Laboratory Manual

For

Software Design and Engineering

(MF 104)

M.Tech (IT)

SEM I

June 2016

Faculty of Technology
Dharmsinh Desai University
Nadiad.
www.ddu.ac.in

Table of Contents

Sample experiment

1 AIM: To perform the user's view analysis: Use case diagram for the coffee-maker.

The Computer Science department at NCSU has a new building on Centennial Campus. We all know that computer scientists love caffeine, so the CSC department is planning on installing a Coffeemaker in a lounge across the hall from the 24-hour computer lab. Our job is to test and model the functionality of the Coffeemaker. We are only working with the logic code behind the hardware, so only a command line interface is used for manual testing. Here is a partial listing of requirements for the Coffeemaker system.

2 TOOLS/APPARATUS: Netbeans 6.0, Microsoft Visio.

3 STANDARD PROCEDURES:

3.1 Analyzing the Problem:

According to the analysis of **nouns** and **verbs** we can have the following actors and use-cases for the given system:

Actors(based on the nouns)	Use-cases(based on the verbs)
User(The person who makes coffee)	Waiting State
	Add a Recipe
	Edit a Recipe
	Delete a Recipe
	Check Inventory
	Add Inventory
	Purchase Beverage

The use-cases based on the requirements stated in the problem are given in the designing part of the solution.

3.2 Designing the Solution:

Requirements - User Stories

Title: Waiting State		
AccTest: checkOptions0	Priority: 1	Story Points: 2
When the Coffee Maker is not in use it waits for user input. There are six different options of user input: 1) add recipe, 2) delete a recipe, 3) edit a recipe, 4) add inventory, 5) check inventory, and 6) purchase beverage.		

Title: Add a Recipe		
AccTest: addRecipe1	Priority: 1	Story Points: 2
Only three recipes may be added to the Coffeemaker. A recipe consists of a name, price, and units of coffee, units of milk, units of sugar, and units of chocolate. Each recipe name must be unique in the recipe list. Price must be handled as an integer. A status message is printed to specify if the recipe was successfully added or not. Upon completion, the Coffeemaker is returned to the waiting state.		

Title: Delete a Recipe		
AccTest: deleteRecipe1	Priority: 2	Story Points: 1
A recipe may be deleted from the Coffeemaker if it exists in the list of recipes in the Coffeemaker. The recipes are listed by their name. Upon completion, a status message is printed and the Coffee Maker is returned to the waiting state.		

Title: Edit a Recipe		
AccTest: editRecipe1	Priority: 2	Story Points: 1
A recipe may be edited in the Coffeemaker if it exists in the list of recipes in the Coffeemaker. The recipes are listed by their name. After selecting a recipe to edit, the user will then enter the new recipe information. A recipe name may not be changed. Upon completion, a status message is printed and the Coffee Maker is returned to the waiting state.		

