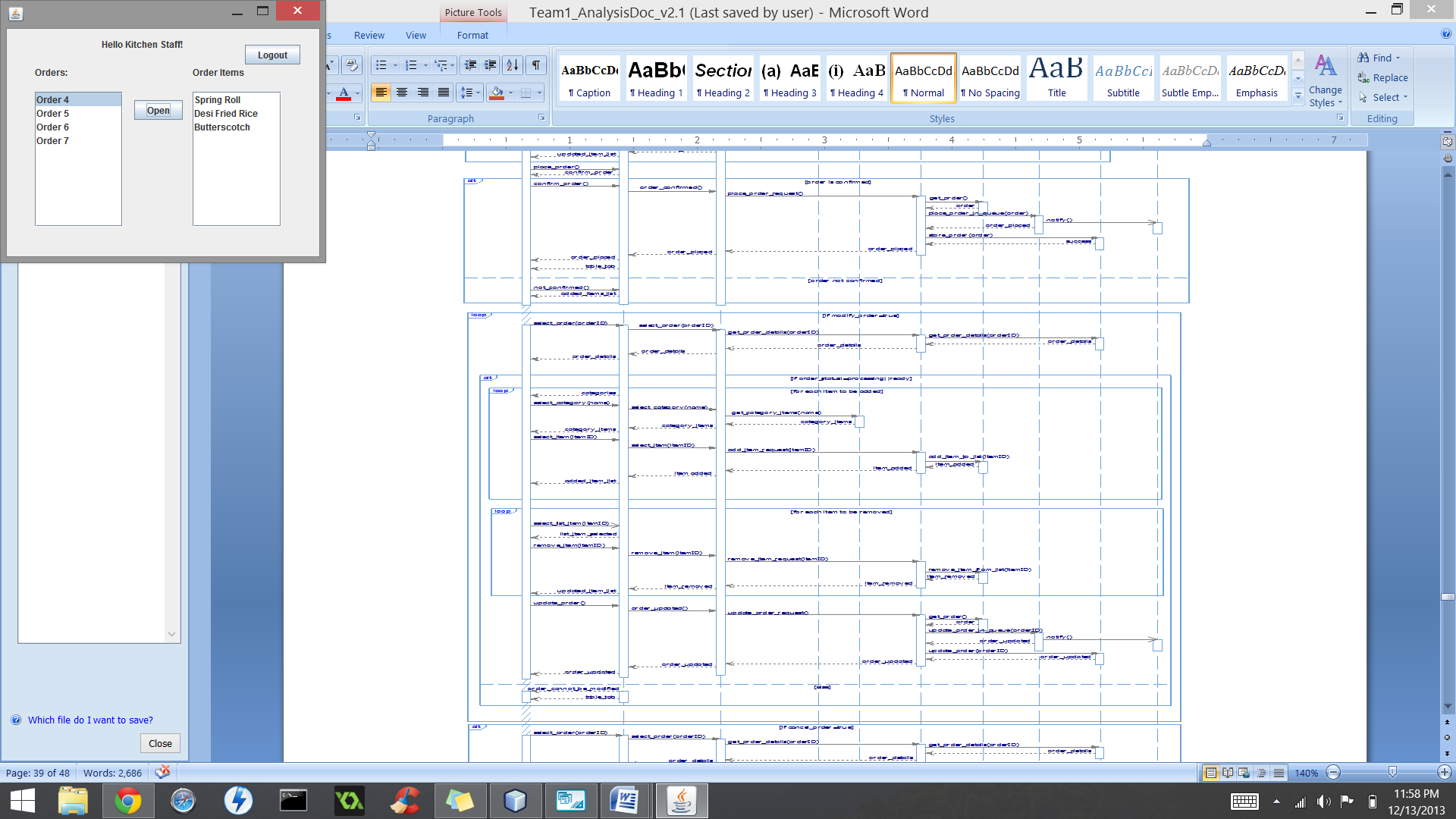
Place Order



