SEUSL

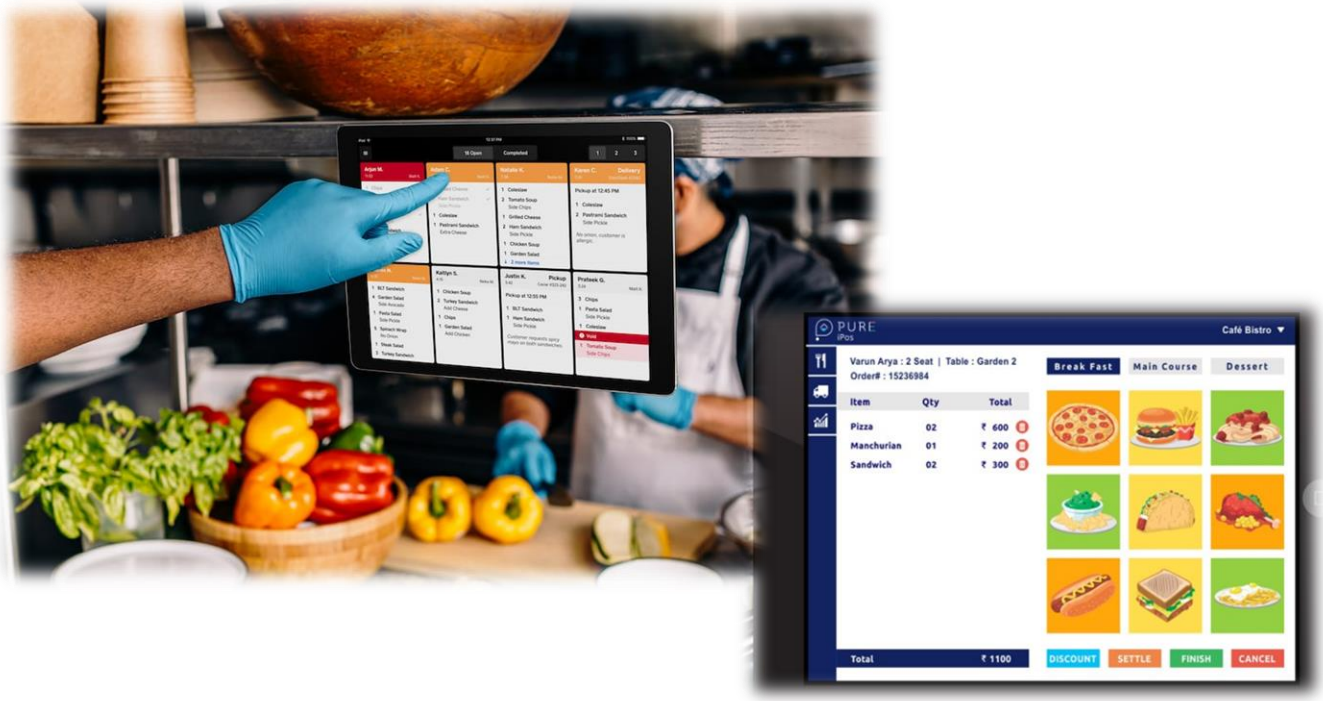
RESTAURANT MANAGEMENT SYSTEM

Group-13

SWT 12031

Practical for Object Oriented Programming

TABLE OF CONTENT



1.Introduction

1.1)Introduction of Project:-

The groups in eating places are now developing continuously. At the identical time, the want for dealing with its operations and obligations arises. The pleasant way to optimize those activities is developing the commercial enterprise on-line as nicely. Today's era encourages excessive-tech offerings specifically over the Internet. Hence the project is evolved proficiently to assist eating place owners automate their commercial enterprise operations. This challenge- “Restaurant Management System” serves the excellent way of retaining patron's invoice and caters their needs.

This record presents information about the complete software program requirements specification for managing bill consisting of the tax in a eating place.

This software program is absolutely self-contained and works distinctly as efficient as different applications associated with the challenge. It affords true and clean graphical user interface to each new in addition to skilled customers of the computer systems.

1.2)Objective of Project:-

To prepare the management for eating place for making billing easier and extra handy. Through this admin can without difficulty calculate the invoice with the tax each time.

2.System Evalution

System analysis refers into the way of inspecting a state of affairs with the motive of enhancing it thru higher processes and strategies. System evaluation is the system of making plans a brand-new device to both update of complement an gift tool. But in advance than any planning is finished the vintage machine assessment, is therefore, the technique of collecting and interpreting records, diagnosing problems and the usage of the information to re-remark development within the machine. Or in other phrases, device evaluation approach an in-depth rationalization or description. Before computerized a device underneath attention, it needs to be analysed. We want to look at the way it capabilities presently, what are the troubles, and what are the requirements that the proposed gadget have to be meet.

The following goals have been considered when conducting system analysis:

1. Determine the desires of the customers
2. Determine the viability of the device concept.
3. Conduct technical and economic evaluation.
4. Assign tasks to the hardware, software, users, databases, and other device additives.
5. Set price range and time restrictions.
6. Establish a device definition so that it will serve as the premise for all ensuing engineering work.

2.1) Analysis of Requirement:

2.1.1) Problem Definition: -

Here, in our "Restaurant Management System" project, we're attempting to make it easier for the admin to compute invoices in a highly linear fashion. like not having to wait in line or lose time when it's his turn to calculate the expenses.

Back to content....

2.1.2) Performance Standards:-

While designing the system, attention should be paid to the following performance characteristics:

- User friendliness: The machine needs to be simple to apply recognize in order that even new customers may also achieve this efficaciously and without any hassle.

- Response time: All operations should have a quick response time, Careful programming can do this.

- User fulfilment: The system must live up to user's expectations.

- Error handling: To hold the gadget from halting, response to users' errors and unwanted situations should be taken care of.

- Robustness: The machine should be capable of get better on its very own from unintentional events.

- Safety: The system must be capable of stopping or preventing catastrophic behaviour.

2.1.3) System Prerequisites:

- Java
- NetBeans
- SQLite Database

Technical Requirements:

- **OS Compatibility:**
  - Windows XP,7,8,10
  - Linux (Enterprises & Version)

- **Minimum Hardware Requirements:**
  - Processor- Pentium four
  - Ram-512MB

Back to content....

2.1.4) Acceptance Criteria:-

For the evaluation of the new system, the following acceptance criteria were established:

• User-Friendliness: The system must satisfy user requirements and be simple to understand and operate.

• Modularity: The system's components should be largely independent and have just one function.

• Maintainability: The system should be able to be improved and maintained with less time and effort in the future.

• Timeliness: - Under normal, peak, and recovery situations, the system ought to function well.

• The system designed should be accurate and thus dependable, meaning the outputs should be constant and accurate, and the error rate should be kept to a minimum.

• The execution and response times should both be extremely brief.

• The system should be effective, meaning that its use of resources should be optimal

2.2) Feasibility Study:

All tasks are feasible given limitless belongings and countless time. Unfortunately, development of laptop-primarily based machine in plenty of instances is much more likely to be plagued thru shortage of resources and delivery date. Hence, we've were given made use the concept of reusability that's what Object Oriented Programming (OOPs) is all about.

The feasibility report of the project holds the blessings and versatility of the mission. This is split into 3 sections:

- Economic feasibility

- Technical feasibility
- Behavioural feasibility
- Operational feasibility

2.2.1) Economical feasibility:

Under financial feasibility we try to investigate that this software program have monetary blessings over the currently current manual device. In destiny if the consumer desires a few changes in the device they may effortlessly be replaced. So, it complements the financial feasibility.

1. . The number of employees needed to manage this software is far lower than the number needed to manage the current technology. This immediately shows up in the pay paid to the company's personnel and will result in a far lower degree of monitoring control, demonstrating the financial viability of the venture.

2. . Additionally, the stationary required by the current system will be replaced by this software. Because of the client-server capabilities of this software, there is no need to spend money on communication when a sales counter person wants to find out the status of stock that is located quite far from that counter and is located elsewhere. This will undoubtedly reduce company expenses to some extent as well.

3. Several bills are generated with the touch of a button, saving the employees' important time who had been laboring over these tasks for a while.

Economically, the "Restaurant Management solution" proposed system is doable because:

1. The system has a very low time factor need.

2. The system would offer the ability to view numerous facts to aid in making wise managerial decisions.

3. Instead of a slow and error-prone environment, the system will offer a quick and effective automated environment.

Back to content....

4. The system will have GUI interface and very less user-training is required to use it.

5. Prone manual system, which cuts down on the amount of time and labour required to operate the system

2.2.2) Technical feasibility:

- In the initial research phase, we look at the feasibility of the project. We find the chance the machine could be useful to the business enterprise.

- We decide whether the development the solution is a possible or now not. For this reason, the analyst sincerely establishes the feasibility of every opportunity checking out for advantages, charges and different assets.

- By the usage of the software program many technical blessings exist over manual gadget. As this will increase he reliability of the proper data entered in addition to all the discipline relates in the information of any record.

- This machine gives accurate calculation at some stage in the numerous procedures within the machine. System is likewise easy to handle, smooth to apply and so forth.

- The primary advantages that during destiny if we need to add or trade a few provision inside the machine we are able to without problems do this so on this way the contemporary machine is technically feasible.

1. We discovered that the assignment is technically feasible because minimum hardware and software program required inside the mission is less. These can be received pretty without problems.

2. The essential benefits that in future if we want to add or alternate some provision in the machine then technically we are able to do that without problems so on this way the present day device is technically viable.

2.3) Proposed System Functionality: -

The proposed device can be designed to support the following features:

- The proposed system has a user-pleasant Interface for porting of statistics to database.

- The proposed device provides the facility to drag the records from the server of the required supply order wide variety and get the respective invoice.

- The proposed machine presents no replication of records.

SRS of Project

Introduction:-

This file affords details about the entire software program necessities specification for the software Restaurant Management System a facts for coping with payments in a restaurant.

Purpose:-

The reason of this mission is to provide easy invoice era facility to the clients of all categories.

Scope:-

The name of the project is Restaurant Management System.

The software provides the following facilities to the customers:

- Facilitates easy bill generation with tax.
- Provides price about the meals available in the restaurant.
- Provides email facility for future correspondence.
- Provides backup facility.
- Can add as many meals at a time.

The software will not provide the following facilities to the customers:

- Cannot cancel the bill once generated
- Cannot make bills without tax
- Responsibility of damages
- The product cannot be changeable when once confirmed

The goal of this software program is to offer smooth assistance to each the patron as will because the admin with right database statistics.

Definitions, Acronyms and Abbreviations:-

- Customer : The person who purchases the food/meal
- Admin : The person who calculates the bill

 Order No : The unique id given to customer at the bill generation

References:-

Refer the appendix for the database information on bill management in a restaurant.

Overview:-

The following subsections provide the complete assessment of the software specifications requirements documentation for Restaurant Management.

The entire SRS is documented in view of each customers and the admin and the following sub sections are organized to give a whole outlook of the software, its attitude, functions, machine necessities and users understand how's.

Product Perspective:-

The software is absolutely self-contained and works fantastically as green as different programs related to the difficulty. It offers simple database as opposed to complex ones for high requirements and it gives excellent and easy graphical consumer interface to each new, native as well as reviews users of computers.

System Interfaces:-

Interfaces:-

The software presents precise graphical interface for the the front end of the database and a terrific informative interface for the rear cease.

Hardware Interfaces:-

The system should have these hardware requirements:-

- The processor should be at least Pentium 4 or above The processor speed should be greater than 400Mhz The video device should support graphics.
- RAM should be or greater than 512 MB

Software Interfaces:-

The software requires the support of the following softwares for the database and other requirements:

- Java Interpreter (IDE)
- SQ Lite for database
- GUI interface for Jpanal (java GUI)

Communication Interfaces:-

- Local intranet and internet protocols.
- Supports all HTTPS, SMTPS and POP3 services.

Operations:-

The operations required by the users are:

- Customer: order food
- Admin: Bill Generation, Money Collection
- Database Administrator: Update Order number in the database

User Characteristics:-

- No pre knowledge of java or netbeans
- No pre knowledge of database management
- Should know English
- Should be able to calculate bill manually with a calculator

Assumptions and Dependencies:-

- The admin assumes that the customer knows the meal it is ordering for its lunch or dinner

3.Project Planning

What is Project Planning?

Project planning comprises a set of inter related activities which, when carried out systematically, result in production of appropriate software project plan.

There are 12 major steps in project planning:

1. Establish project goals, objective and scope.
2. Determine the strategies to be followed for achievement of goals and objective.
3. Identify the tasks and activities to be performed for achievements of project goal and objectives using work break down structure (WBS) approach.
4. Identify project risk.
5. Estimate resources required for each activity.
6. Develop an activity schedule.
7. Develop a project budget.
8. Forcast possible project scenarios at future milestone.
9. Design a suitable project organization for successful execution of the project.
10. Develop a policy framework as guide for decision making in various different functions.
11. Develop detailed procedure for implementing policies and carrying out variousactivities.
12. Develop performance standard for measuring performance of individuals group, and the project as a whole.

The most significant managerial task is planning. Real monitoring and controlling of the project are impossible without a sound plan. In many software projects, planning may also be the weakest activity, and poor planning is often a contributing factor in failures brought on by poor management.

Why Do Project Planning?

- Many projects are giant, with many pieces that should be coordinated.
- Long time period tasks require making plans to make sure development (within a semester) and continuity (throughout semesters).
- Short-time period tasks require planning to meet formidable time table dreams.
    - ✓ A common plan helps make sure that each one group participants are working toward the equal purpose. • Planning helps you decide which tasks to do first and which may be postponed.
- Planning helps you decide which tasks to do first and which may be postponed.
- Planning enables you discover resource desires before the resources are wanted.
    - ✓ You can improve your potential to estimate how lengthy duties will take by maintaining music of your projected and accomplished schedules.

The assignment plan makes a speciality of the following key troubles:

- ✓ Schedules and milestones
- ✓ Plans for Software first-class warranty
- ✓ Plans for configuration control
- ✓ Plans for Project monitoring

- ✓ Cost projection
- ✓ Management of chance

Planning for project

The most crucial operations, after cost estimation, may be schedule estimation and staff demand estimation. In the absence of phase-wise cost information, both are related. Estimating the schedule has as its objective figuring out how long the project will take overall and how long each phase will last.

The person-month cost has no bearing on the schedule. It is not possible to generate a schedule directly from the overall estimate by deciding on the average staff size, then calculating the total time needed by multiplying the entire effort by the typical staff size.

Project Milestones and scheduling:

A schedule for the project can be created once we have an estimate of the work and time needed for the various phases. The Gantt chart, which represents project schedule using a calendar-oriented chart, is a conceptually straightforward and practical scheduling approach.

4.System Design

Designing is the maximum critical segment of software application improvement. It requires a careful planning and thinking at the a part of the system clothier. Designing software program way to devise how the numerous additives of the software program are going to attain the favored goals. It need to be completed with utmost care due to the fact if the section carries any mistakes then a good way to have an effect on overall performance of the machine, as a give up end result it can take greater processing time, more reaction time, extra coding workload and so forth

Software format sits at the technical kernel of the software program engineering technique and completed no matter the software process version this is used. After the software program necessities were analyzed and precise, software program design is the number one of the 3 technical activities Designing, Coding and Testing thet are required to construct and verify the software program. Each pastime transforms statistics on this type of manner that in the long run consequences in established laptop software program.

5.1 DESIGN TARGETS

The While developing the system, the following objectives were kept in mind:

• Improve the system's usability. This was required so that the system could be used effectively and serve as a catalyst for attaining goals.

• Ensure system compatibility, or that it will work with the entire integrated system. Future upkeep and improvement must be scaled back.

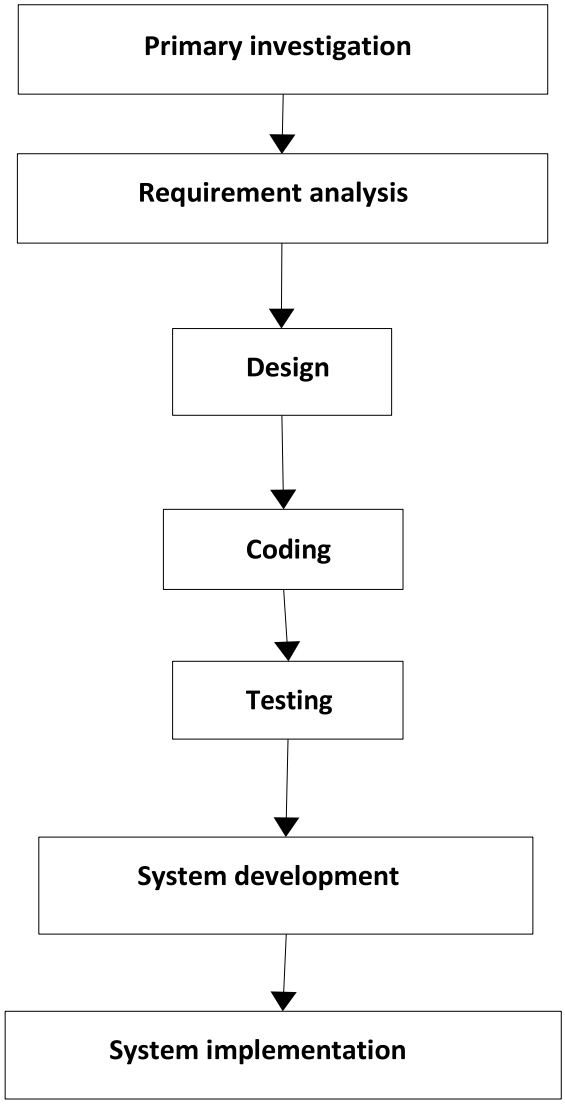
• Create compatibility inside the system so that it can incorporate other system modules.

• Make the system dependable, clear, and economical.

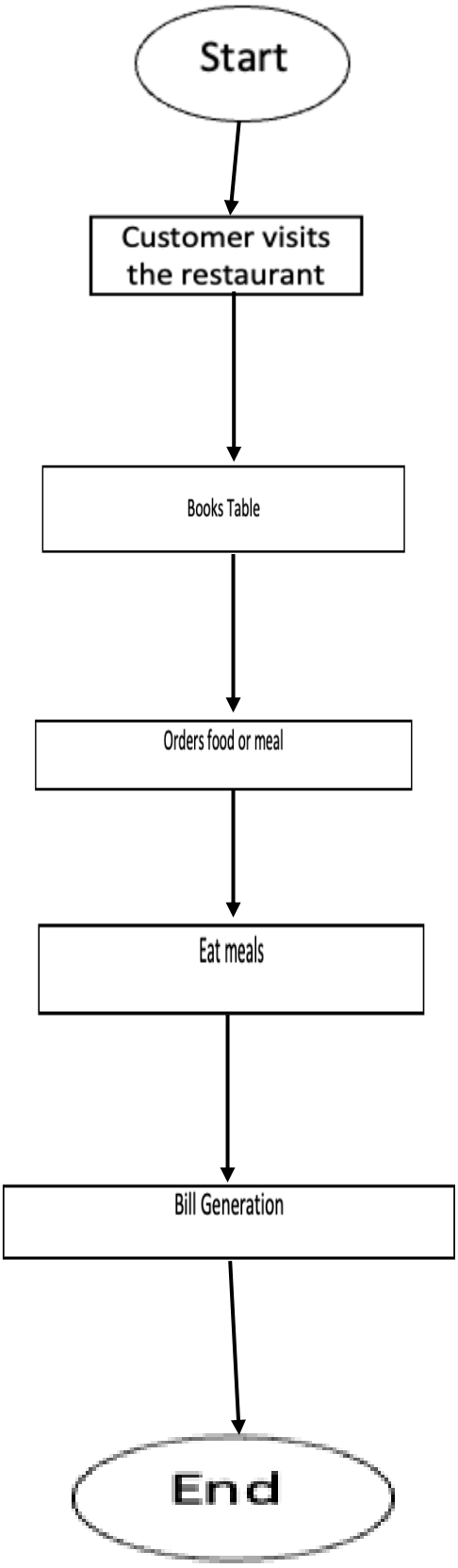
Back to content....

SDLC (Software development life cycle):-

The main reason for having a SDLC process is that it breaks the problem of development software into successfully performing a set of phases, each handling a different concern of software development.