Title: Add Inventory		
AccTest: addInventory 1	Priority: 1	Story Points: 2

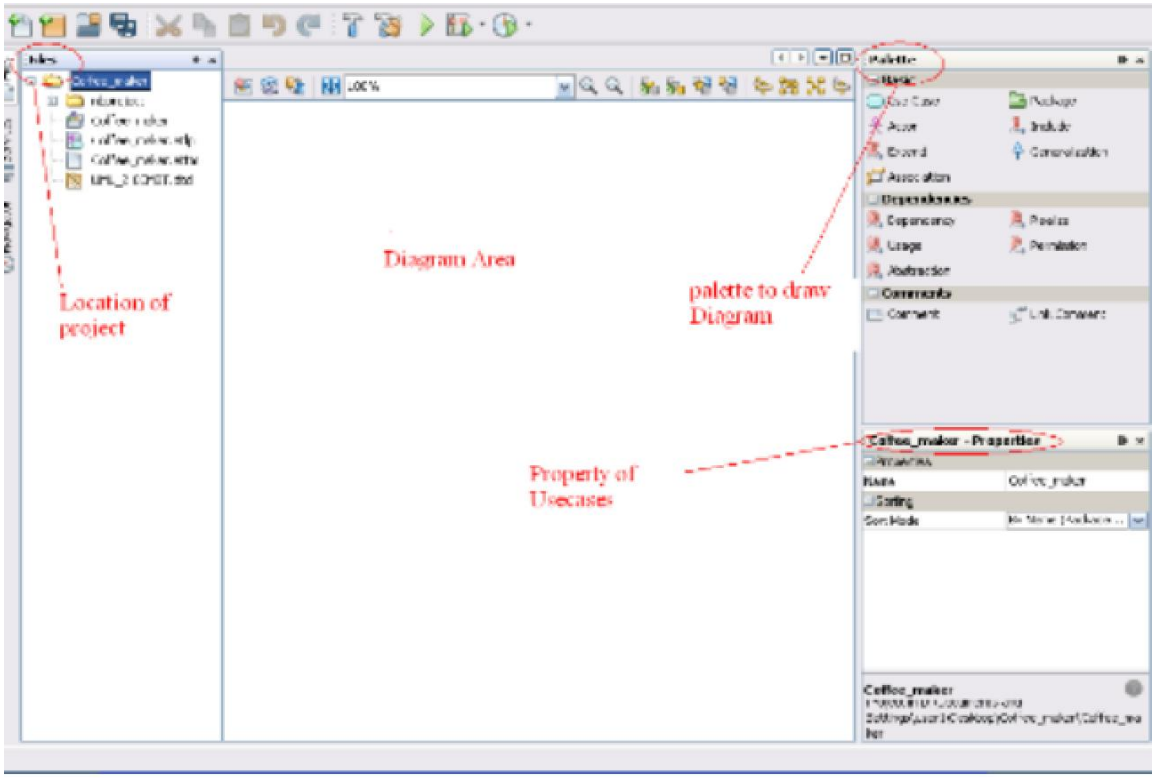
Inventory may be added to the machine at any time from the main menu, and is added to the current inventory in the Coffeemaker. The types of inventory in the Coffeemaker are coffee, milk, sugar, and chocolate. The inventory is measured in integer units. Inventory may only be removed from the Coffeemaker by purchasing a beverage. Upon completion, a status message is printed and the Coffeemaker is returned to the waiting state.

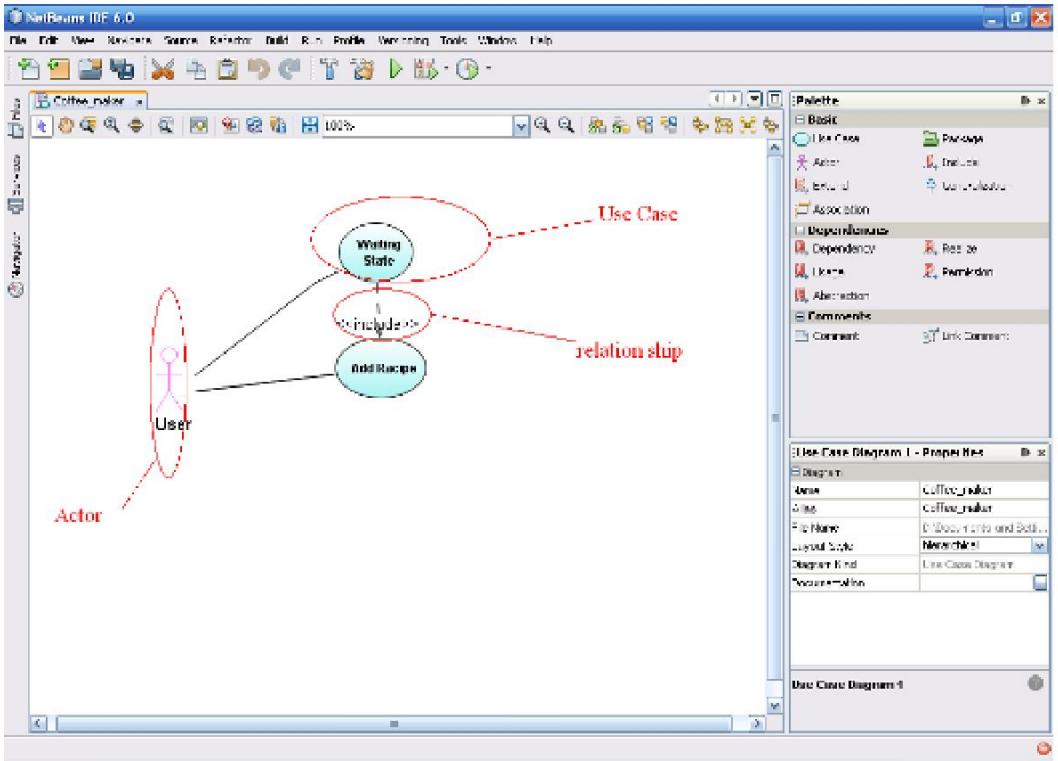
Title: Check Inventory		
AccTest: checkInventory	Priority: 2	Story Points: 1
Inventory may be checked at any time from the main menu. The units of each item in the inventory are displayed. Upon completion, the Coffee Maker is returned to the waiting state.		