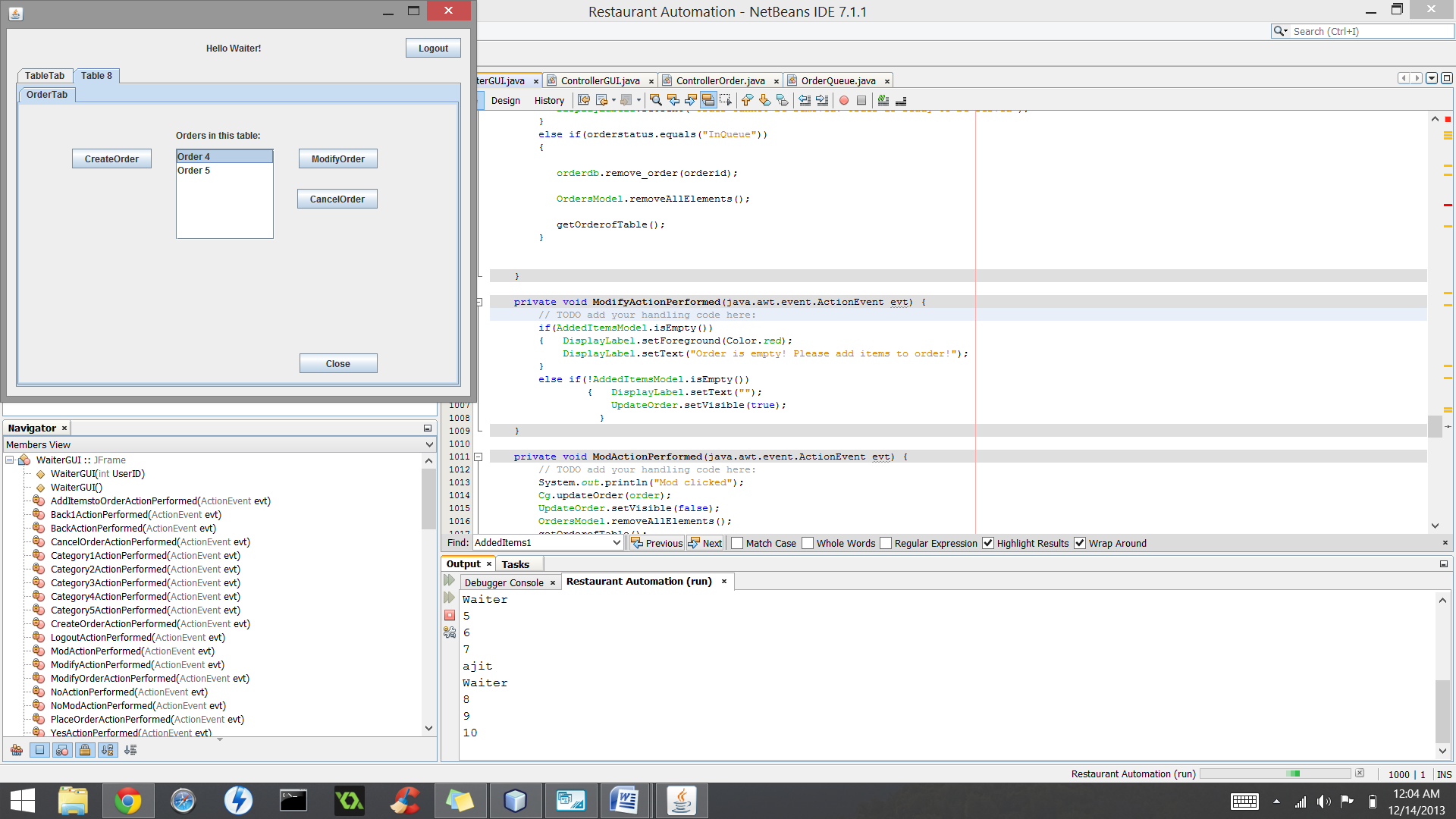
Waiter Login