5.2 Functional flow of the system

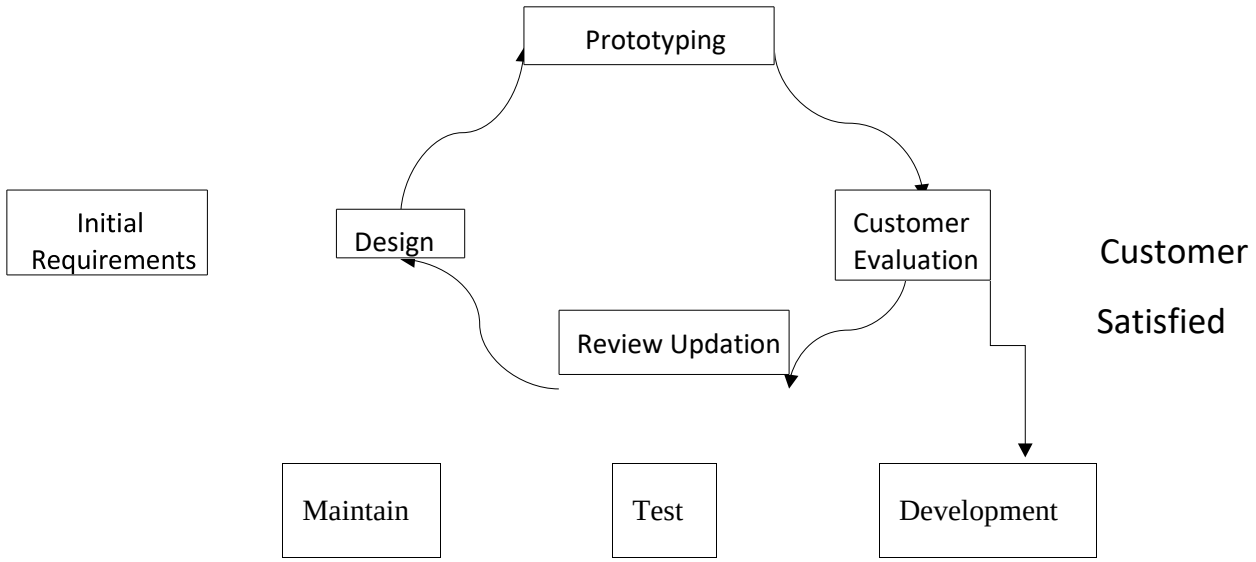


5.3) Methodology Model

Prototyping Model

For better development of a project, it is very important to select a relevant model. Prototype Model can be one of the best options for such a purpose.

This model is completely suitable for our project, so we are selecting this model, more details about this model is following.



Prototype Model

5.4) ENTITY RELATIONSHIP DIAGRAM

- ERD can express the overall logical structure of a database graphically.

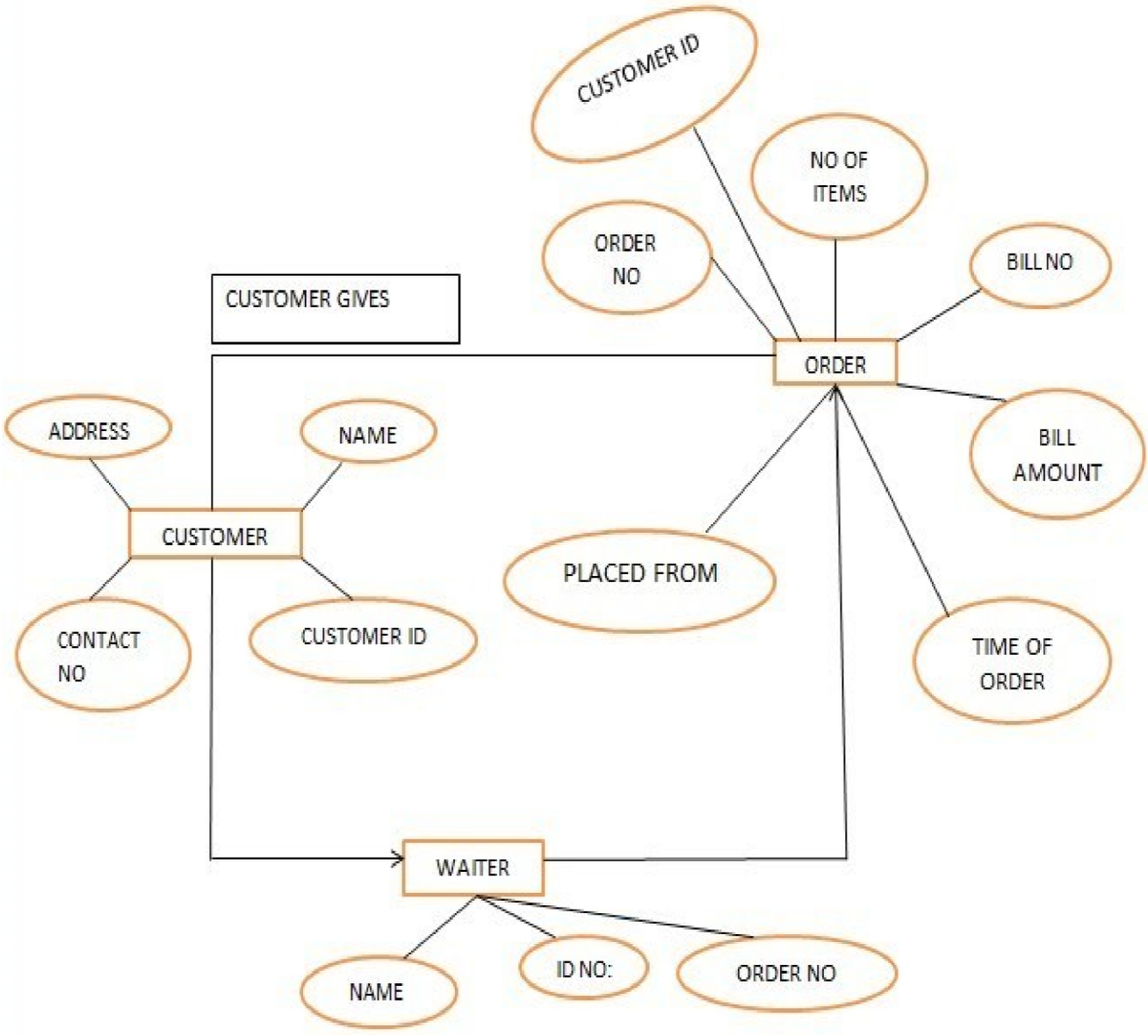
  ERD are simple and clear qualities that may well account in large part of the widespread use if the ER – Model.

- An entity, which is a "Thing" in the actual world with a separate existence, is the fundamental object that the ER model depicts.
- For each of an entity's attributes, a value will be assigned.

The database's data is largely composed of the attribute values used to describe each entity.

Back to content....

ER Diagram for Ordering Food in a Restaurant



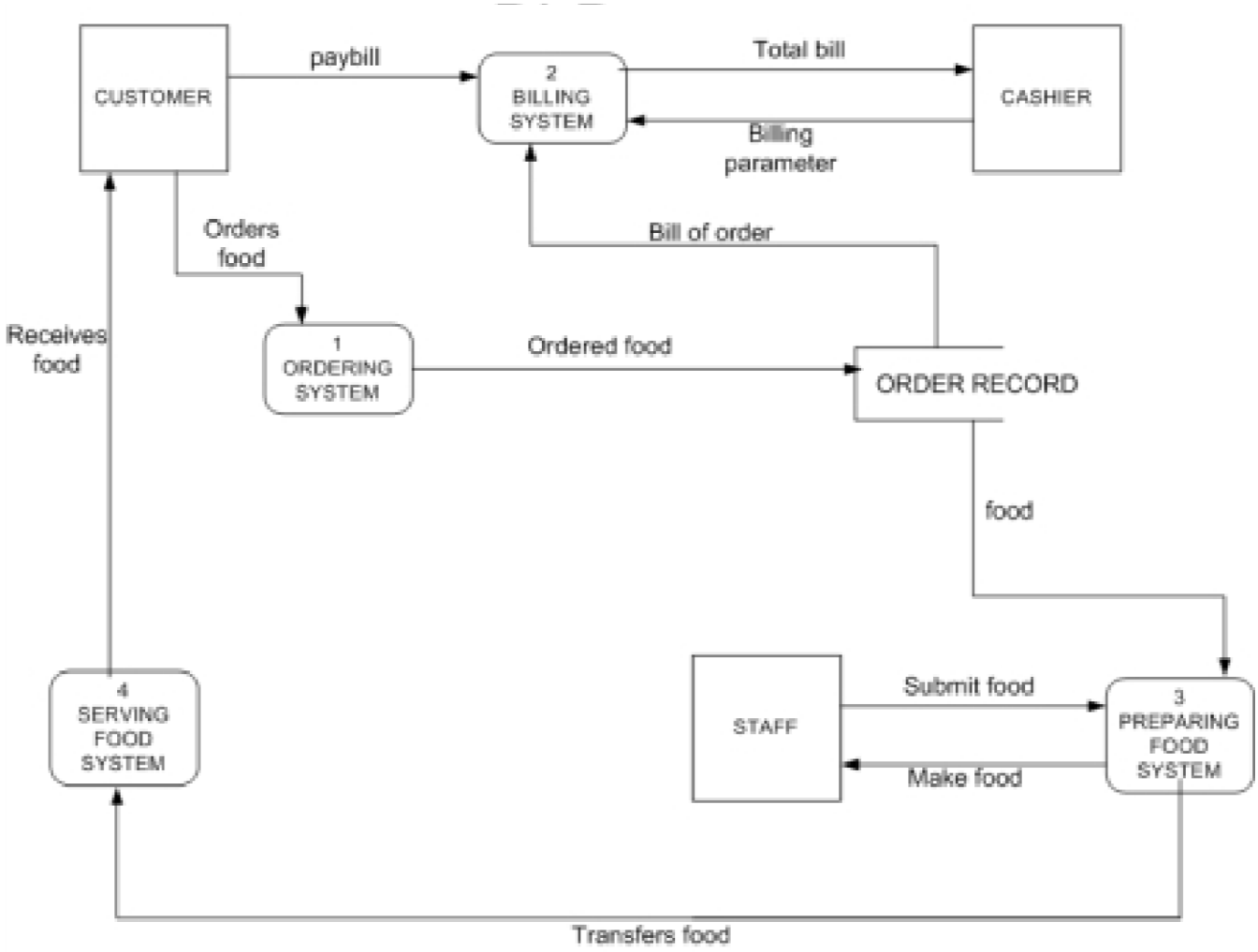
5.5) DATA FLOW DIAGRAM (DFD)

- DFD are commonly used during problem analysis.
- A DFD depicts the data flow through a system.
- The DFD tries to record the changes made to the input data by the system so that finally the output data is produced.
- Procedures are not represented by a DFD.
- DFD is a logical design of the flow of information between the module.
- The DFD shows the interaction of one module with the other module.

There are several level of DFD are used to describe the flow of information

1) **0 Level / Context Level**
2) **1 Level**
3) **2 Level**
4) **3 Level**

DFD for Restaurant Management System (Level-1)



5.Technologies Used

1. **Front End:**
   - NetBeans-Java
2. **Middle Ware:**
   - java v6.6
3. **Back End:**
   - SQLite Database

Technical Requirement:

a. **OS Compatibility**

   i. Windows 7,8,10

   ii. Linux(Enterprise & Version)

b. **Processor** - Pentium4

c. **RAM** – 512 MB

6.Testing

Software checking out is a critical element of the ultimate overview of the specification layout coding. Testing of software program ends in the uncovering of mistakes inside the software useful and overall performance necessities are met. Testing also provides a great indication of software program reliability and software quality as an entire. The result of various stages of checking out are evaluated and then in comparison with the anticipated outcomes. If the errors are uncovered, they're debugged and corrected. A method to software program trying out has the familiar characteristics:

- Tests start at the module level and progress "outwards" to cover the complete computer-based system.
- At certain times, various testing methodologies are appropriate.
- Debugging must be taken into account in the testing approach even though testing and debugging are separate tasks.

7.1 Aspirations and Objectives

A program is tested by being run with the goal of identifying errors. A decent test case will likely uncover an error that hasn't been found yet. An effective test is one that identifies an error that hasn't been found "*Testing is a process of executing a program with the intent of finding an error*". A good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error. Our Objective is to design test process that that systematically uncover different classes of errors and do so with minimum amount of time and effort.

7.2 Statement of scope

A description of the software testing's scope has been developed. The following lists all the features that need to be tested. The fundamental tenets that direct software testing are,

- Every test case should be able to be linked back to customer requirements. From the perspective of the customer, the most serious flaws are those that result in the program failing to achieve its requirements.

- Test cases should be planned well in advance of testing. As soon as the requirement model is finished, the testing plan may be started. As soon as the design is finalized, the test cases can be defined in detail. As a result, the full test can be prepared before any code is written.

- Testing ought to start "in the tiny" and work its way up to "in the large." The initial tests that are planned and carried out usually concentrate on the various modules. Testing changes its focus over time in an effort to identify integration issues in clusters of modules and finally in the whole system.

7.3 Test Case

The project must pass a test cases suit before it is launched to ensure that the required functionality is satisfied and that the system's previous functionality is not broken. To do this, there is an existing test case that verifies the previous functionality. To test for the new feature, fresh cases are created and added to the current test suit. The test case gives an input description and compares the observed output with the anticipated output to determine the test case's outcome. If it is different, a failure has occurred and needs to be found

7.Enforcement

The system was tested before the implementation got underway. Successfully implementing a known system design is a key step in the system development life cycle. Implementation is merely putting a new system design into use. The first question that comes to everyone's mind when considering the truth is if the system will be able to produce all required results from a system perspective. User education and file conversion to a computer system's comprehensive replacement. The term "immolation" refers to the process of turning a newly changed system design into a functioning one. Among the implementation's components is conversion. The host implementation evaluation and software upkeep are the additional factors. There are three different implementation styles:

The system was examined earlier than the implementation were given underway. Successfully imposing a acknowledged gadget design is a key step inside the system improvement lifestyles cycle. Implementation is simply setting a brand new gadget layout into use. The first question that involves every person's mind when thinking about the reality is if the machine will be able to produce all required consequences from a machine attitude. User education and record conversion to a computer device's comprehensive replacement. The term "immolation" refers to the process of turning a newly modified device layout right into a functioning one. Among the implementation's additives is conversion. The host implementation evaluation and software program maintenance are the additional factors. There are three unique implementation styles:

- The use of a computer system to replace a manual one.
- Putting in place a new computer system to replace an old one.
- The adoption of an updated application to take the place of an older one.

8.Scope and Limitation

Back to content....

Future scope:

This project is too flexible as simple to take place changes in policies, programme, schedule and facilities with in due time. It will be very easy for programmer to get project formulated on those guidelines.

In future with the help of this project all the users will get more facilities available on line. Right now this application deals with waste quantity of client and fulfils the requirements but I hope in future, it will provides some extra facilities to the user under same roof.

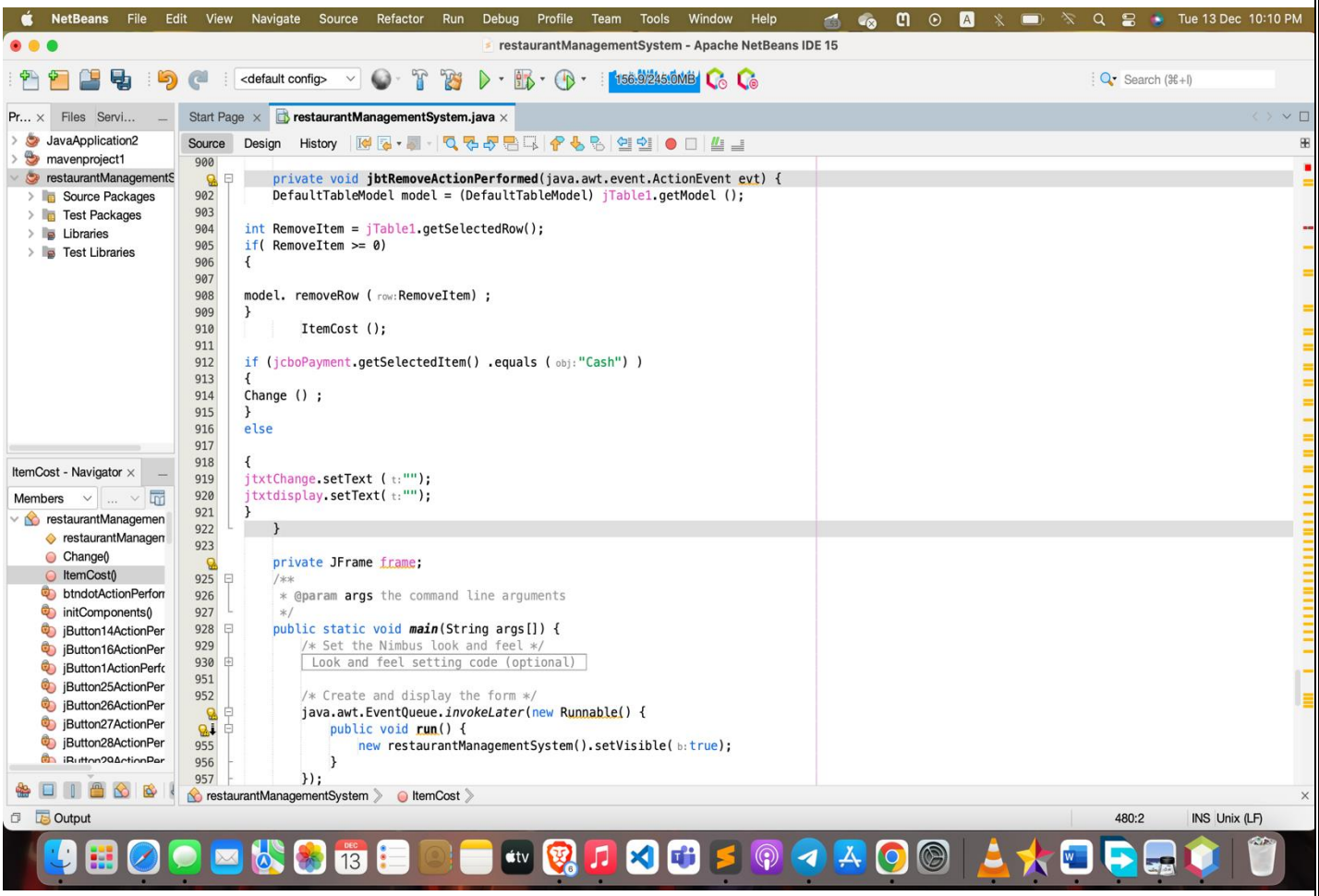
Further inhancement:

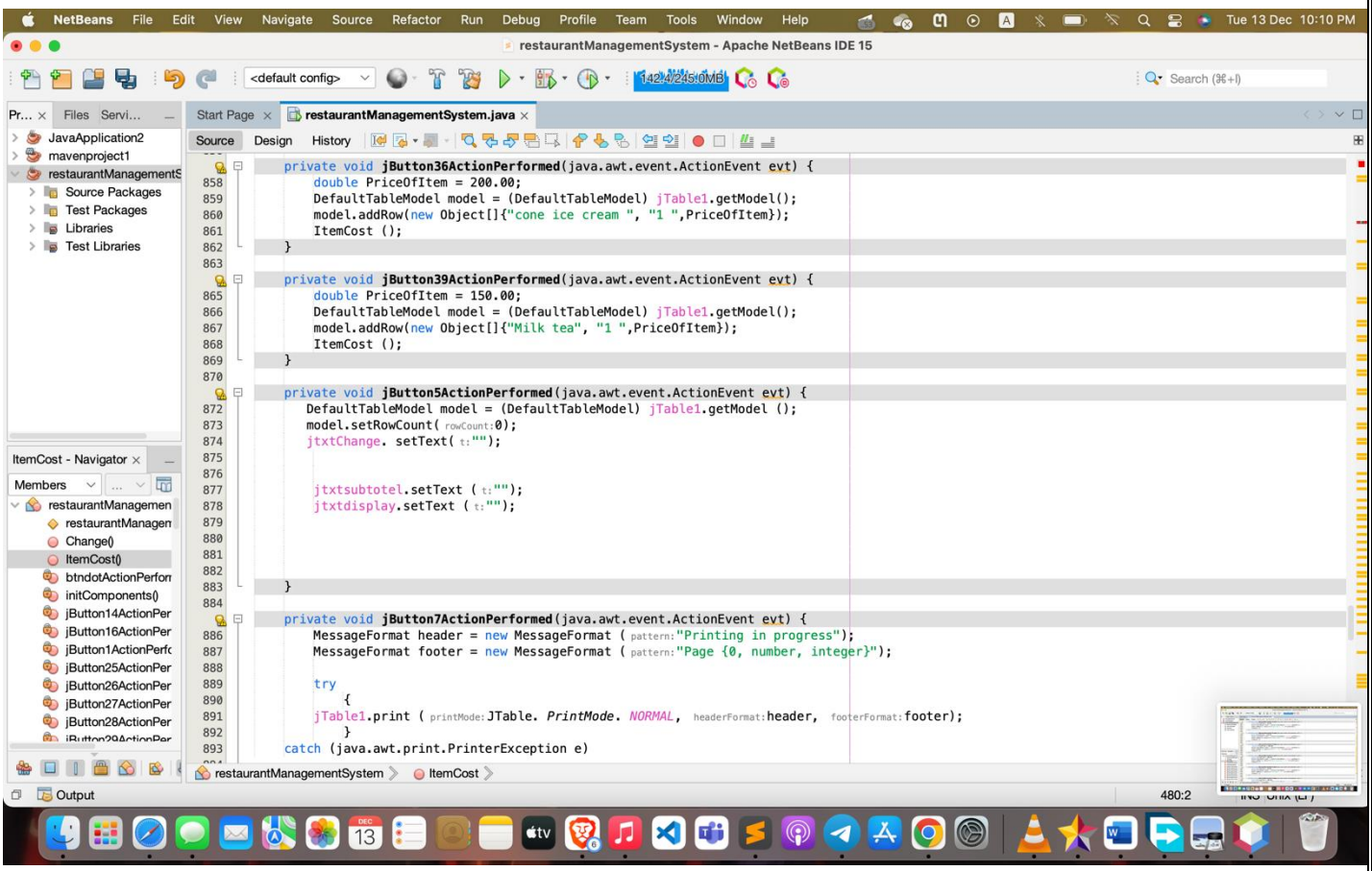
Since project is so flexible and simple that in future it can further inhanced very easily. The project has some limitations will be removed after the satisfaction from client end.

