

Laboratory Manual

For

Software Engineering

(CT 616)

B.Tech (IT)

SEM VI



June 2010

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

Table of Contents

EXPERIMENT-1

Phases in software development project, overview, need, coverage of topics 7

EXPERIMENT-2

To assign the requirement engineering tasks 8

EXPERIMENT-3

To perform the system analysis : Requirement analysis, SRS 9

EXPERIMENT-4

To perform the function oriented diagram : DFD and Structured chart 10

EXPERIMENT-5

To perform the user's view analysis : Use case diagram 11

EXPERIMENT-6

To draw the structural view diagram : Class diagram, object diagram 12

EXPERIMENT-7

To draw the behavioral view diagram : Sequence diagram, Collaboration diagram 13

EXPERIMENT-8

To draw the behavioral view diagram : State-chart diagram, Activity diagram 14

EXPERIMENT-9

To draw the implementation view diagram: Component diagram 15

EXPERIMENT-10

To draw the environmental view diagram : Deployment diagram 16

EXPERIMENT-11

To perform various testing using the testing tool unit testing, integration testing 17

LABWORK BEYOND CURRICULA

EXPERIMENT-12

To draw UML diagrams using Rational rose software 18

EXPERIMENT-13

To draw UML diagrams using Microsoft Visio software 19

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

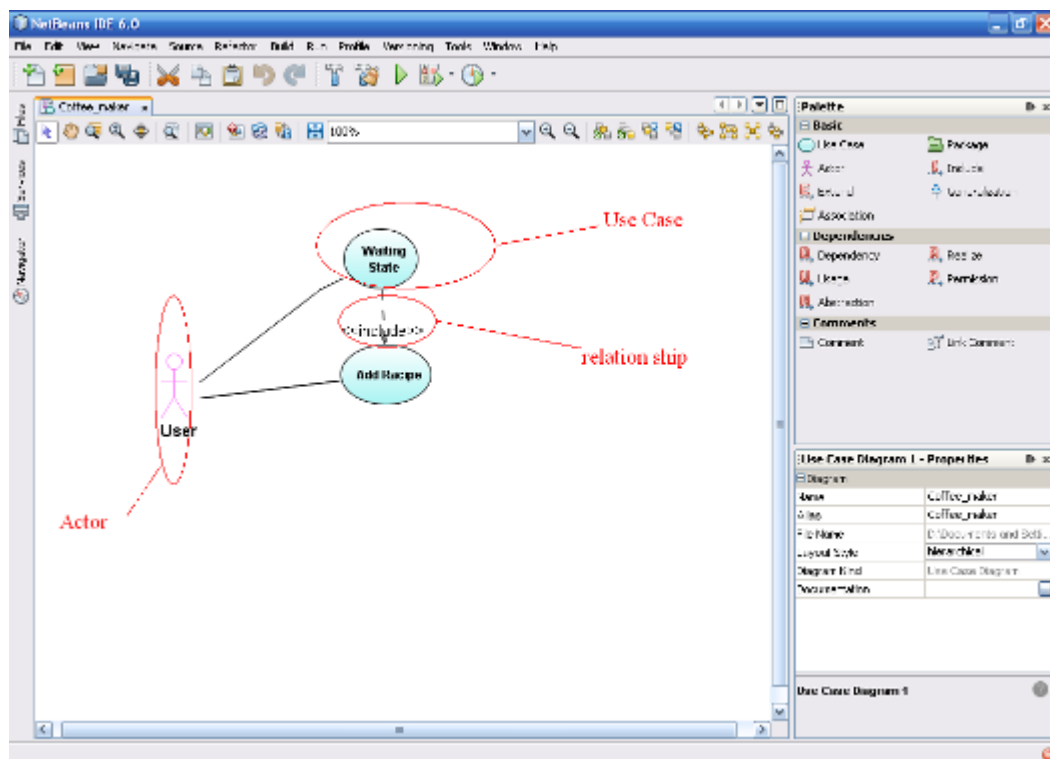
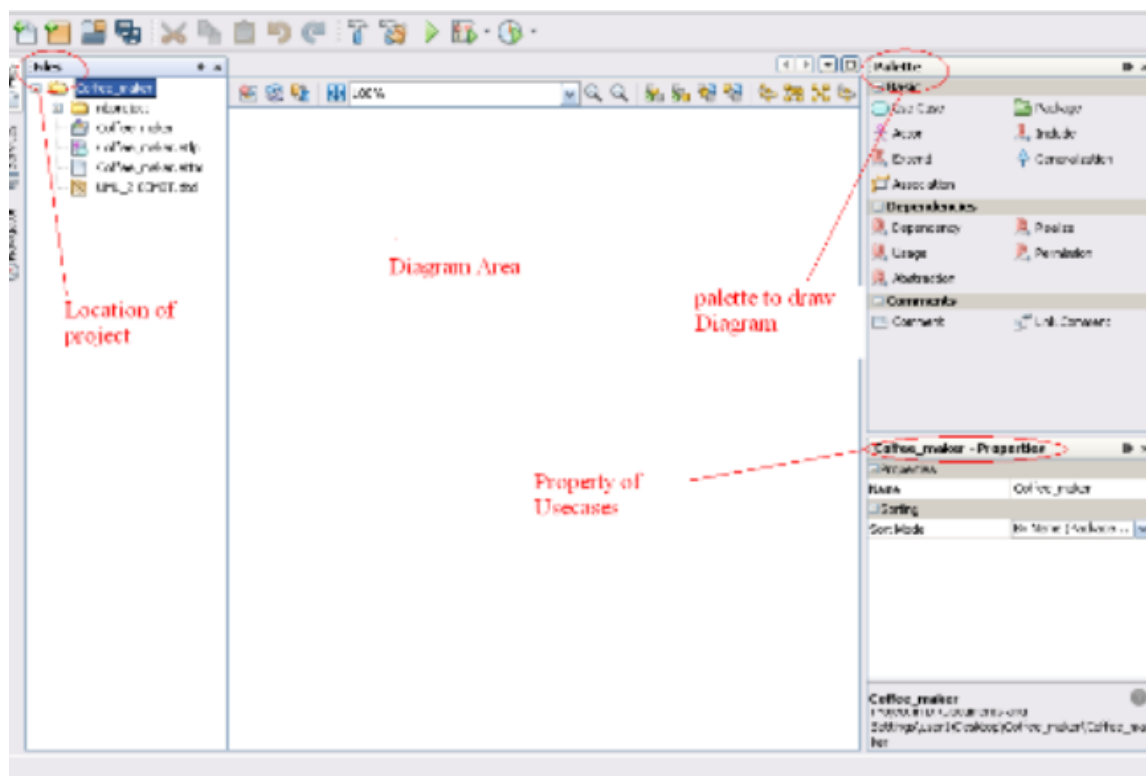
Priority: 1