Title: Purchase Beverage		
AccTest: purchaseBeverage1	Priority: 1	Story Points: 2
The user selects a beverage and inserts an amount of money. The money must be an integer. If the beverage is in the RecipeBook and the user paid enough money the beverage will be dispensed and any change will be returned. The user will not be able to purchase a beverage if they do not deposit enough money into the Coffeemaker. A user's money will be returned if there is not enough inventories to make the beverage. Upon completion, the Coffee Maker displays a message about the purchase status and is returned to the main menu.		

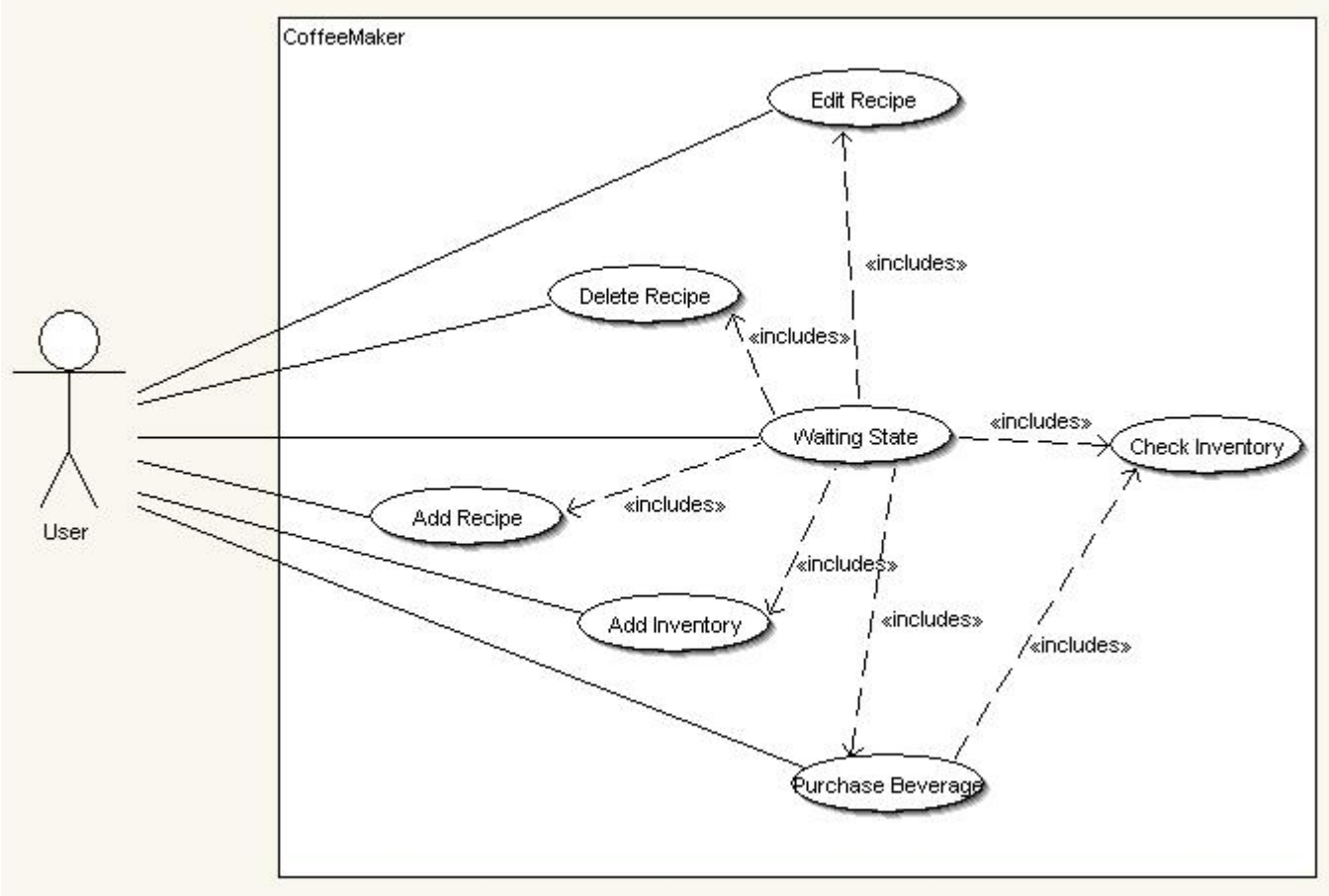
3.3 Implementing the Solution

The use-case diagram can be drawn using the netbeans version 6.0.





Requirements - Use Cases



3.4 Testing the Solution

Here the problem statement is : CSC department is planning on installing a Coffeemaker in a lounge across the hall from the 24-hour computer lab so we have prepared the use-case accordingly:

Requirements given in the problem statement(based on the verbs)	**Use-cases(based on the verbs)**
When the Coffee Maker is not in use it waits for user input. There are six different options of user input	Waiting State
Only three recipes may be added to the Coffeemaker. A recipe consists of a name, price, and units of coffee, units of milk, units of sugar, and units of chocolate	Add a Recipe
A recipe may be edited in the Coffeemaker if it exists in the list of recipes in the Coffeemaker	Edit a Recipe
A recipe may be deleted from the	Delete a Recipe

Coffeemaker if it exists in the list of recipes in the Coffeemaker	
Inventory may be checked at any time from the main menu	Check Inventory
Inventory may be added to the machine at any time from the main menu, and is added to the current inventory in the Coffeemaker	Add Inventory
The user selects a beverage and inserts an amount of money. The money must be an integer	Purchase Beverage