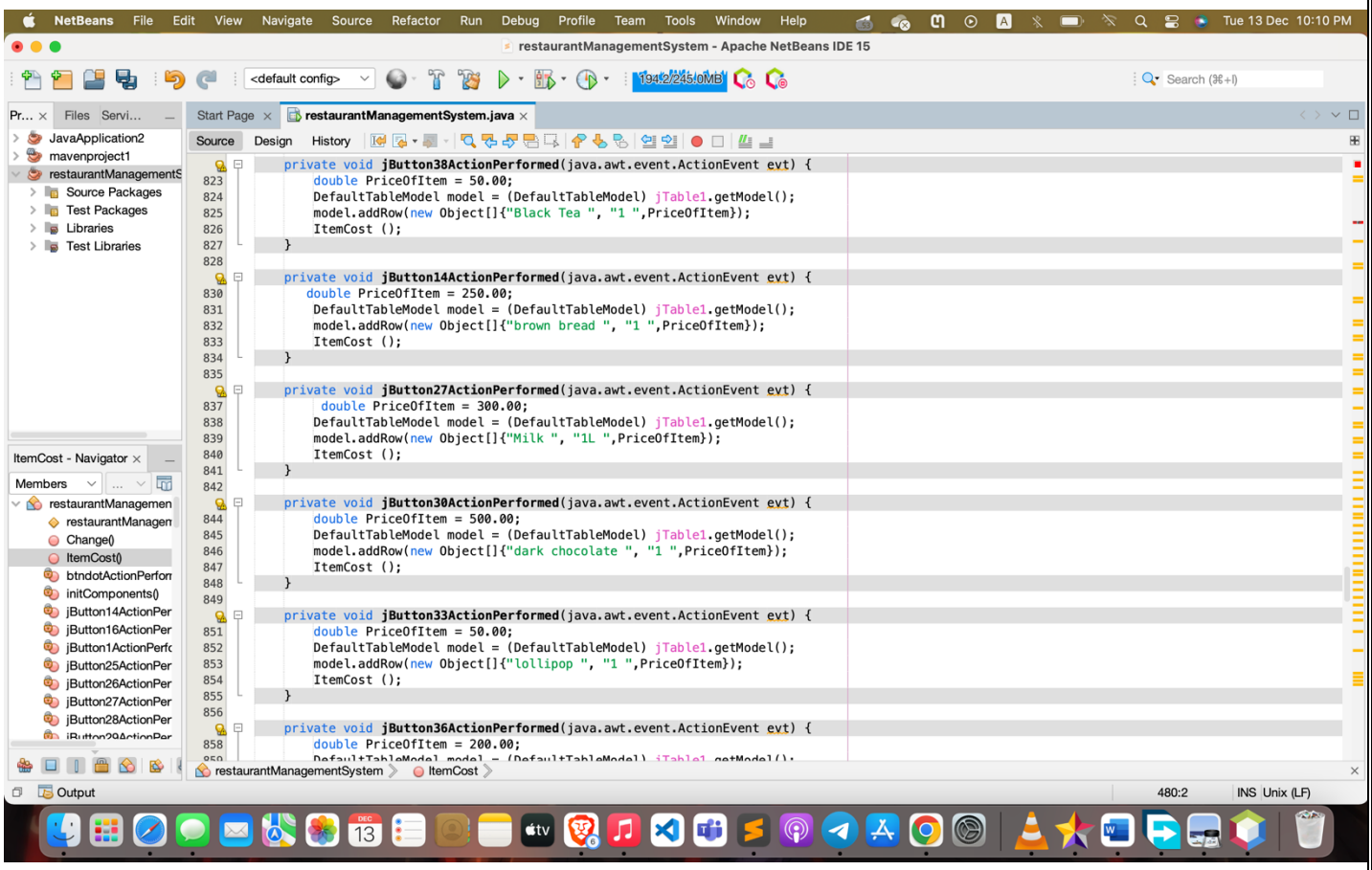
According to me some main topic for further inhancement will be as follows:

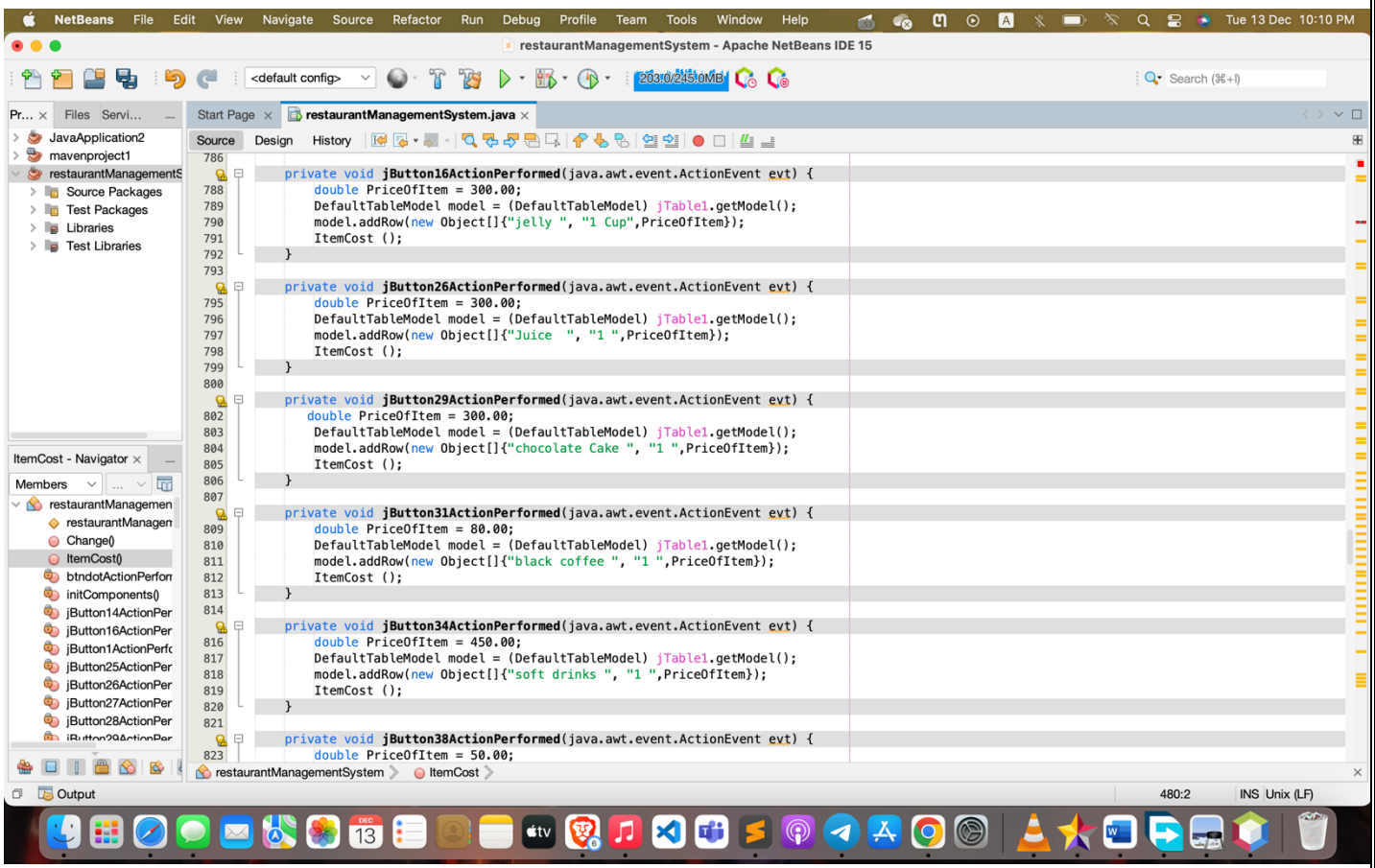
- This project can be inhanced by online table booking for a customer.
- In future client can order the food themselves on the table i.e. no need of the waiter, with the sensors on the table to order food.
- We have also provided a plan for the future enhancers so that they can modify the application in such a way which can generate bills faster.

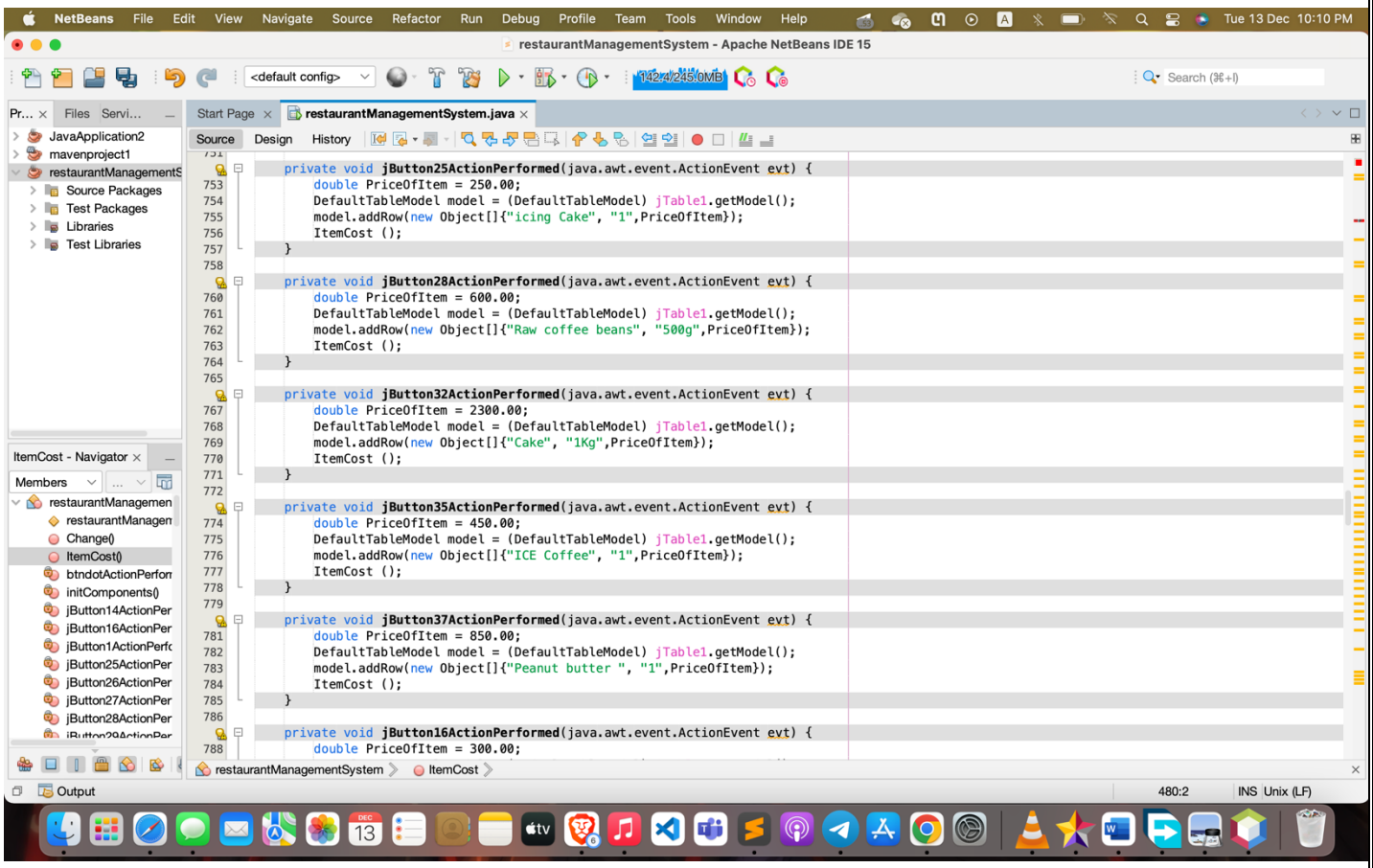
9. Program Coding:

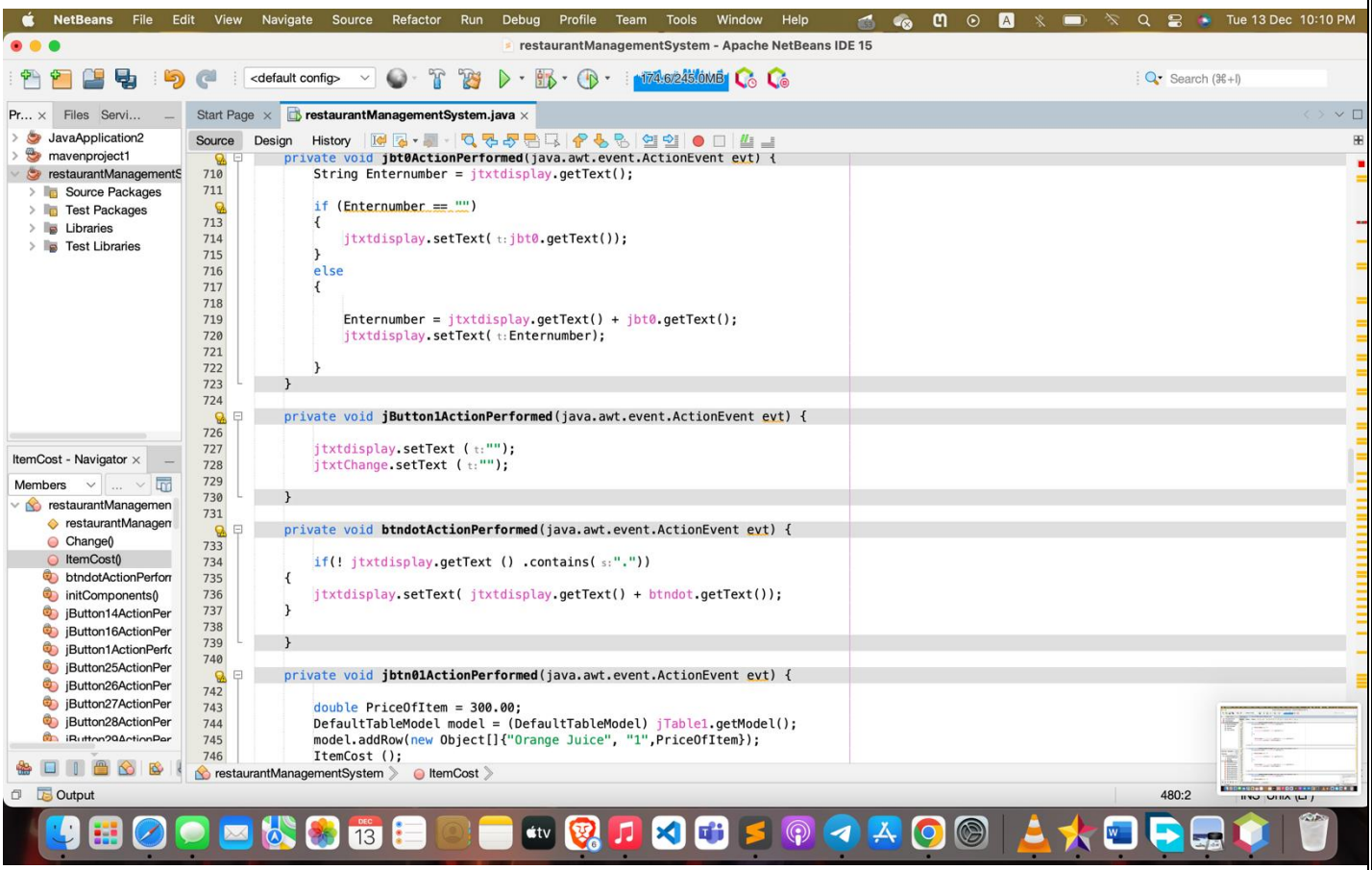


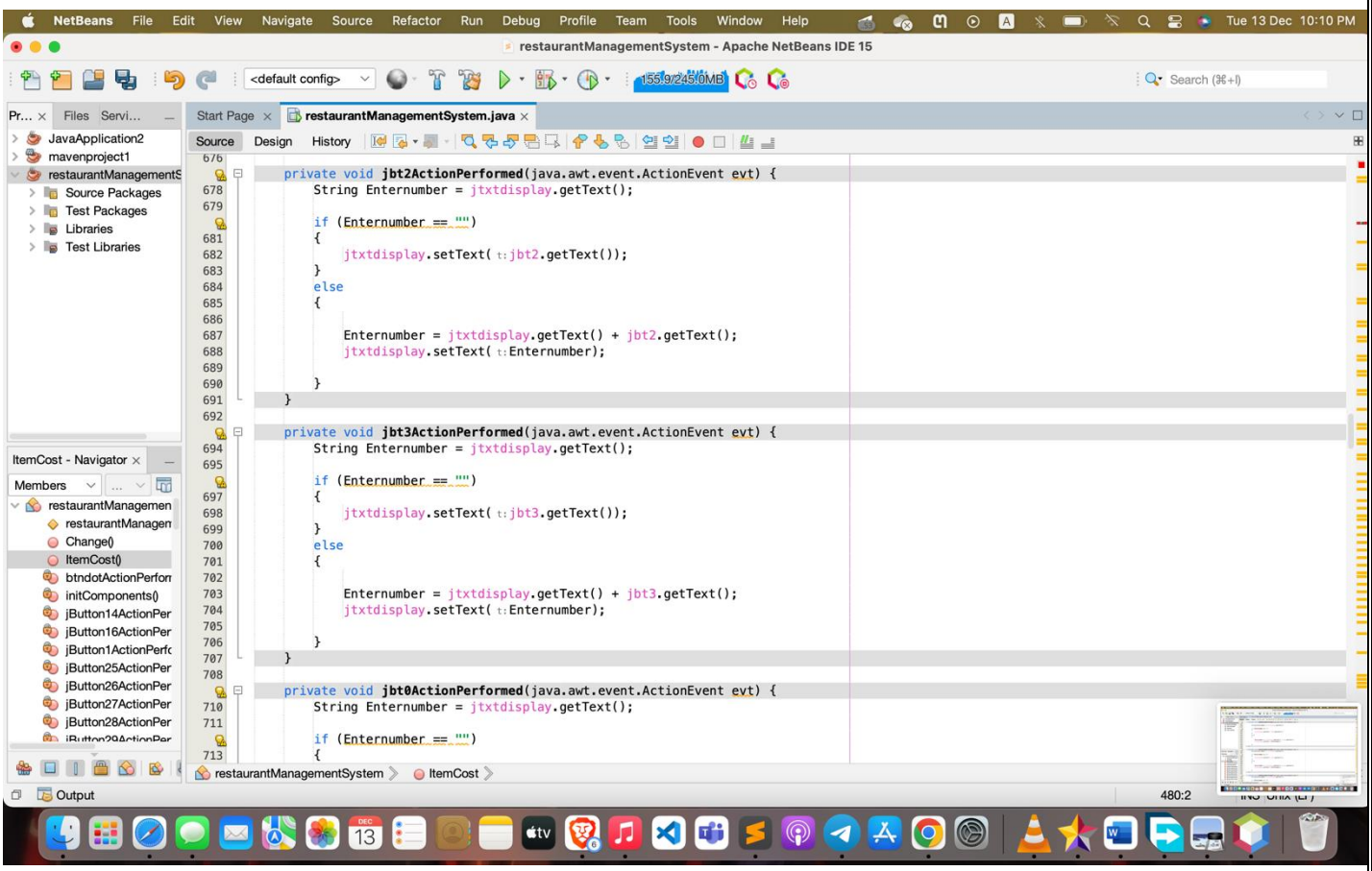

restaurantManagementSystem - Apache NetBeans IDE 15
Start Page
restaurantManagementSystem.java
Source
Design
History
private void jButton36ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 200.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"cone ice cream ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton39ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 150.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Milk tea", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
DefaultTableModel model = (DefaultTableModel) jTable1.getModel ();
model.setRowCount(rowCount:0);
jtxtChange. setText(t:"");
jtxtsubtotel.setText (t:"");
jtxtdisplay.setText (t:"");
}
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
MessageFormat header = new MessageFormat (pattern:"Printing in progress");
MessageFormat footer = new MessageFormat (pattern:"Page {0, number, integer}");
try
{
jTable1.print (printMode:JTable. PrintMode. NORMAL, headerFormat:header, footerFormat:footer);
}
catch (java.awt.print.PrinterException e)
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
restaurantManagementSystem
ItemCost
Output
480:2