Story Points: 2

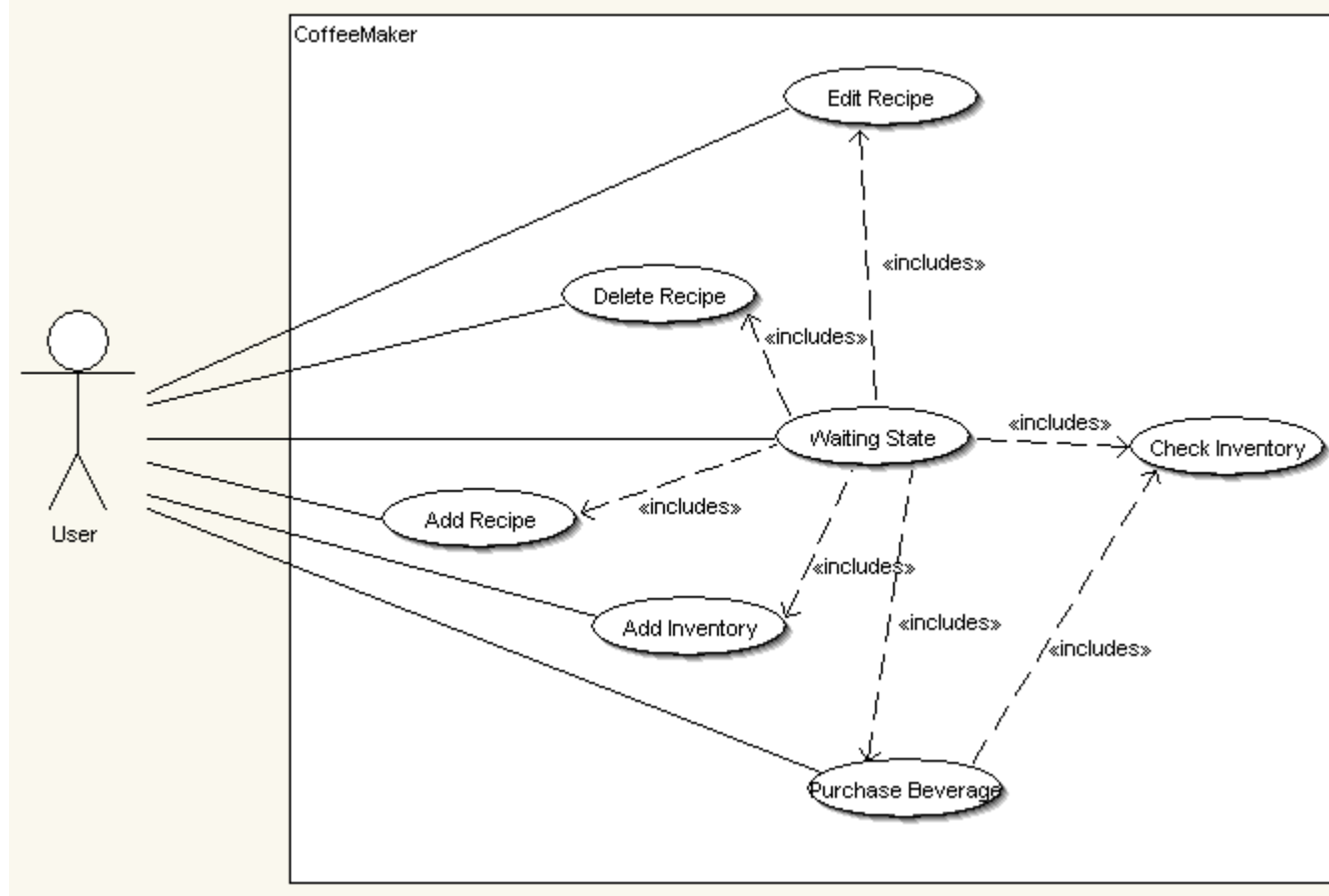
The user selects a beverage and inserts an amount of money. The money must be an integer. If the beverage is in the RecipeBook and the user paid enough money the beverage will be dispensed and any change will be returned. The user will not be able to purchase a beverage if they do not deposit enough 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: Phases in software development project, overview, need, coverage of topics