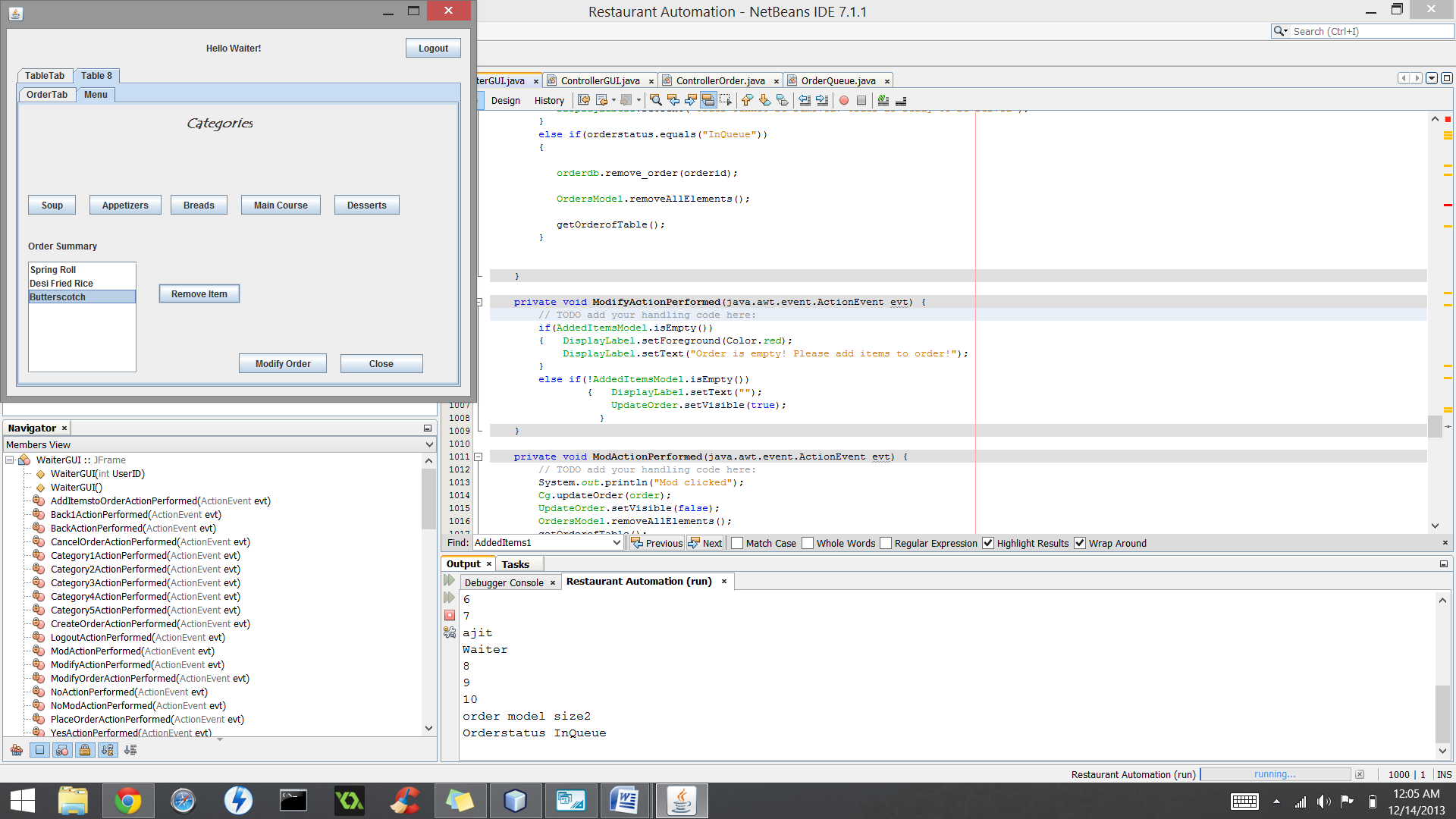


Kitchen Staff GUI with OrderQueue:

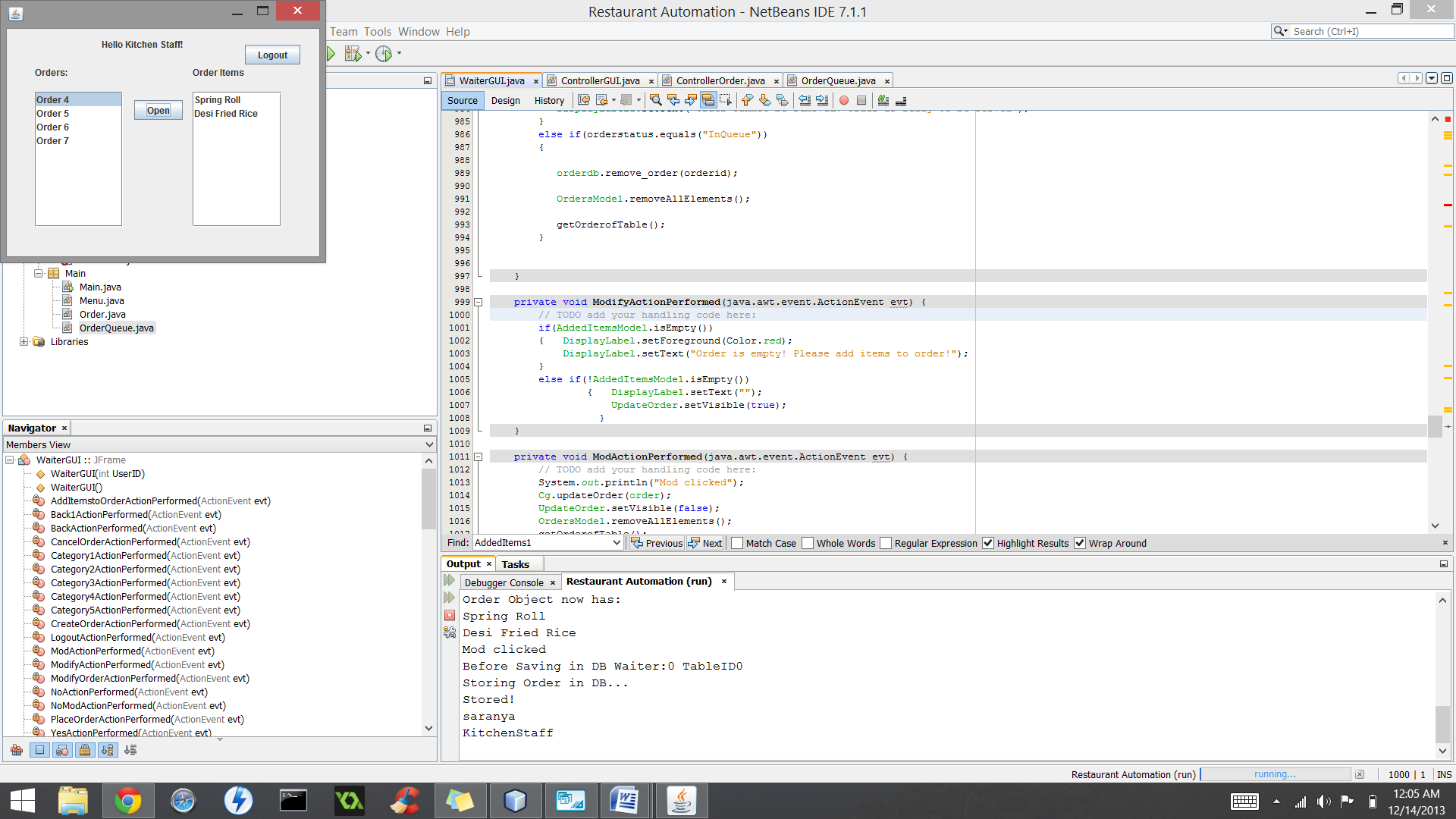


Modify Order

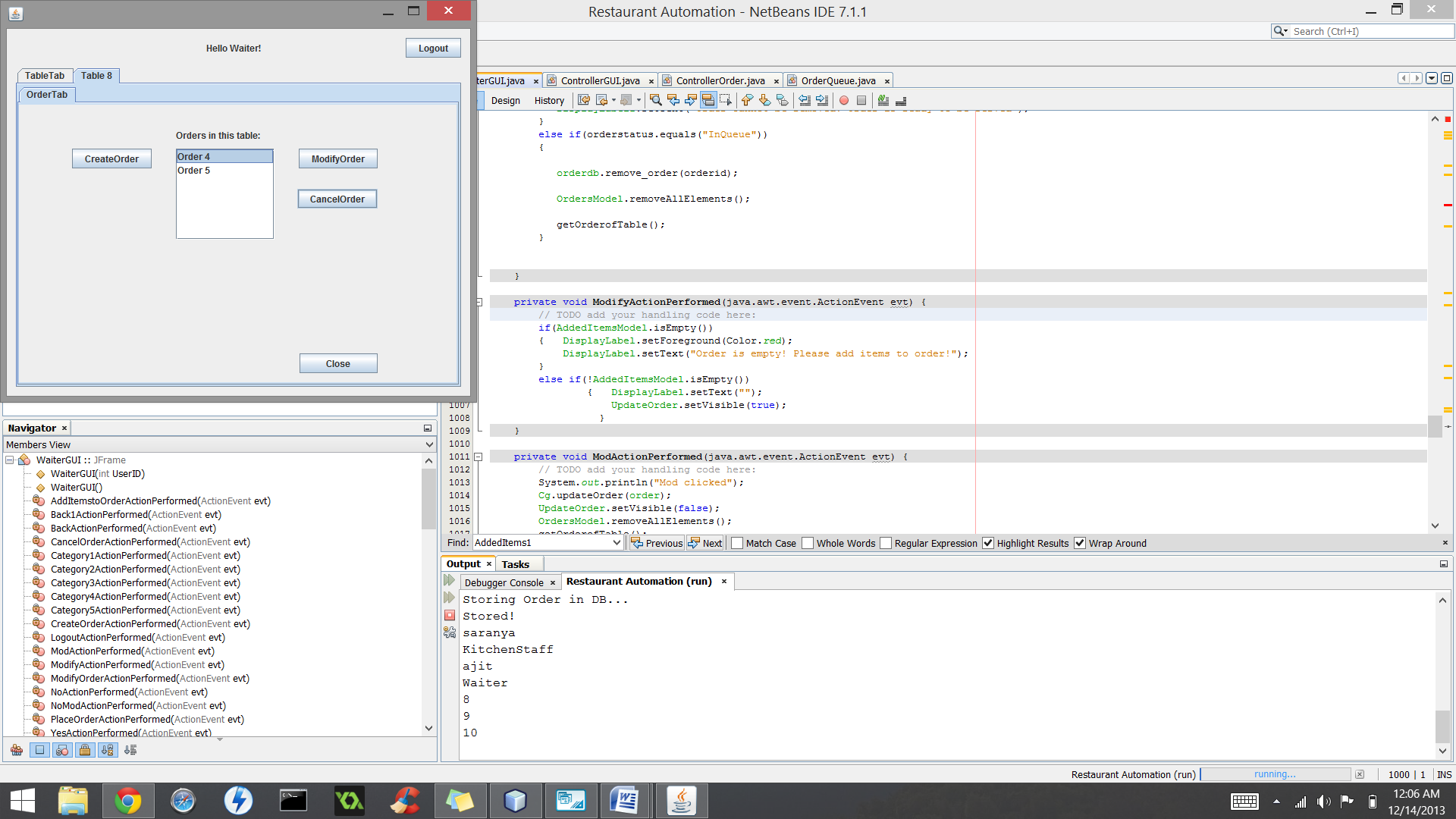
****

****

Kitchen Staff GUI with updated orderQueue



Cancel Order



Order removed from orderqueue