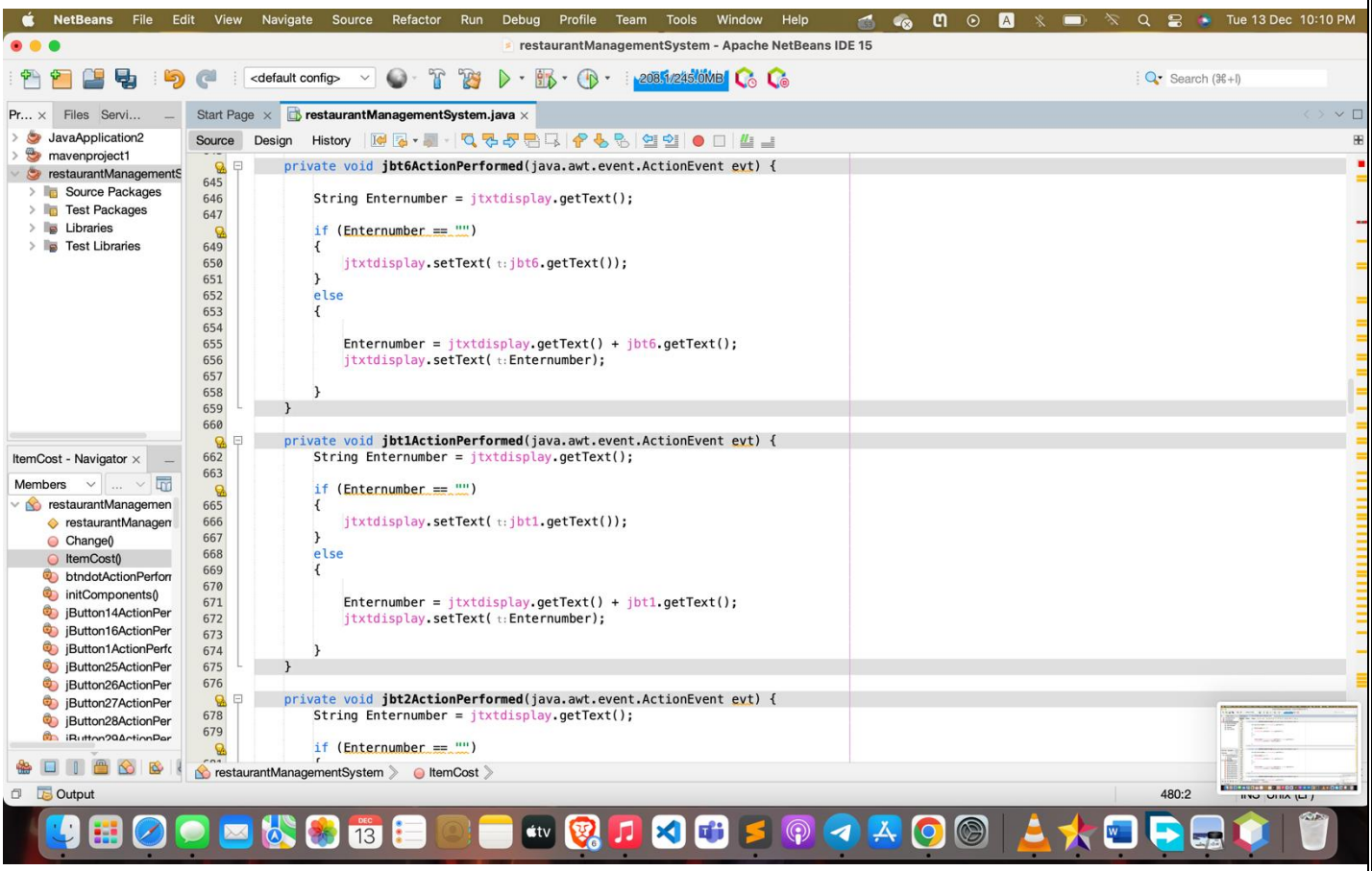

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:10 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfc
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jButton38ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 50.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Black Tea ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 250.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"brown bread ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton27ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Milk ", "1L ",PriceOfItem});
ItemCost ();
}
private void jButton30ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 500.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"dark chocolate ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton33ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 50.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"lollipop ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton36ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 200.00;
restaurantManagementSystem
ItemCost
Output
480:2
INS Unix (LF)

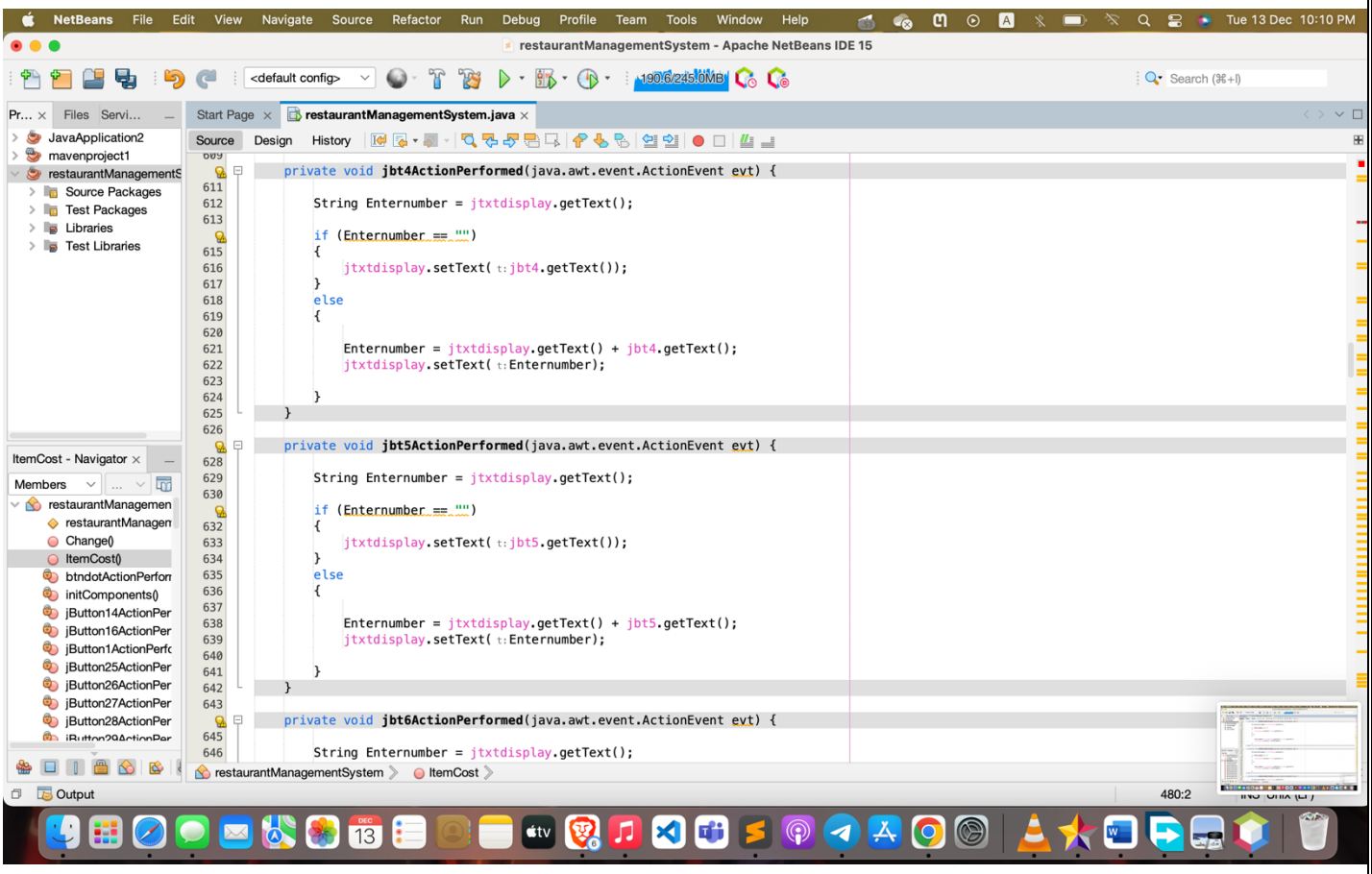

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:10 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfc
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
786 private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {
788 double PriceOfItem = 300.00;
789 DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
790 model.addRow(new Object[]{"jelly ", "1 Cup",PriceOfItem});
791 ItemCost ();
792 }
793
795 private void jButton26ActionPerformed(java.awt.event.ActionEvent evt) {
796 double PriceOfItem = 300.00;
797 DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
798 model.addRow(new Object[]{"Juice ", "1 ",PriceOfItem});
799 ItemCost ();
800 }
802 private void jButton29ActionPerformed(java.awt.event.ActionEvent evt) {
803 double PriceOfItem = 300.00;
804 DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
805 model.addRow(new Object[]{"chocolate Cake ", "1 ",PriceOfItem});
806 ItemCost ();
807 }
809 private void jButton31ActionPerformed(java.awt.event.ActionEvent evt) {
810 double PriceOfItem = 80.00;
811 DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
812 model.addRow(new Object[]{"black coffee ", "1 ",PriceOfItem});
813 ItemCost ();
814 }
816 private void jButton34ActionPerformed(java.awt.event.ActionEvent evt) {
817 double PriceOfItem = 450.00;
818 DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
819 model.addRow(new Object[]{"soft drinks ", "1 ",PriceOfItem});
820 ItemCost ();
821 }
823 private void jButton38ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 50.00;
restaurantManagementSystem
ItemCost
Output
480:2
INS Unix (LF)

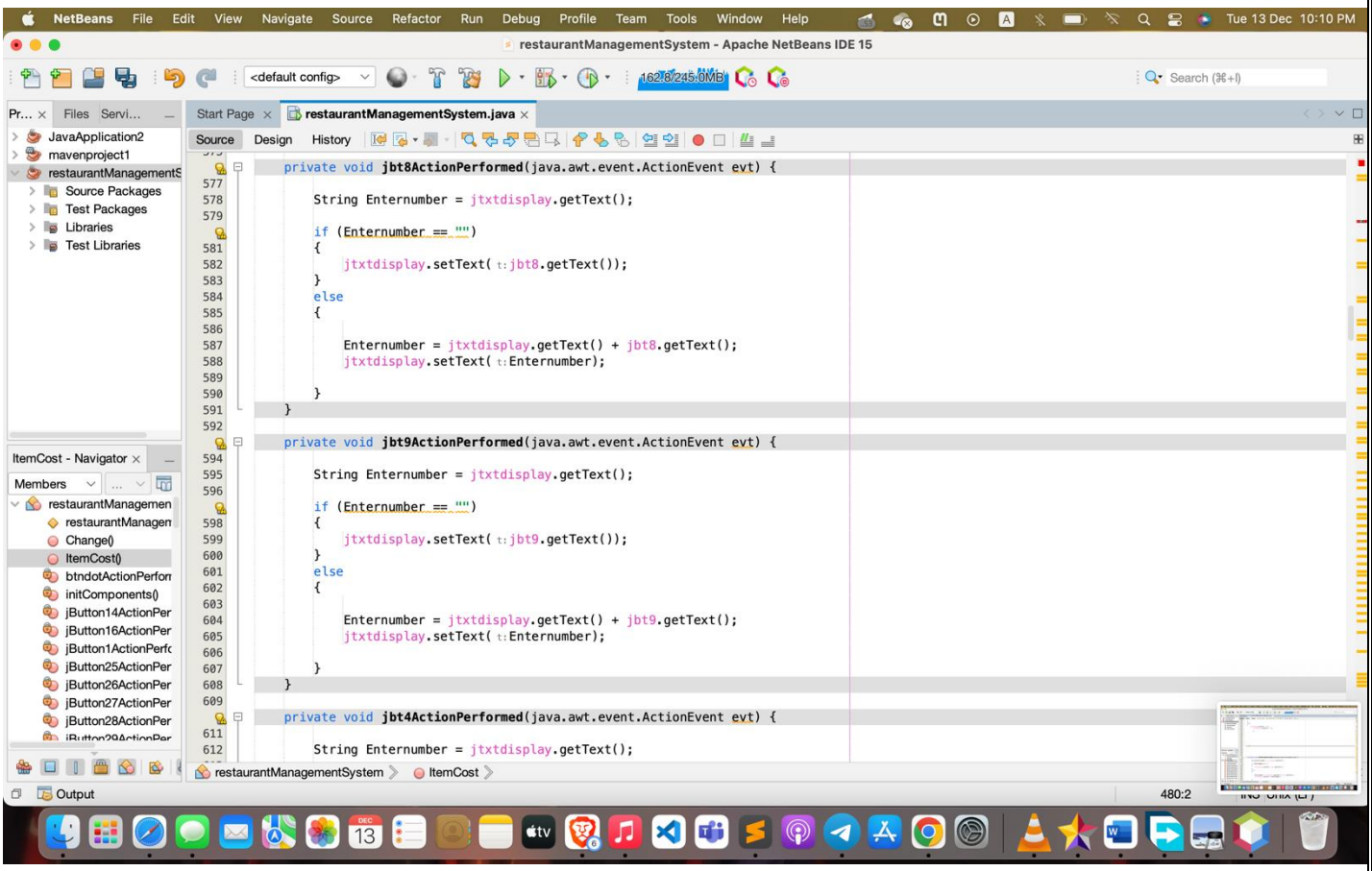

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:10 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfc
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jButton25ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 250.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"icing Cake", "1",PriceOfItem});
ItemCost ();
}
private void jButton28ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 600.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Raw coffee beans", "500g",PriceOfItem});
ItemCost ();
}
private void jButton32ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 2300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Cake", "1Kg",PriceOfItem});
ItemCost ();
}
private void jButton35ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 450.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"ICE Coffee", "1",PriceOfItem});
ItemCost ();
}
private void jButton37ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 850.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Peanut butter ", "1",PriceOfItem});
ItemCost ();
}
private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 300.00;
restaurantManagementSystem
ItemCost
Output
480:2
INS Unix (LF)

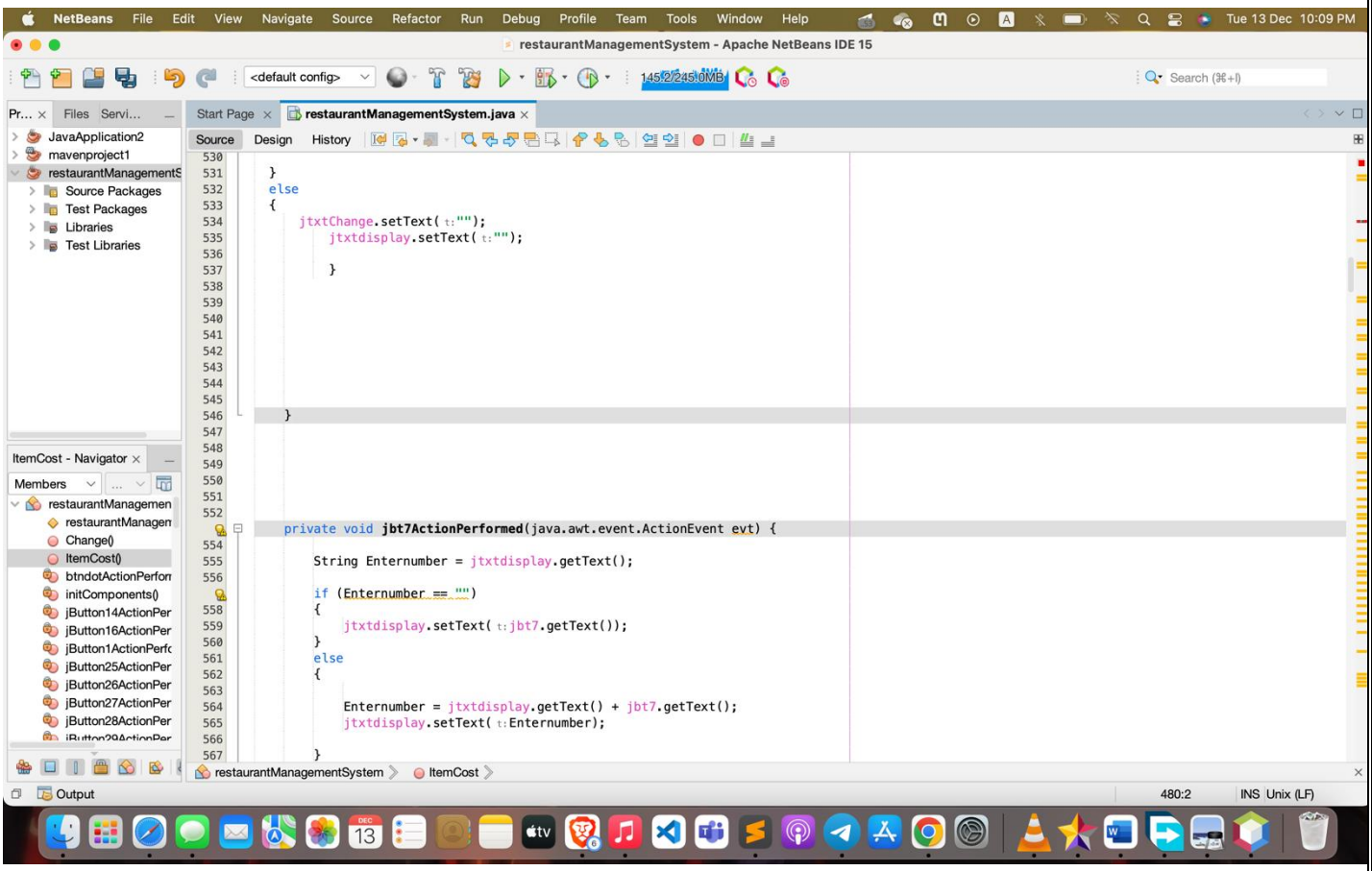

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:10 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jbt0ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt0.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt0.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
jtxtdisplay.setText (t:"");
jtxtChange.setText (t:"");
}
private void btndotActionPerformed(java.awt.event.ActionEvent evt) {
if(! jtxtdisplay.getText () .contains(s:"."))
{
jtxtdisplay.setText(jtxtdisplay.getText() + btndot.getText());
}
}
private void jbtn01ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Orange Juice", "1",PriceOfItem});
ItemCost ();
restaurantManagementSystem
ItemCost
Output
480:2