Tools/ Apparatus: None.

Procedure:

- 1) Open an appropriate software engineering guide and study the software development life cycle and related topics.
- 2) Study the need of the software engineering.
- 3) Study the coverage of topics such as life cycle models and their comparisons.

EXPERIMENT-2

Aim: To assign the requirement engineering tasks.

Tools/ Apparatus: None.

Procedure:

- 1) Identify the different requirement engineering tasks.
- 2) Assign these tasks to various students to set the ball rolling.
- 3) Ask the students to start working on the given tasks.

EXPERIMENT-3

Aim: To perform the system analysis : Requirement analysis, SRS

Tools/ Apparatus: None.

Procedure:

- 1) Assign the group of the students different tasks of system analysis.
- 2) Ask students to meet different users and start analysis the requirements.
- 3) Ask students to give presentations group-wise of their system requirements analysis.

EXPERIMENT-4

Aim: To perform the function oriented diagram : DFD and Structured chart

Tools/Apparatus: Netbeans 6.0.

Procedure:

- 1) Identify various processes, data store, input, output etc. of the system and ask students to analyse.
- 2) Use processes at various levels to draw the DFDs.
- 3) Identify various modules, input, output etc. of the system and ask students to analyse.
- 4) Use various modules to draw Structured charts.

EXPERIMENT-5

Aim: To perform the user's view analysis : Use case diagram

Tools/Apparatus: Netbeans 6.0.

Procedure:

- 1) Identify various processes, use-cases, actors etc. of the system and ask students to analyse.
- 2) Use processes at various levels to draw the use-case diagram.

EXPERIMENT-6

Aim: To draw the structural view diagram : Class diagram, object diagram

Tools/Apparatus: Netbeans 6.0.

Procedure:

- 1) Identify various elements such as classes, member variables, member functions etc. of the class diagram
- 2) Draw the class diagram as per the norms.
- 3) Identify various elements such as various objects of the object diagram
- 4) Draw the object diagram as per the norms.

EXPERIMENT-7

Aim: To draw the behavioral view diagram : Sequence diagram, Collaboration diagram

Tools/Apparatus: Netbeans 6.0.

Procedure:

- 1) Identify various elements such as controller class, objects, boundaries, messages etc. of the sequence diagram
- 2) Draw the sequence diagram as per the norms.
- 3) Identify various elements such as for the sequence diagram of the collaboration diagram
- 4) Draw the collaboration diagram as per the norms.

EXPERIMENT-8

Aim: To draw the behavioral view diagram : State-chart diagram, Activity diagram

Tools/Apparatus: Netbeans 6.0.

Procedure:

- 1) Identify various elements states and their different transition of the state-chart diagram
- 2) Draw the state-chart diagram as per the norms.
- 3) Identify various elements such as different activity their boundaries etc. of the activity diagram
- 4) Draw the activity diagram as per the norms.

EXPERIMENT-9

Aim: To draw the implementation view diagram: Component diagram.

Tools/Apparatus: Netbeans 6.0.

Procedure:

- 1) Identify various elements of the component diagram such as the various components like client, server, network elements etc.
- 2) Draw the component diagram as per the norms.

EXPERIMENT-10

Aim: To draw the implementation view diagram: deployment diagram

Tools/Apparatus: Netbeans 6.0.

Procedure:

- 1) Identify various elements such as the hardware components of the deployment diagram
- 2) Draw the deployment diagram as per the norms.

EXPERIMENT-11

Aim: To perform various techniques for testing using the testing tool : unit testing, integration testing

Tools/Apparatus: Winrunner.

Procedure:

- 1) Identify various modules of the system so that they can be tested stand alone.
- 2) Identify the groups of the module that can be tested together in integration.
- 3) Perform the testing of the modules as a unit and in integration by using the testing tool.

EXPERIMENT-12

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-13

Aim: To draw UML diagrams using MS Visio software.

Tools/Apparatus: MS Visio software.

Procedure:

- 1) Identify various elements of the system to be drawn using the IDE.
- 2) Use the UML options of the MS Visio software to draw the diagram from experiment 4 to 10.

References

Reference books:

- Fundamentals of sSoftware 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)