The use-cases are tested to match up with the requirements stated in the above table and they are complying with the problem statement requirements.

4 Conclusions

The use-case diagram can provide the user's view for designing of the software product. And it can also be tested by matching up the requirements with the use-cases.

EXPERIMENT-1

Aim: Study of visio design tools (online / offline)

Tools/ Apparatus: None.

Procedure:

- Study the various tools (Online / Offline) available for software designing.
- List the tools and study any one tool in detail.
- Prepare the report on the tool selected.

EXPERIMENT-2

Aim: Implementation / Simulation of Object Oriented Concepts using Different Object Oriented Programming Languages.(Make Comparative Study).

Tools/ Apparatus: Turbo C++/ JDK1.6/.Net Framework.

Procedure:

1) Identify the different object oriented concepts.
2) Apply OOP concepts on simple problem.
3) Implement OOP concepts using different OOP languages.

EXPERIMENT-3

Aim: Prepare UML / DFD diagram for library management system.
(Using Different OOM Grady Booch, James Rumbaugh & Ivar Jacobson)

Tools/ Apparatus: Edraw / Microsoft Visio.
Procedure:

- Definition in detail : -
    - The Library Management System is designed & developed for a receipt and issuance of books in the library along with the student's details. The books received in the library are entered in Books Entry form and the new student is entered in the student entry form. When the student wants to get the desired book the same is issued on the availability basis to the student. The issuance and due date for the returning of the book is also entered into the Book Issue form under third menu Book Issue. The student has to pay the fine if any on the basis of no. of days delayed deposit of the book in the library.
- Prepare SRS for the given definition.
- Draw DFD and UML diagrams for it.