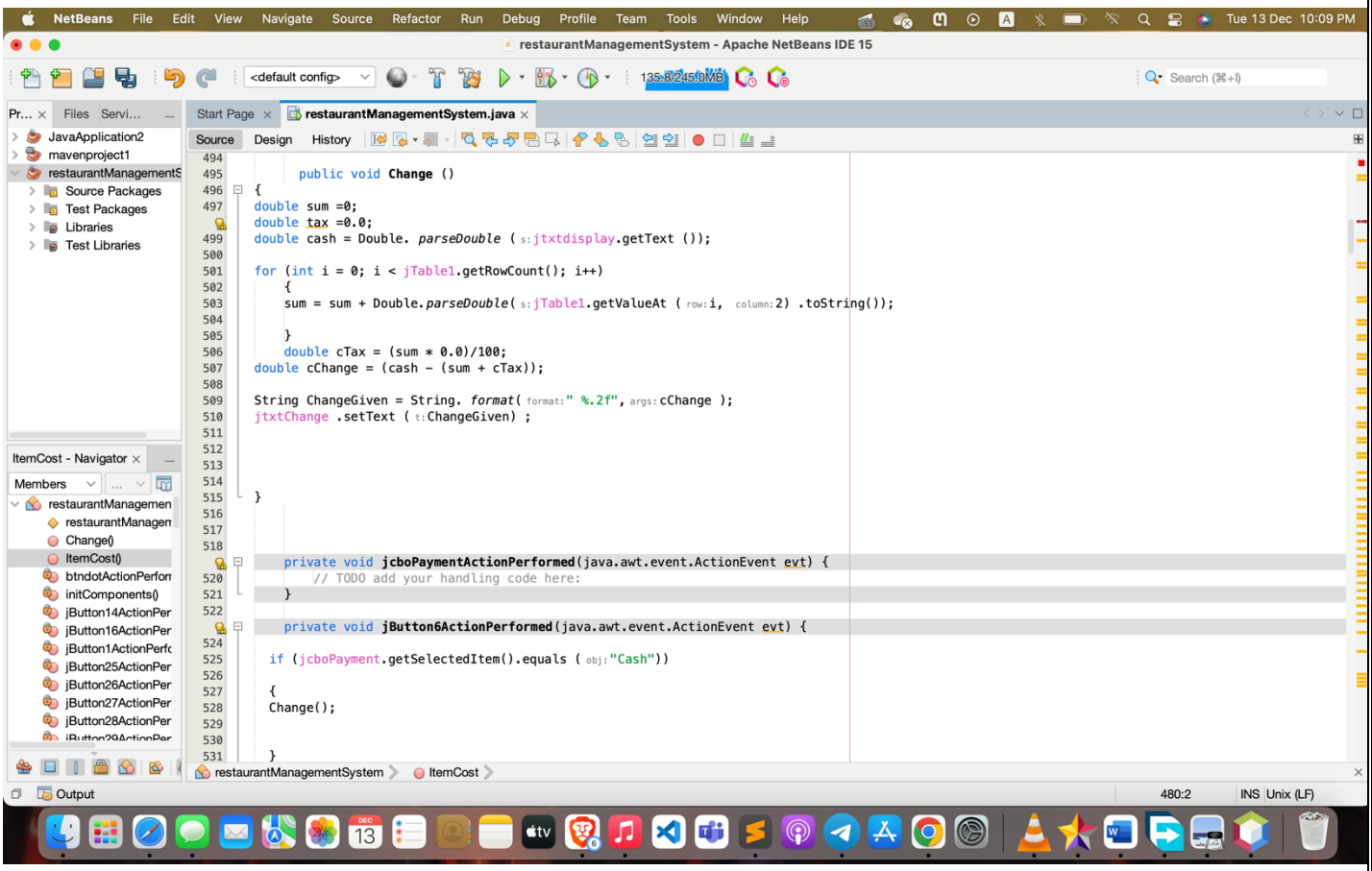

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:10 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
155.9/245.0MB
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jbt2ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t: jbt2.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt2.getText();
jtxtdisplay.setText(t: Enternumber);
}
}
private void jbt3ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t: jbt3.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt3.getText();
jtxtdisplay.setText(t: Enternumber);
}
}
private void jbt0ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
restaurantManagementSystem
ItemCost
Output
480:2

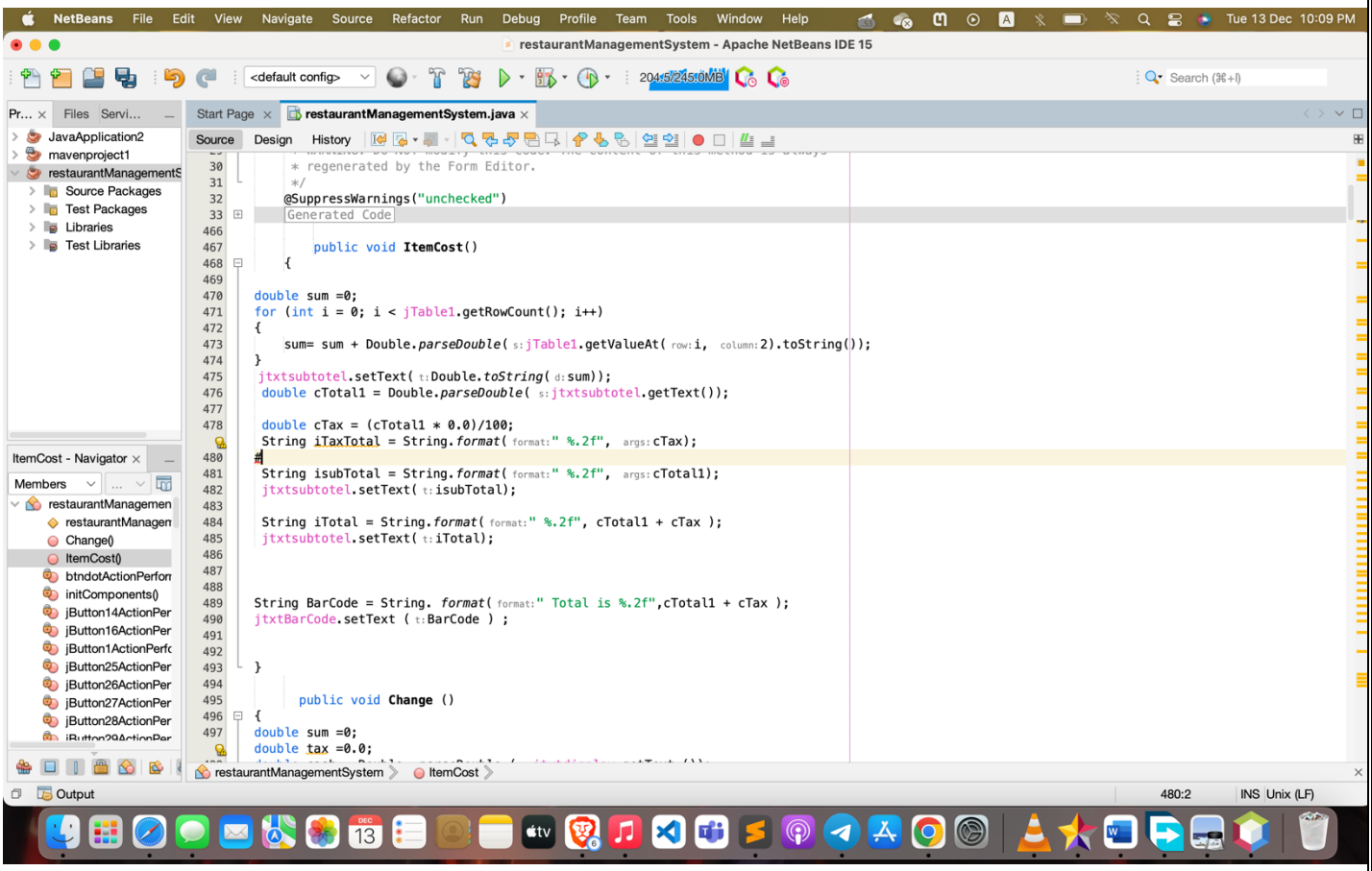

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:10 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jbt6ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt6.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt6.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jbt1ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt1.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt1.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jbt2ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
restaurantManagementSystem
ItemCost
Output
480:2

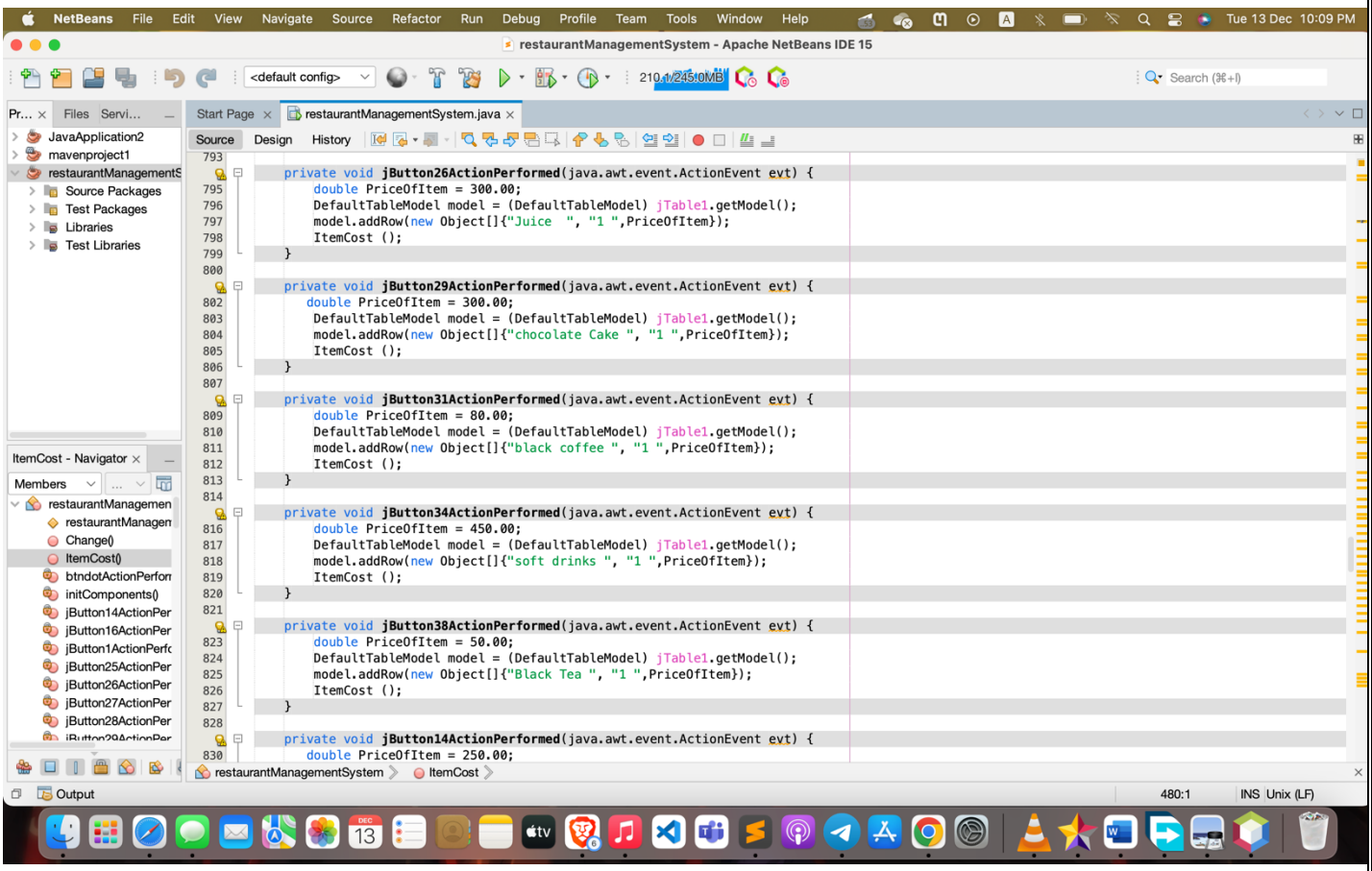

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:10 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jbt4ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt4.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt4.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jbt5ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt5.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt5.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jbt6ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
restaurantManagementSystem
ItemCost
Output
480:2

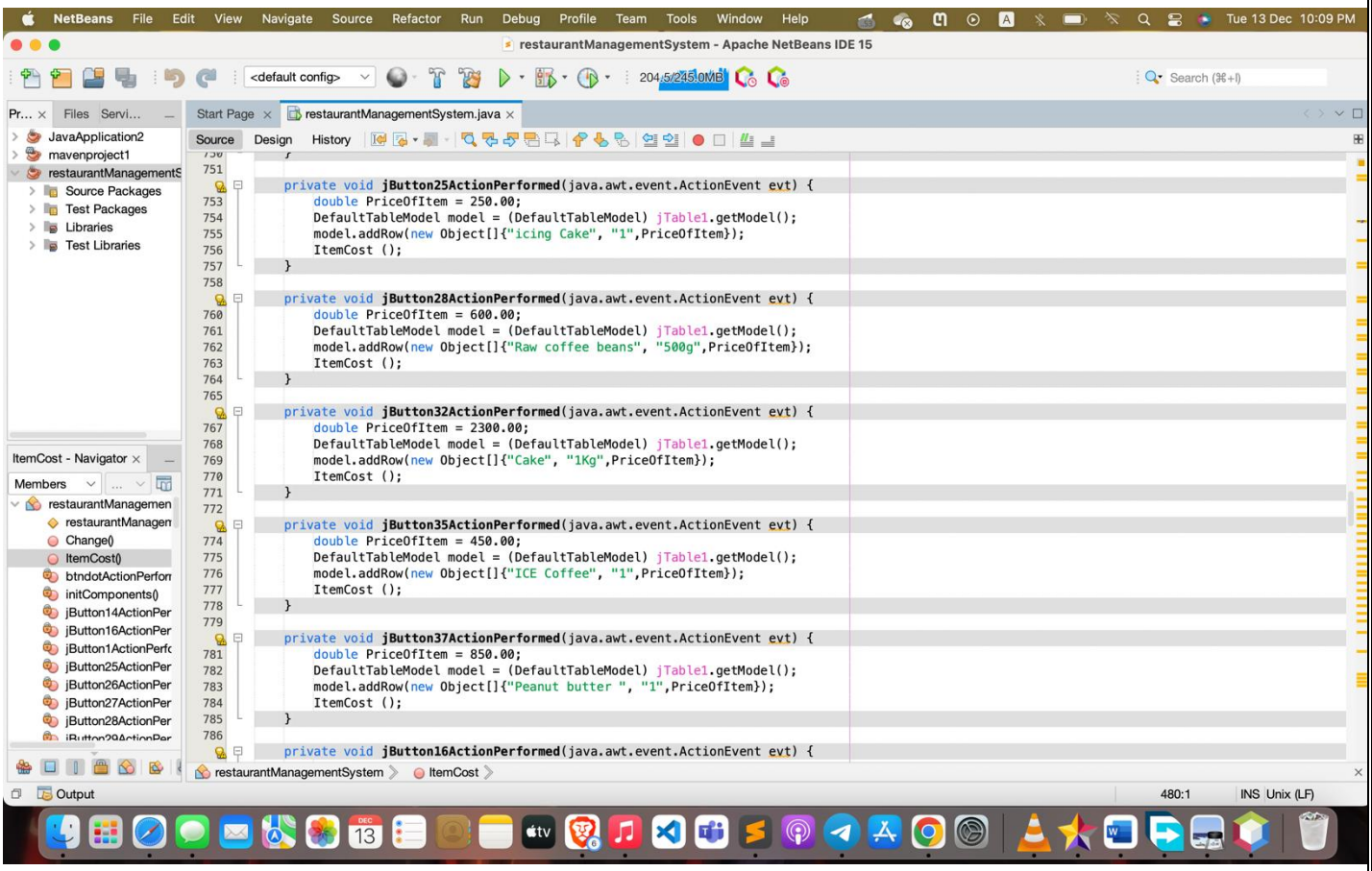

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:10 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jbt8ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt8.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt8.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jbt9ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt9.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt9.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jbt4ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
restaurantManagementSystem
ItemCost
Output
480:2

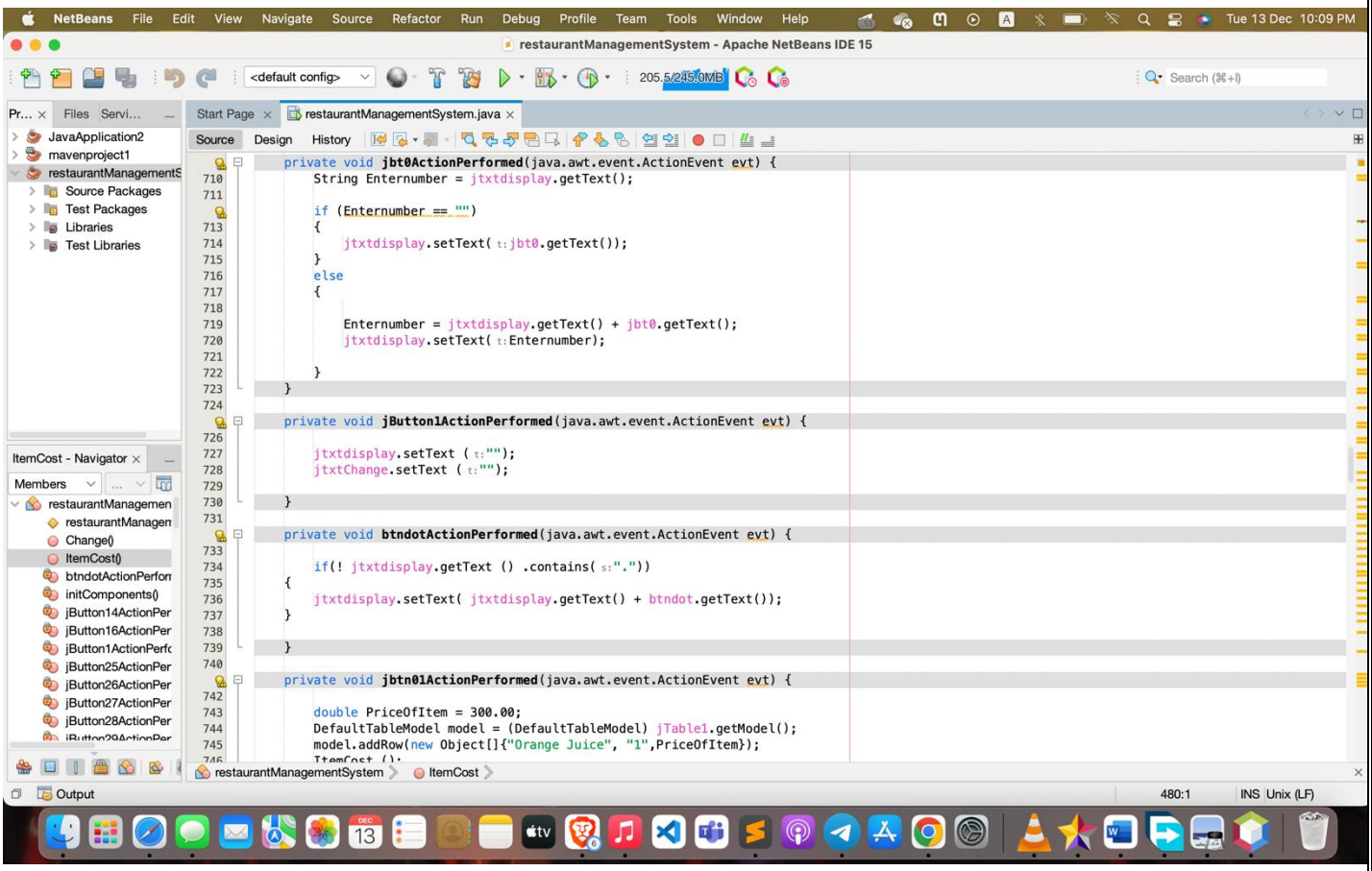

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
}
else
{
jtxtChange.setText(t:"");
jtxtdisplay.setText(t:"");
}
}
private void jbt7ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt7.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt7.getText();
jtxtdisplay.setText(t:Enternumber);
}
restaurantManagementSystem
ItemCost
Output
480:2
INS Unix (LF)

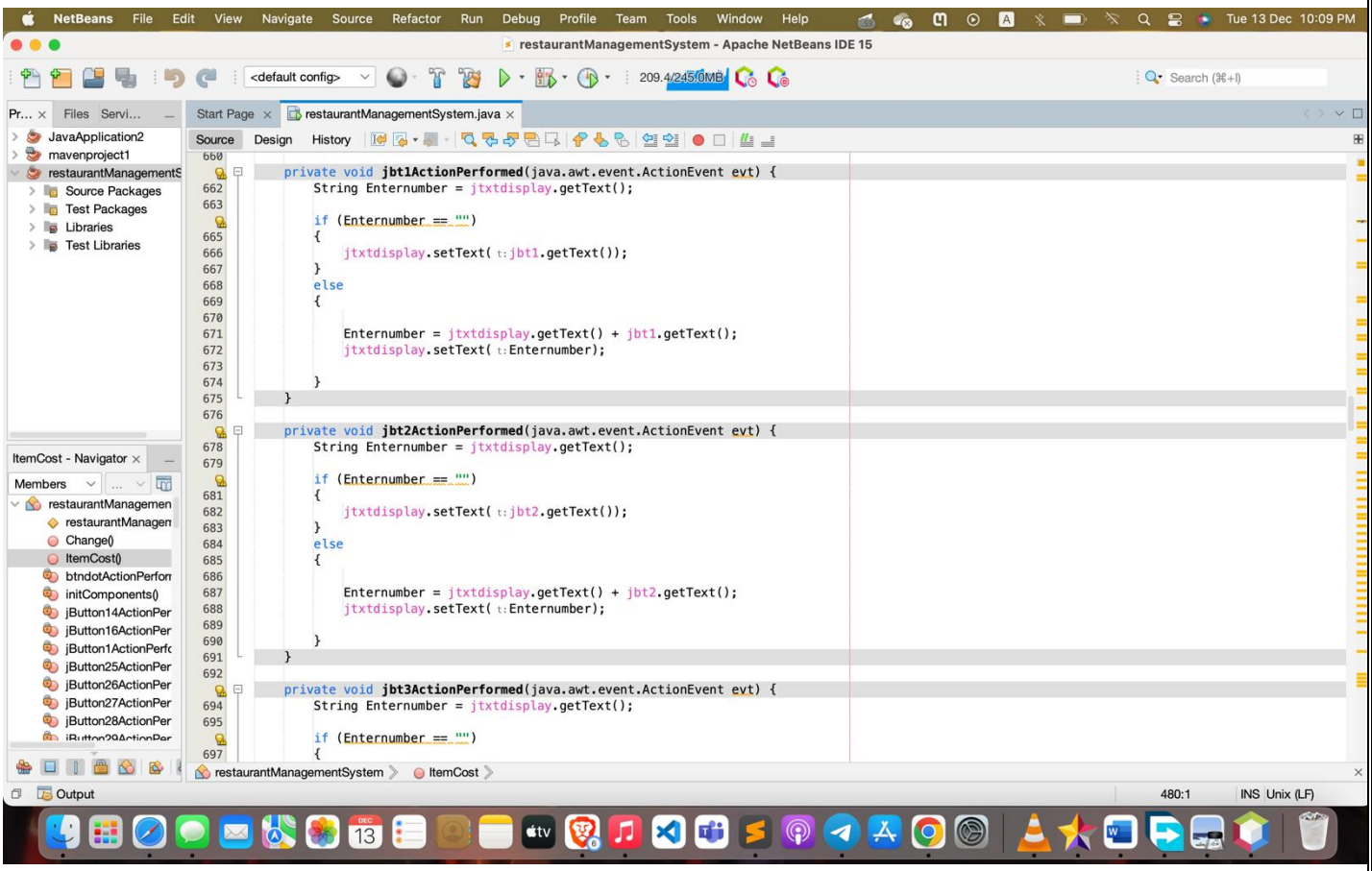

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
494
495 public void Change ()
496 {
497 double sum =0;
double tax =0.0;
499 double cash = Double. parseDouble (s:jtxtdisplay.getText ());
500
501 for (int i = 0; i < jTable1.getRowCount(); i++)
502 {
503 sum = sum + Double.parseDouble(s:jTable1.getValueAt (row:i, column:2) .toString());
504
505 }
506 double cTax = (sum * 0.0)/100;
507 double cChange = (cash - (sum + cTax));
508
509 String ChangeGiven = String. format(format:" %.2f", args:cChange);
510 jtxtChange .setText (t:ChangeGiven) ;
511
512
513
514
515 }
516
517
518
520 // TODO add your handling code here:
521 }
522
524
525 if (jcboPayment.getSelectedItem().equals (obj:"Cash"))
526
527 {
528 Change();
529
530
531 }
private void jcboPaymentActionPerformed(java.awt.event.ActionEvent evt) {
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
restaurantManagementSystem
ItemCost
Output
480:2
INS Unix (LF)

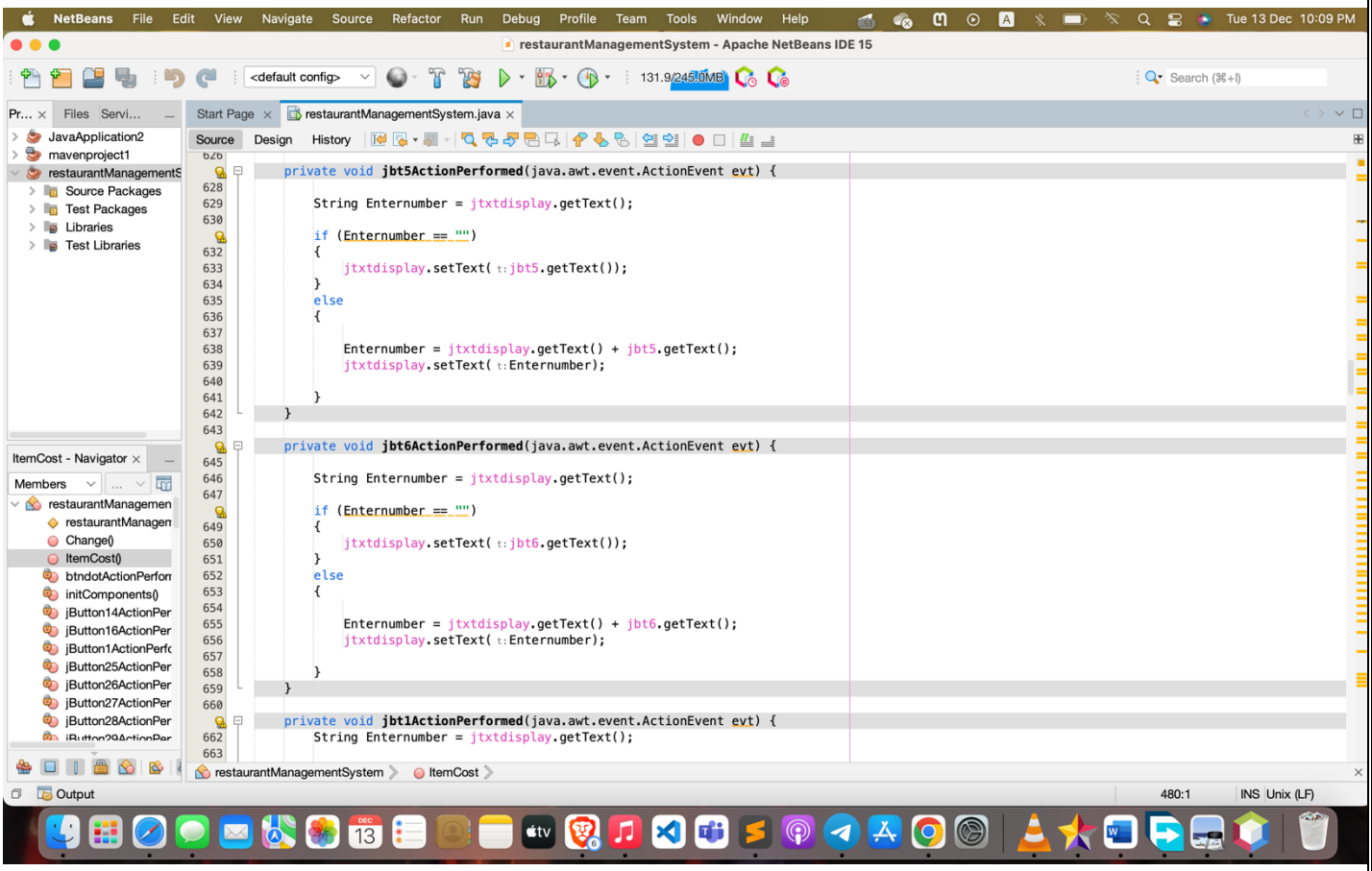

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
Generated Code
public void ItemCost()
{
double sum =0;
for (int i = 0; i < jTable1.getRowCount(); i++)
{
sum= sum + Double.parseDouble(s:jTable1.getValueAt(row:i, column:2).toString());
}
jtxtsubtotel.setText(t:Double.toString(d:sum));
double cTotal1 = Double.parseDouble(s:jtxtsubtotel.getText());
double cTax = (cTotal1 * 0.0)/100;
String iTaxTotal = String.format(format:" %.2f", args:cTax);
String isubTotal = String.format(format:" %.2f", args:cTotal1);
jtxtsubtotel.setText(t:isubTotal);
String iTotal = String.format(format:" %.2f", cTotal1 + cTax);
jtxtsubtotel.setText(t:iTotal);
String BarCode = String. format(format:" Total is %.2f",cTotal1 + cTax);
jtxtBarCode.setText (t:BarCode) ;
}
public void Change ()
{
double sum =0;
double tax =0.0;
restaurantManagementSystem
ItemCost
Output
480:2
INS Unix (LF)

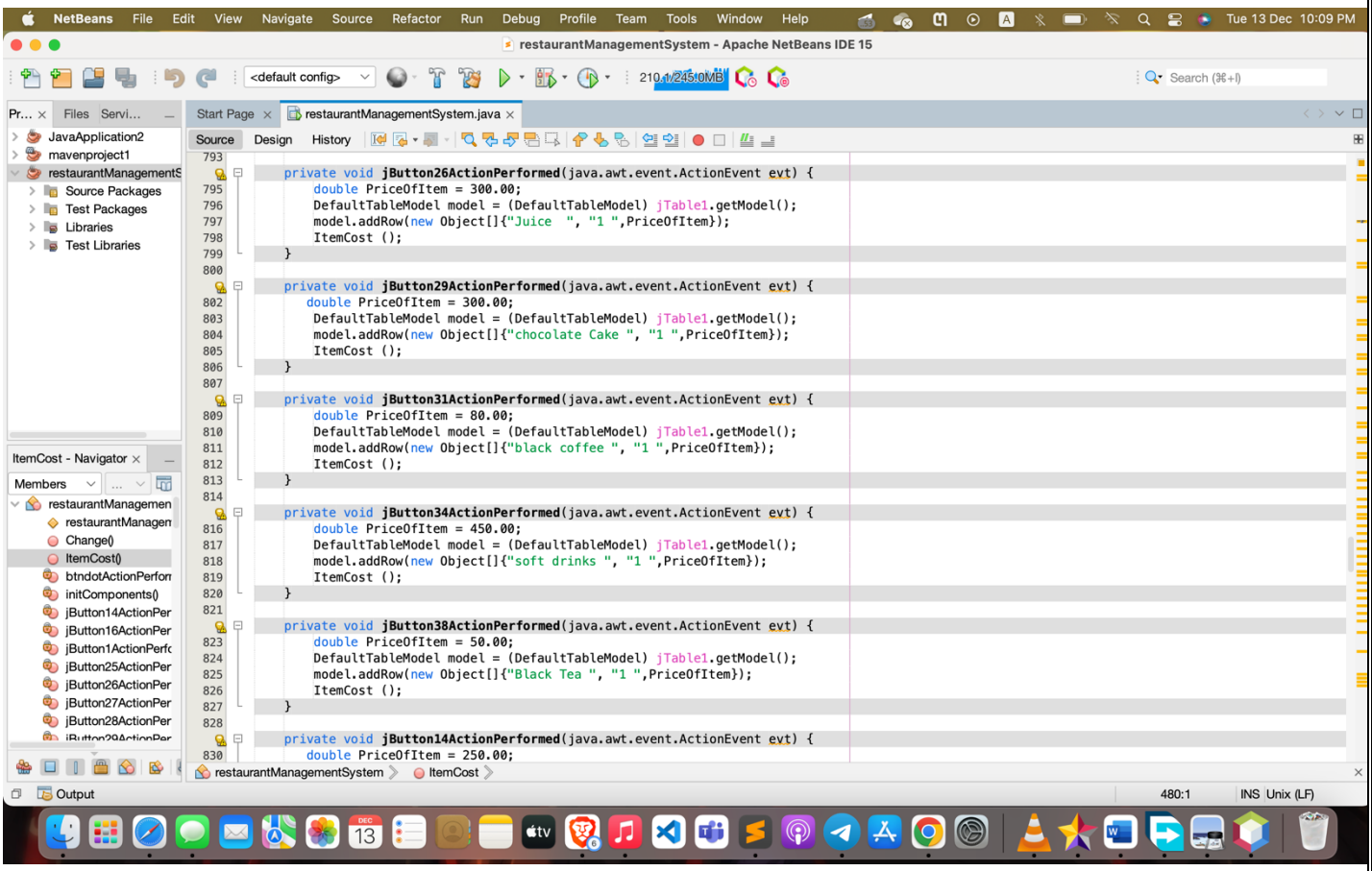

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jButton26ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Juice ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton29ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"chocolate Cake ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton31ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 80.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"black coffee ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton34ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 450.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"soft drinks ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton38ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 50.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Black Tea ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 250.00;
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

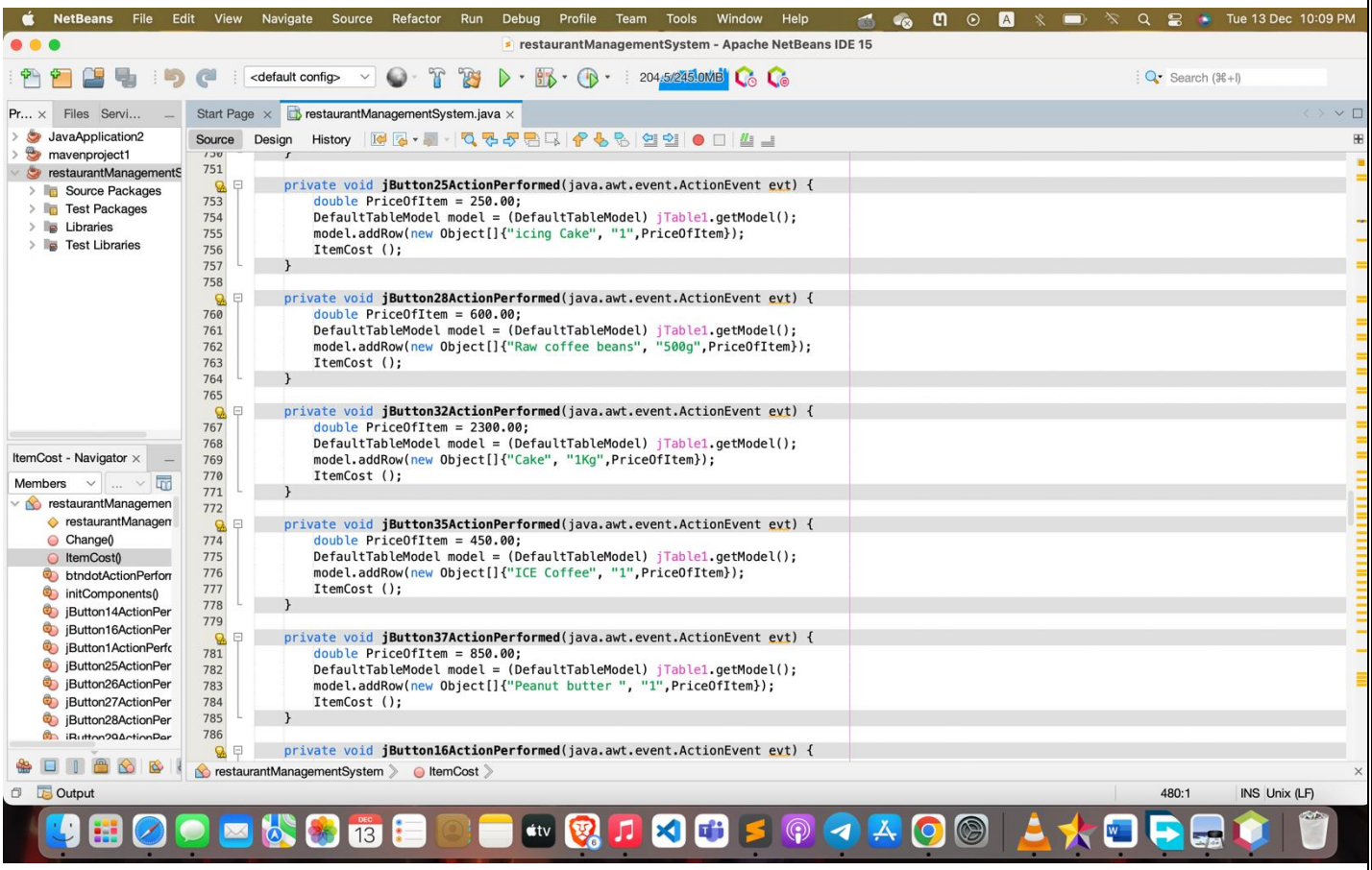

restaurantManagementSystem - Apache NetBeans IDE 15
Start Page
restaurantManagementSystem.java
Source
Design
History
private void jButton25ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 250.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"icing Cake", "1",PriceOfItem});
ItemCost ();
}
private void jButton28ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 600.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Raw coffee beans", "500g",PriceOfItem});
ItemCost ();
}
private void jButton32ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 2300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Cake", "1Kg",PriceOfItem});
ItemCost ();
}
private void jButton35ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 450.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"ICE Coffee", "1",PriceOfItem});
ItemCost ();
}
private void jButton37ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 850.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Peanut butter ", "1",PriceOfItem});
ItemCost ();
}
private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {
ItemCost - Navigator
Members
Change()
ItemCost()
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