EXPERIMENT-4

Aim: Prepare an use case diagram for the given system.

Tools/Apparatus: Edraw / Microsoft Visio.

Procedure:

1) Identify various processes, actors ,use cases of the system.
2) Draw all the notations used in Usecase diagram with its description.
3) Identify the purpose of usecase diagram and specify when it should be used.
4) Use processes at varios levels to draw the usecase diagram.

EXPERIMENT-5

Aim: Prepare the interaction diagram.

- Sequence diagram.
- Collaboration diagram.

Tools/Apparatus: Netbeans 6.0 / Edraw / Microsoft Visio.

Procedure:

- Identify various elements such as controller class, objects, boundaries, messages etc. of the sequence diagram
- Draw all the notations used in sequence and collaboration diagram with its description.
- Identify the purpose of both the diagram and specify when sequence or collaboration diagram should be used.
- Draw sequence and collaboration diagram as per the norms.

EXPERIMENT-6

Aim: Prepare state chart diagram

Tools/Apparatus: Netbeans 6.0 / Edraw / Microsoft Visio.

Procedure:

1) Identify various elements such as states, events, do actions, transitions etc. of the state chart diagram
2) Draw all the notations used in state chart diagram with its description.
3) Identify the purpose state chart diagram and specify when it should be used.
4) Draw state chart diagram as per the norms.

EXPERIMENT-7

Aim: Prepare class diagram with advanced concept (design phase).
Tools/Apparatus: Netbeans 6.0 / Edraw / Microsoft Visio
Procedure:

1) Identify various elements such as classes, relationships, object, attributes, member functions etc. of the class diagram
2) Draw all the notations used in class diagram with its description.
3) Identify the purpose class diagram and specify when it should be used.
4) Draw class diagram as per the norms.

EXPERIMENT-8

Aim: Prepare the activity diagram

Tools/Apparatus: Netbeans 6.0 / Edraw / Microsoft Visio

Procedure:

1) Identify various elements such as different activity their boundaries ,fork and join conditions, decision branches etc. of the activity diagram
2) Draw all the notations used in activity diagram with its description. Identify the purpose of activity diagram and specify when it should be used.
3) Draw activity diagram as per the norms.

EXPERIMENT-9

Aim: Prepare physical diagram

- Component diagram.
- Deployment diagram.

.

Tools/Apparatus: Netbeans 6.0 / Edraw / Microsoft Visio

Procedure:

1) Identify various elements of the componant diagram such as the various components like client, server, network elements etc.
2) Identify the environment where system will be deployed
3) Draw the component and deployment diagram as per the norms.

EXPERIMENT-10

Aim: Implement any one design pattern

Tools/Apparatus: Netbeans 6.0/.Net framework / c++.

Procedure:

1) Identify the purpose of design pattern and in which situation , which pattern should be used
2) Write a program showing working of a design pattern.

EXPERIMENT-11

Aim: To draw UML diagrams using Rational rose software.

Tools/Apparatus: Rational rose software.

Procedure:

1) Identify various elements of the system to be drawn using the IDE.
2) Use the UML options of the rational rose to draw the diagrams from experiment 4 to 10.

EXPERIMENT-12

Aim: Identify different types of performance and regression testing.

Tools/Apparatus: None.

Procedure:

1) Identify different types of testing performed under performance and regression testing
2) Apply these test cases on your project definition.
3) Generate appropriate error report.

References

Reference books:

- Fundamentals of Software engineering
  - Rajib Mall.
- Software design – From programming to architecture
  - Eric Braude
- Object-oriented software engineering – A use case driven approach
  - Ivar Jacobson(Computer language productivity award winner)
- UML DISTILLED, second edition, Prentice Hall of India / Pearson Education, 2002
  - Martin Fowler