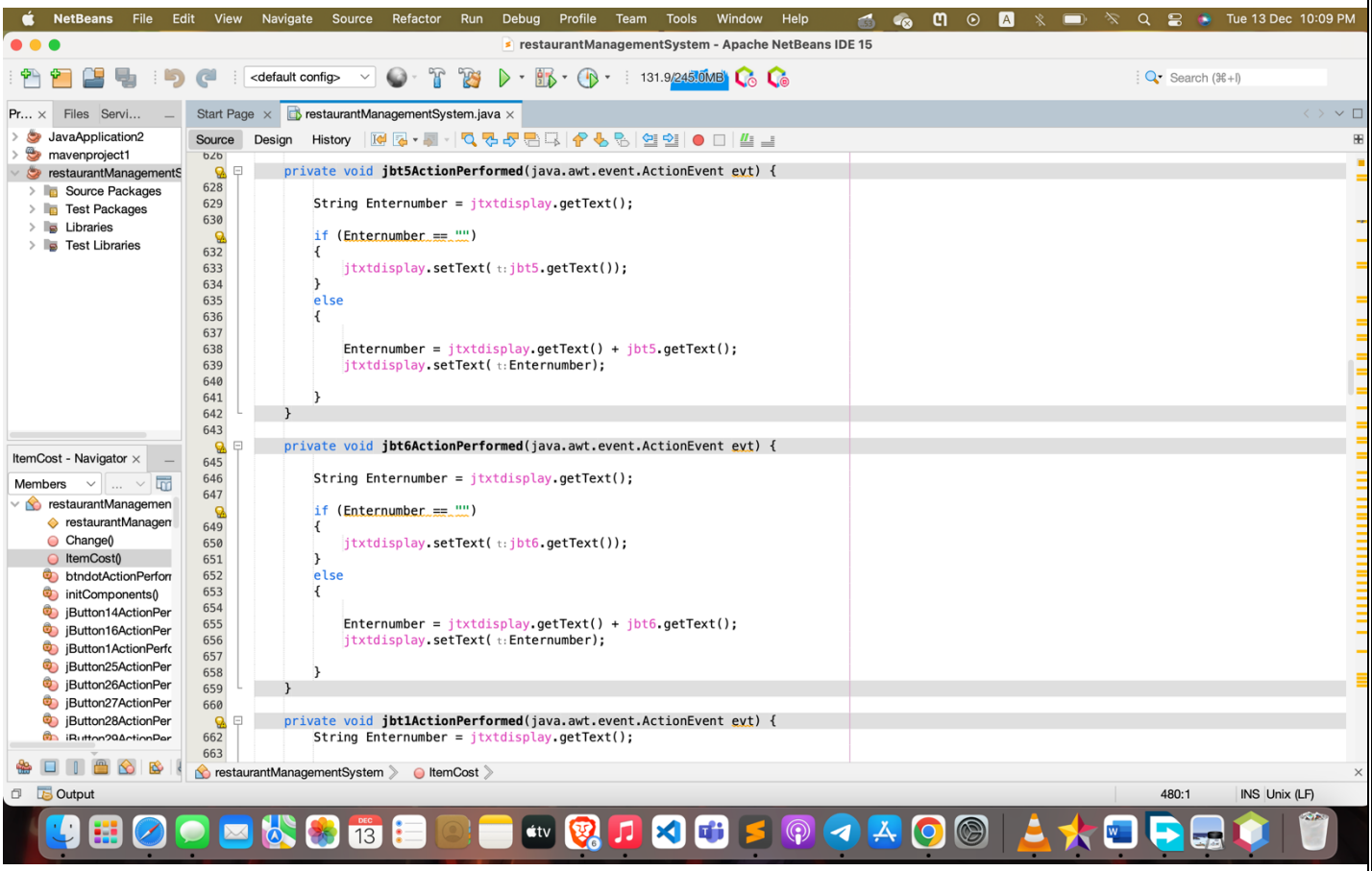

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
Start Page
restaurantManagementSystem.java
Source Design History
private void jbt0ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
jtxtdisplay.setText(t: jbt0.getText());
else
Enternumber = jtxtdisplay.getText() + jbt0.getText();
jtxtdisplay.setText(t: Enternumber);
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
jtxtdisplay.setText (t: "");
jtxtChange.setText (t: "");
private void btndotActionPerformed(java.awt.event.ActionEvent evt) {
if(! jtxtdisplay.getText () .contains(s: "."))
jtxtdisplay.setText(jtxtdisplay.getText() + btndot.getText());
private void jbtn01ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Orange Juice", "1",PriceOfItem});
ItemCost - Navigator
Members
restaurantManagemen
Change()
ItemCost()
initComponents()
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

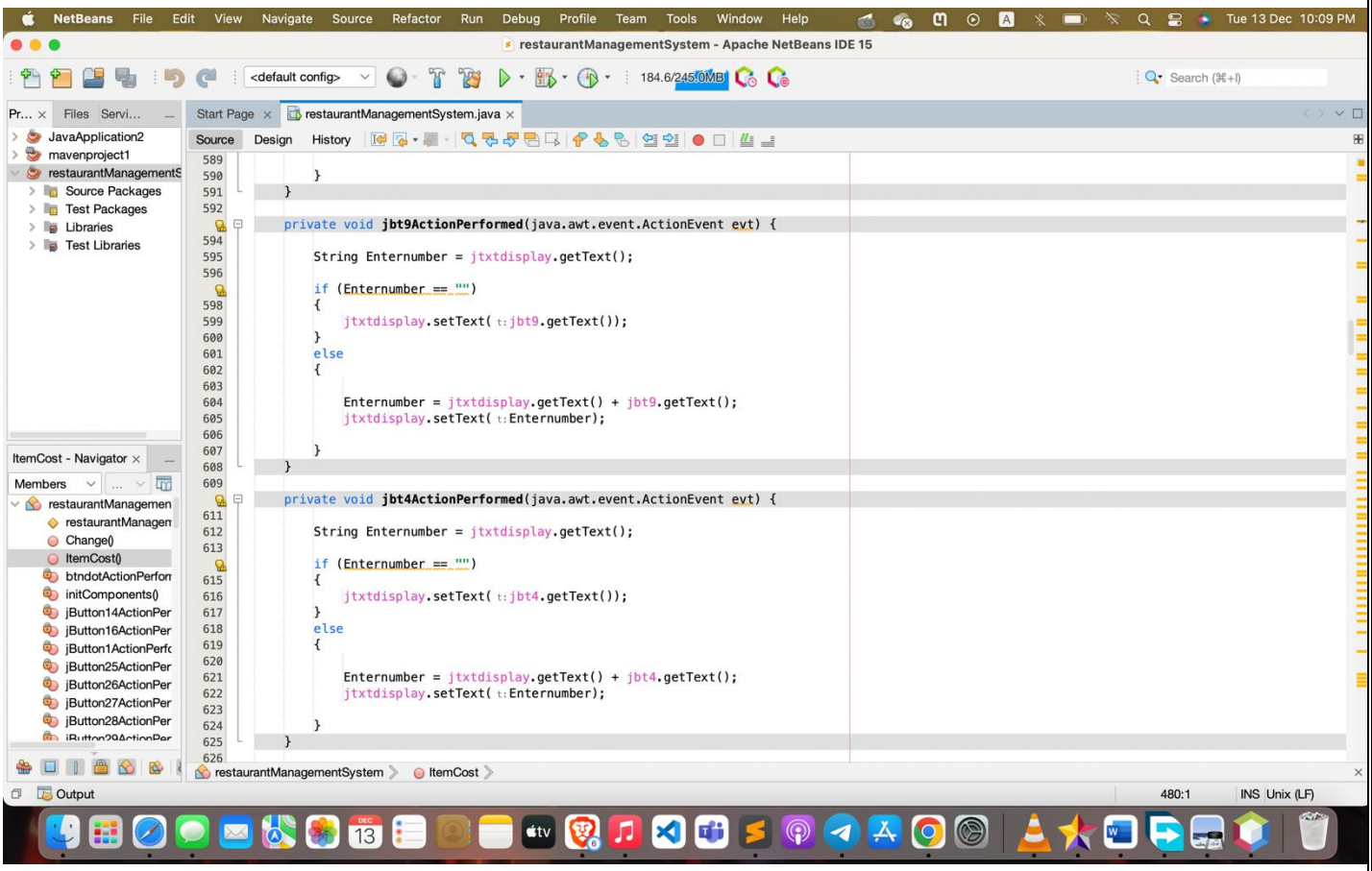

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
Start Page
restaurantManagementSystem.java
Source Design History
private void jbt1ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
jtxtdisplay.setText(t: jbt1.getText());
else
Enternumber = jtxtdisplay.getText() + jbt1.getText();
jtxtdisplay.setText(t: Enternumber);
private void jbt2ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
jtxtdisplay.setText(t: jbt2.getText());
else
Enternumber = jtxtdisplay.getText() + jbt2.getText();
jtxtdisplay.setText(t: Enternumber);
private void jbt3ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
ItemCost - Navigator
Change()
ItemCost()
initComponents()
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

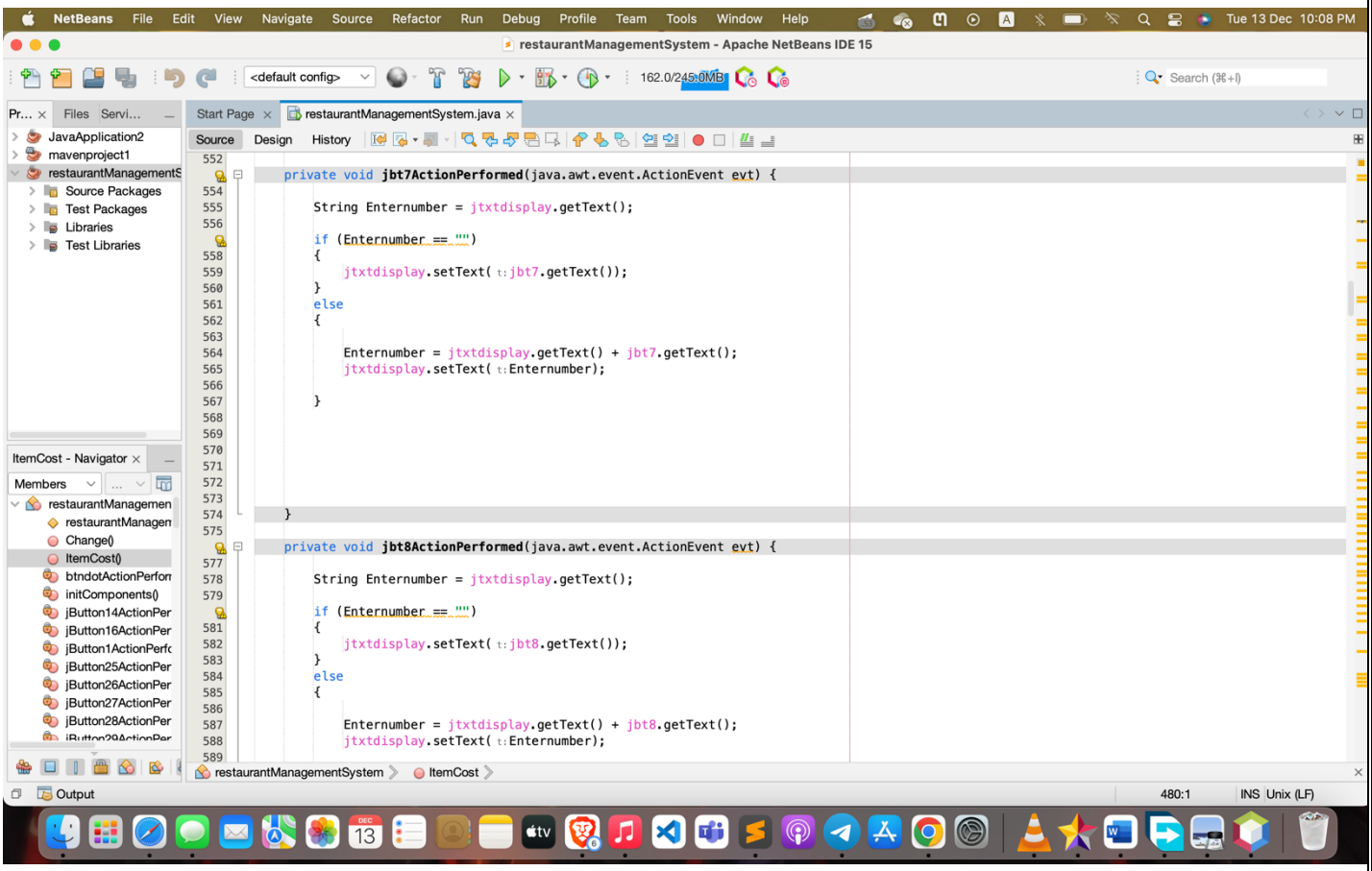

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
Start Page
restaurantManagementSystem.java
Source Design History
private void jbt5ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t: jbt5.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt5.getText();
jtxtdisplay.setText(t: Enternumber);
}
}
private void jbt6ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t: jbt6.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt6.getText();
jtxtdisplay.setText(t: Enternumber);
}
}
private void jbt1ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

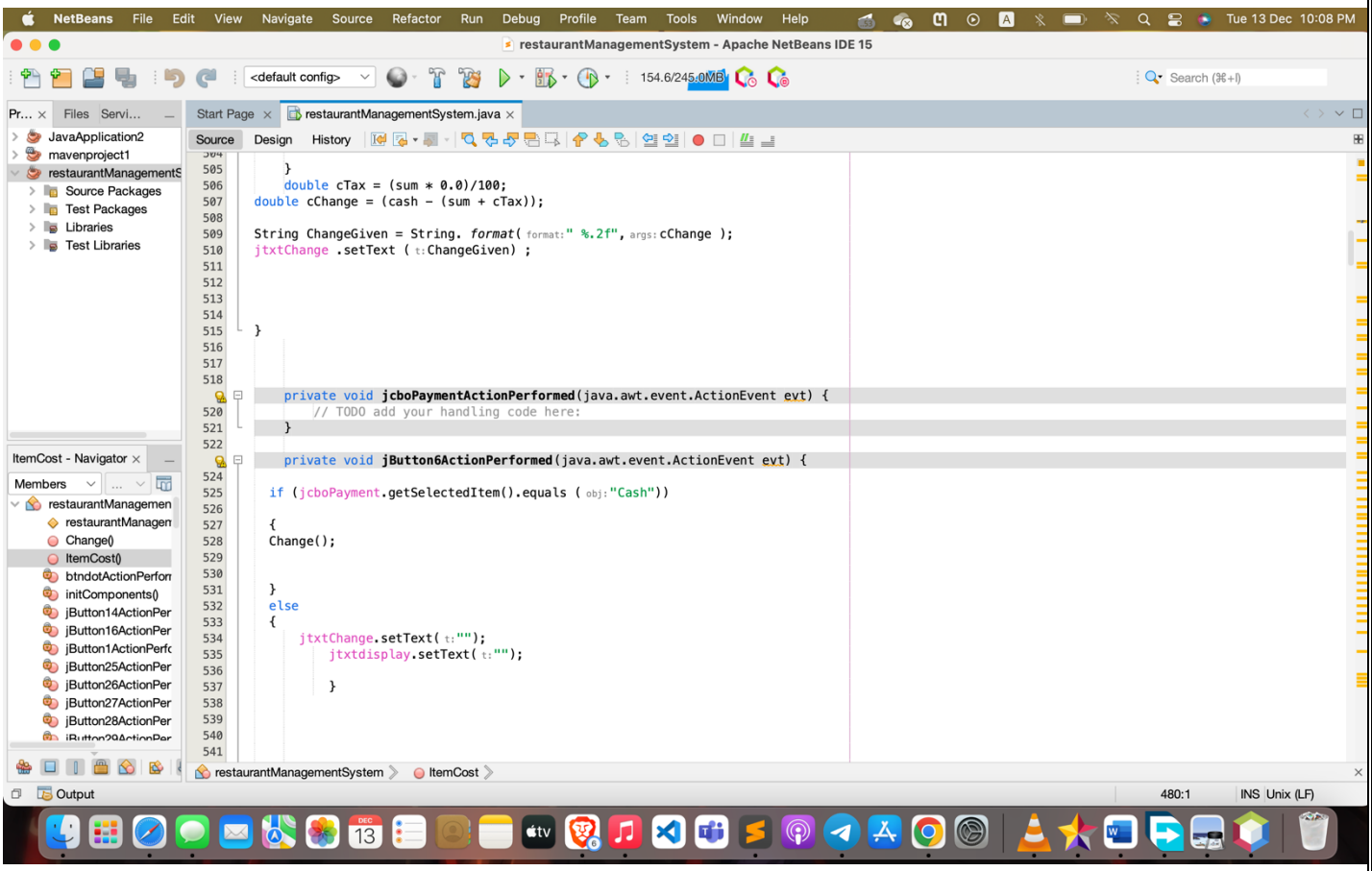

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jButton26ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Juice ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton29ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"chocolate Cake ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton31ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 80.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"black coffee ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton34ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 450.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"soft drinks ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton38ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 50.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Black Tea ", "1 ",PriceOfItem});
ItemCost ();
}
private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 250.00;
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

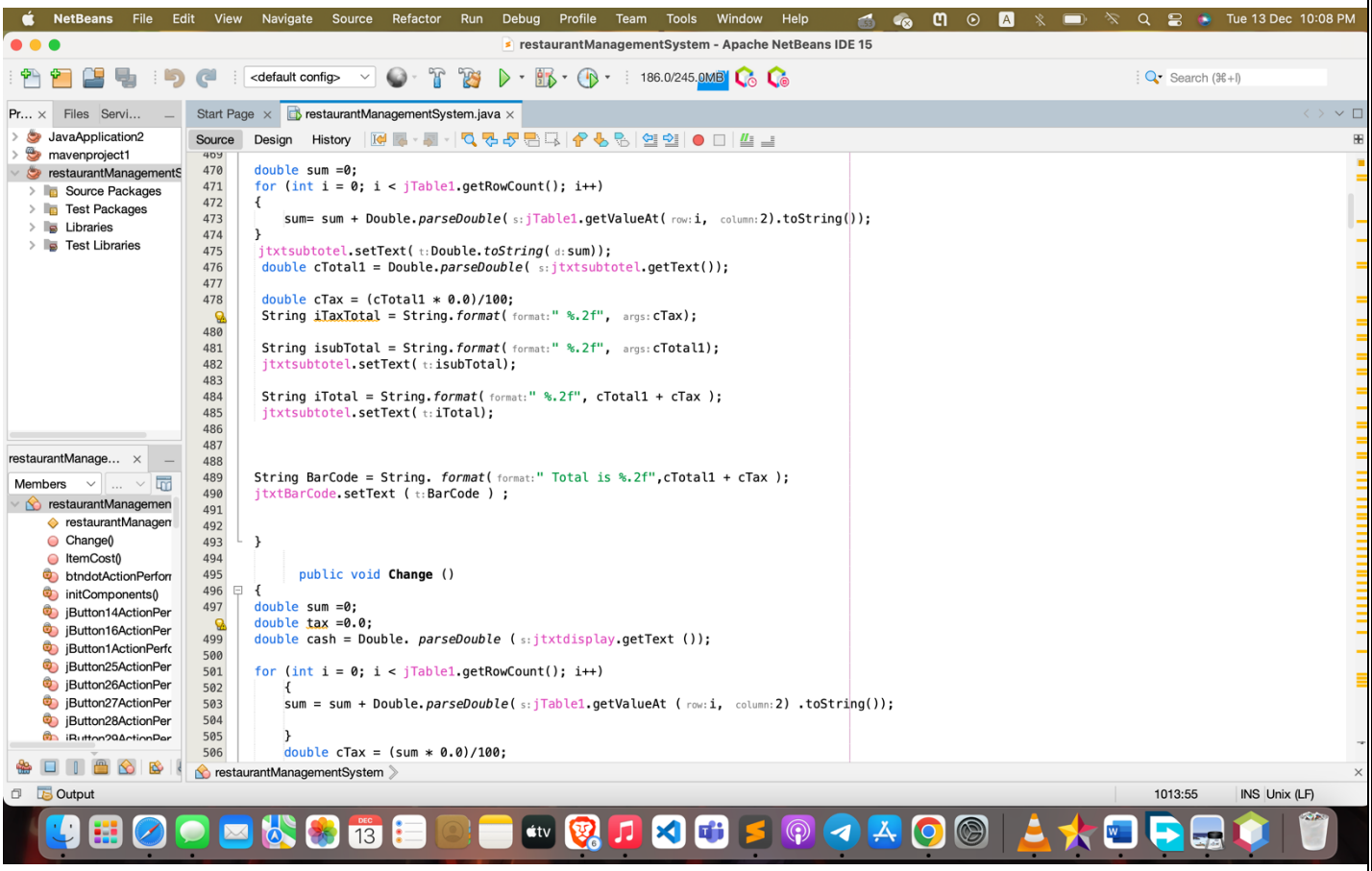

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfc
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jButton25ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 250.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"icing Cake", "1",PriceOfItem});
ItemCost ();
}
private void jButton28ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 600.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Raw coffee beans", "500g",PriceOfItem});
ItemCost ();
}
private void jButton32ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 2300.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Cake", "1Kg",PriceOfItem});
ItemCost ();
}
private void jButton35ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 450.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"ICE Coffee", "1",PriceOfItem});
ItemCost ();
}
private void jButton37ActionPerformed(java.awt.event.ActionEvent evt) {
double PriceOfItem = 850.00;
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
model.addRow(new Object[]{"Peanut butter ", "1",PriceOfItem});
ItemCost ();
}
private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)


NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
131.9/245.0MB
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
private void jbt5ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt5.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt5.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jbt6ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
if (Enternumber == "")
{
jtxtdisplay.setText(t:jbt6.getText());
}
else
{
Enternumber = jtxtdisplay.getText() + jbt6.getText();
jtxtdisplay.setText(t:Enternumber);
}
}
private void jbt1ActionPerformed(java.awt.event.ActionEvent evt) {
String Enternumber = jtxtdisplay.getText();
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

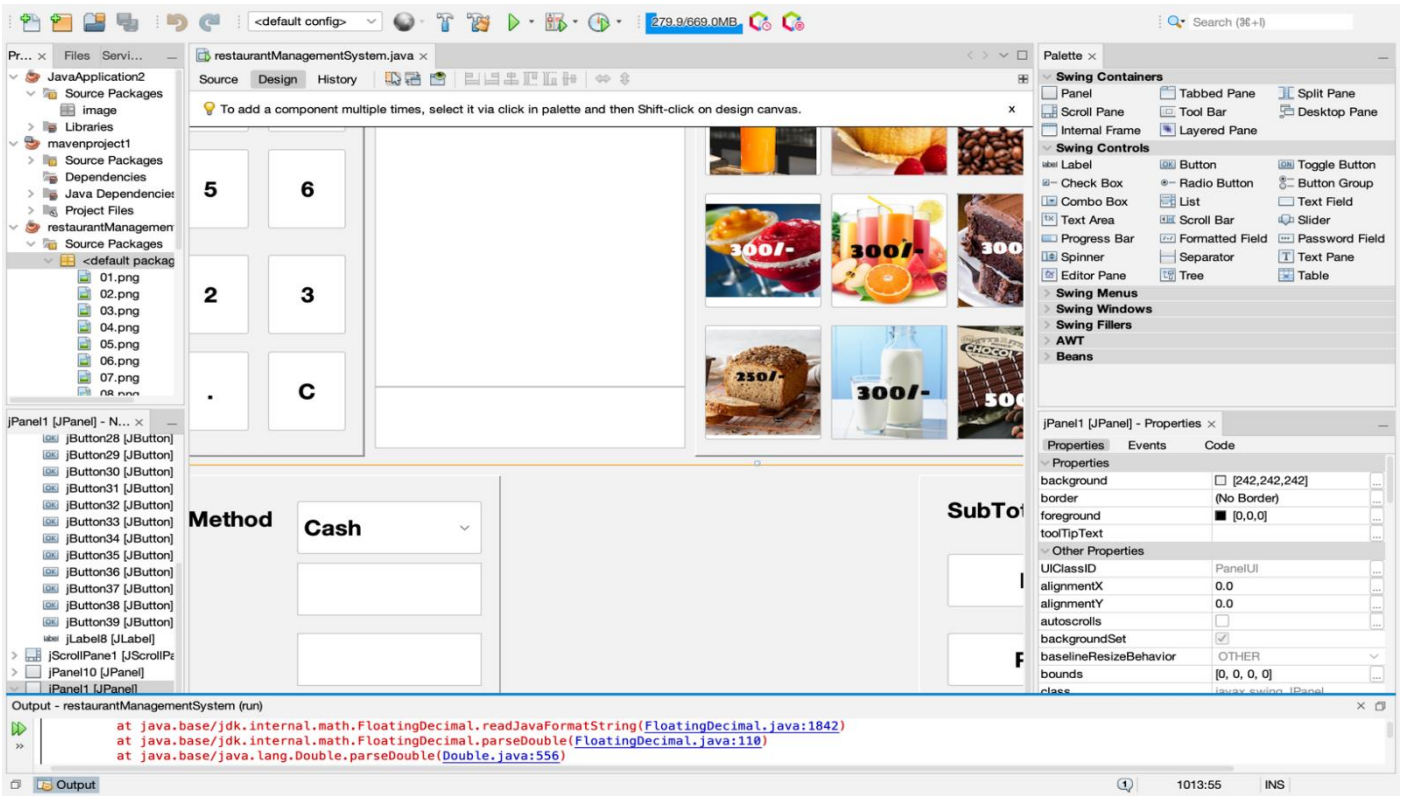

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:09 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
184.6/245.0MB
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
589 }
590 }
591
592 private void jbt9ActionPerformed(java.awt.event.ActionEvent evt) {
594 String Enternumber = jtxtdisplay.getText();
596 if (Enternumber == "")
598 {
599 jtxtdisplay.setText(t:jbt9.getText());
600 }
601 else
602 {
603
604 Enternumber = jtxtdisplay.getText() + jbt9.getText();
605 jtxtdisplay.setText(t:Enternumber);
606
607 }
608 }
609
610 private void jbt4ActionPerformed(java.awt.event.ActionEvent evt) {
611
612 String Enternumber = jtxtdisplay.getText();
613
615 if (Enternumber == "")
616 {
617 jtxtdisplay.setText(t:jbt4.getText());
618 }
619 else
620 {
621
622 Enternumber = jtxtdisplay.getText() + jbt4.getText();
623 jtxtdisplay.setText(t:Enternumber);
624
625 }
626 }
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

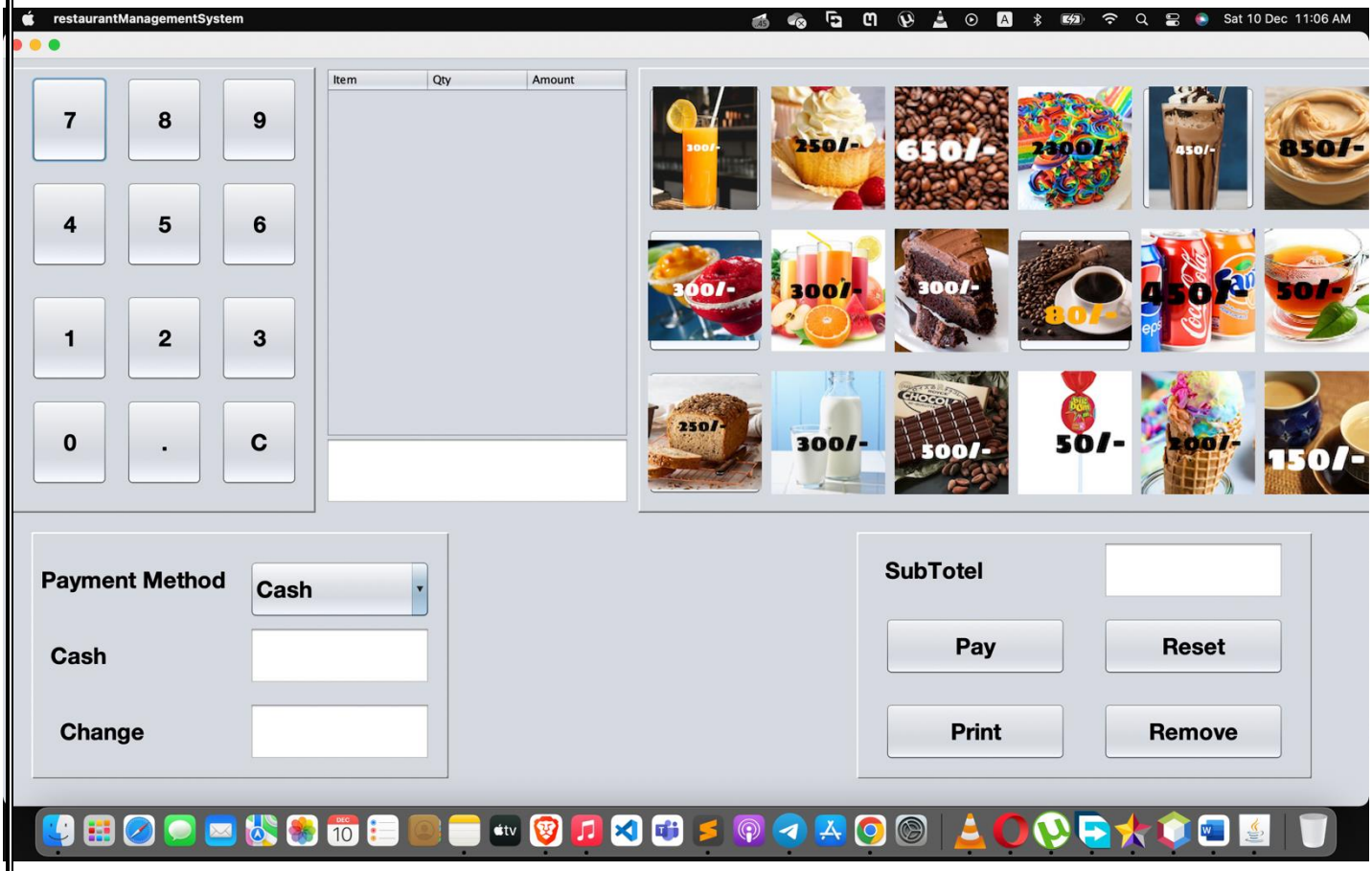

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:08 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
162.0/245.0MB
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
552 private void jbt7ActionPerformed(java.awt.event.ActionEvent evt) {
554 String Enternumber = jtxtdisplay.getText();
555
556 if (Enternumber == "")
558 {
559 jtxtdisplay.setText(t:jbt7.getText());
560 }
561 else
562 {
563
564 Enternumber = jtxtdisplay.getText() + jbt7.getText();
565 jtxtdisplay.setText(t:Enternumber);
566
567 }
568
569
570
571
572
573
574 }
575
577 private void jbt8ActionPerformed(java.awt.event.ActionEvent evt) {
578 String Enternumber = jtxtdisplay.getText();
579
581 if (Enternumber == "")
582 {
583 jtxtdisplay.setText(t:jbt8.getText());
584 }
585 else
586 {
587 Enternumber = jtxtdisplay.getText() + jbt8.getText();
588 jtxtdisplay.setText(t:Enternumber);
589
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)


NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:08 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
154.6/245.0MB
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
ItemCost - Navigator
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
}
double cTax = (sum * 0.0)/100;
double cChange = (cash - (sum + cTax));
String ChangeGiven = String. format(format:" %.2f", args:cChange);
jtxtChange .setText (t:ChangeGiven) ;
}
private void jcboPaymentActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
if (jcboPayment.getSelectedItem().equals (obj:"Cash"))
{
Change();
}
else
{
jtxtChange.setText(t:"");
jtxtdisplay.setText(t:"");
}
restaurantManagementSystem
ItemCost
Output
480:1
INS Unix (LF)

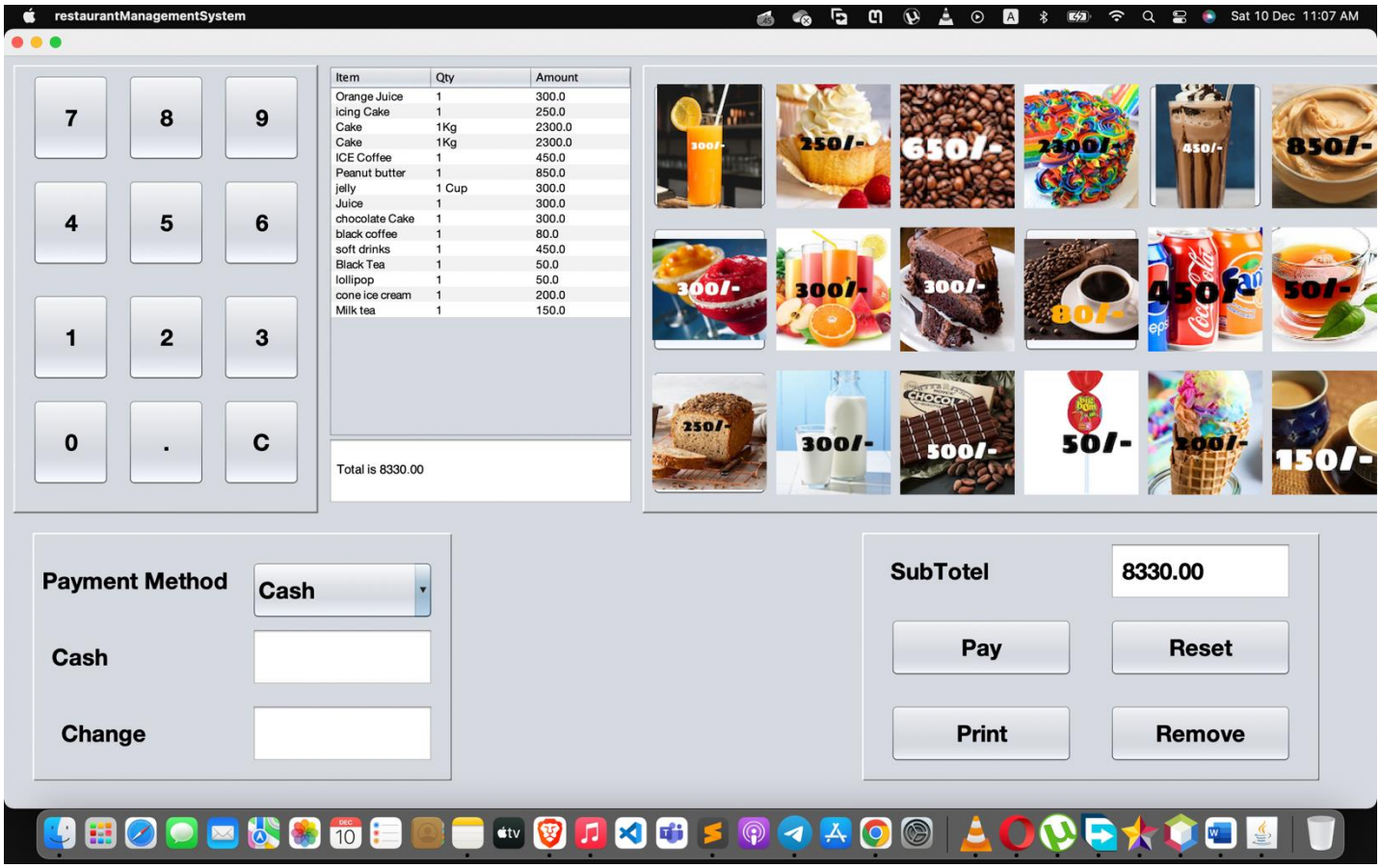

NetBeans File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Tue 13 Dec 10:08 PM
restaurantManagementSystem - Apache NetBeans IDE 15
<default config>
186.0/245.0MB
Search (⌘+I)
Start Page
restaurantManagementSystem.java
Source Design History
JavaApplication2
mavenproject1
restaurantManagementS
Source Packages
Test Packages
Libraries
Test Libraries
Members
restaurantManagemen
restaurantManagen
Change()
ItemCost()
btndotActionPerfor
initComponents()
jButton14ActionPer
jButton16ActionPer
jButton1ActionPerfo
jButton25ActionPer
jButton26ActionPer
jButton27ActionPer
jButton28ActionPer
double sum =0;
for (int i = 0; i < jTable1.getRowCount(); i++)
{
sum= sum + Double.parseDouble(s:jTable1.getValueAt(row:i, column:2).toString());
}
jtxtsubtotel.setText(t:Double.toString(d:sum));
double cTotal1 = Double.parseDouble(s:jtxtsubtotel.getText());
double cTax = (cTotal1 * 0.0)/100;
String iTaxTotal = String.format(format:" %.2f", args:cTax);
String isubTotal = String.format(format:" %.2f", args:cTotal1);
jtxtsubtotel.setText(t:isubTotal);
String iTotal = String.format(format:" %.2f", cTotal1 + cTax);
jtxtsubtotel.setText(t:iTotal);
String BarCode = String. format(format:" Total is %.2f",cTotal1 + cTax);
jtxtBarCode.setText (t:BarCode) ;
}
public void Change ()
{
double sum =0;
double tax =0.0;
double cash = Double. parseDouble (s:jtxtdisplay.getText ());
for (int i = 0; i < jTable1.getRowCount(); i++)
{
sum = sum + Double.parseDouble(s:jTable1.getValueAt (row:i, column:2) .toString());
}
double cTax = (sum * 0.0)/100;
restaurantManagementSystem
Output
1013:55
INS Unix (LF)

10. PROJECT GUI





Back to content....


restaurantManagementSystem
Sat 10 Dec 11:07 AM
7
8
9
4
5
6
1
2
3
0
.
C
Item Qty Amount
Orange Juice 1 300.0
icing Cake 1 250.0
Cake 1Kg 2300.0
Cake 1Kg 2300.0
ICE Coffee 1 450.0
Peanut butter 1 850.0
jelly 1 Cup 300.0
Juice 1 300.0
chocolate Cake 1 300.0
black coffee 1 80.0
soft drinks 1 450.0
Black Tea 1 50.0
lollipop 1 50.0
cone ice cream 1 200.0
Milk tea 1 150.0
Total is 8330.00
Payment Method
Cash
Cash
Change
SubTotel
8330.00
Pay
Reset
Print
Remove


restaurantManagementSystem
Sat 10 Dec 11:07 AM
7
8
9
4
5
6
1
2
3
0
.
C
Item Qty Amount
Orange Juice 1 300.0
icing Cake 1 250.0
Cake 1Kg 2300.0
Cake 1Kg 2300.0
ICE Coffee 1 450.0
Peanut butter 1 850.0
jelly 1 Cup 300.0
Juice 1 300.0
chocolate Cake 1 300.0
black coffee 1 80.0
soft drinks 1 450.0
Black Tea 1 50.0
lollipop 1 50.0
cone ice cream 1 200.0
Milk tea 1 150.0
Total is 8330.00
Payment Method
Cash
Cash
100000
Change
91670.00
SubTotel
8330.00
Pay
Reset
Print
Remove

11. Conclusion & Future Work

After undergoing a 1.5 months of summer time education at E-Global Consultancy, I learnt diverse principles of Oops in Java and a gui based software using java. The period of my schooling become from twenty eighth June 2018 to 8th August 2018. I turned into worried inside the place of java Programming for the duration of my work time period, all of that allows you to be outlined in this file. There is a lot of latest expertise that can be discovered and I got to recognize altogether on how this organization performs an vital role in industrial area.

Exposure that have been given to me by using employer's body of workers about the working and technical thing is a very meaningful knowledge to me in an effort to put together myself before moving into the actual work environment on the imminent days. The group of workers on the employer may be very useful and friendly. They gave me quite a few exposure at the terms of reference and tactics related to the system of my work.

Other than that, the publicity travelling the enterprise's ecosystem gave a beneficial expertise to me. Exposure that had been given to me at this corporation can offer the image on a actual-existence scenario, the undertaking and duty that would be carried through some humans on the sector. Lastly, college students involvement in summer season education like this might show and similarly beef up student's identification in undergoing schooling in technical discipline..

Apart from that, the layout record that wishes to be executed by way of students after undergoing commercial education also can teach every of the student in preparing technical document that is whole, compact and in a right order that can be made as an crucial know-how after they face a actual state of affairs later. I have protected some historical past records on the tasks.

I turned into involved in, in addition to info on how the projects have been advanced in this report Throughout the economic schooling, I discovered that numerous things are essential:

- **Critical and Analytical Thinking**

  To organise our tasks and challenge, we want to examine our problems and challenge, and to formulate a very good way to the problem. We could must set contingency plan for the solution, in order that we're properly organized for the unforeseeable conditions.

- **Time Management**

  As usual technician and programmer are constantly racing in opposition to tight timeline and packed agenda, a right time control will minimise dealing with late closing dates. An powerful time control allows us to do our challenge successfully and meet our schedules. Scheduling avoids time wastage and permits us to plan in advance, and gaining extra as a result.

- **Goal Management**

  Opposing to a Herculean intention appeared to be reachable at the start sight, it's miles higher to sub-divide the desires to three plausible tasks, so that we can be gaining extra self-assurance by engaging in those obligations.

- **Colleague Interactions**

  In operating environment, teamwork is essential in contributing to a robust enterprise. Teamwork is likewise vital in accomplishing the dreams of the corporation as an entity. Thus, speaking and sharing is lots wished in the operating environment. Therefore, we should be respecting each different in paintings, and running together as a crew, rather than running by myself. This is because working together as a group is less complicated in accomplishing our targets, in place of working in my view.

12. References

- ✓ https://developers.goole .com/java/
- ✓ https://stackoverflow.com/
- ✓ https://github.com/
- ✓ https://www.youtube.com/
- ✓ https://material.io/icons/
- ✓ https://www.flaticon.com/
- ✓ https://www.google.co.in

Back to content....

Group Members

1. SEU.IS.19.ICT.059-BJF.JATHEELA(Group leader)
2. SEU.IS.19.ICT.079 - S. HANA HASMATH
3. SEU.IS.19.ICT.058 -HIRUNI HANSIKA
4. SEU.IS.19.ICT.033 - AMP.KAVEEN
5. SEU.IS.19.ICT.027 - HC. DILIPUN

Group Work Parts:

Source Code: ***BJF.JATHEELA***
Report: ***S. HANA HASMATH, HIRUNI HANSIKA***
Report Editing: ***AMP.KAVEEN, HC. DILIPUN***
Report Reediting: ***BJF.JATHEELA***

Back to content....