CoffeeMaker Example

Laurie Williams, Dright Ho, and Sarah Heckman [Contact Authors]
North Carolina State University
CSC 326 - Software Engineering

Back to Software Engineering Tutorials

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. Use this as a reference when completing the CSC 326 tutorials.

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, 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.		

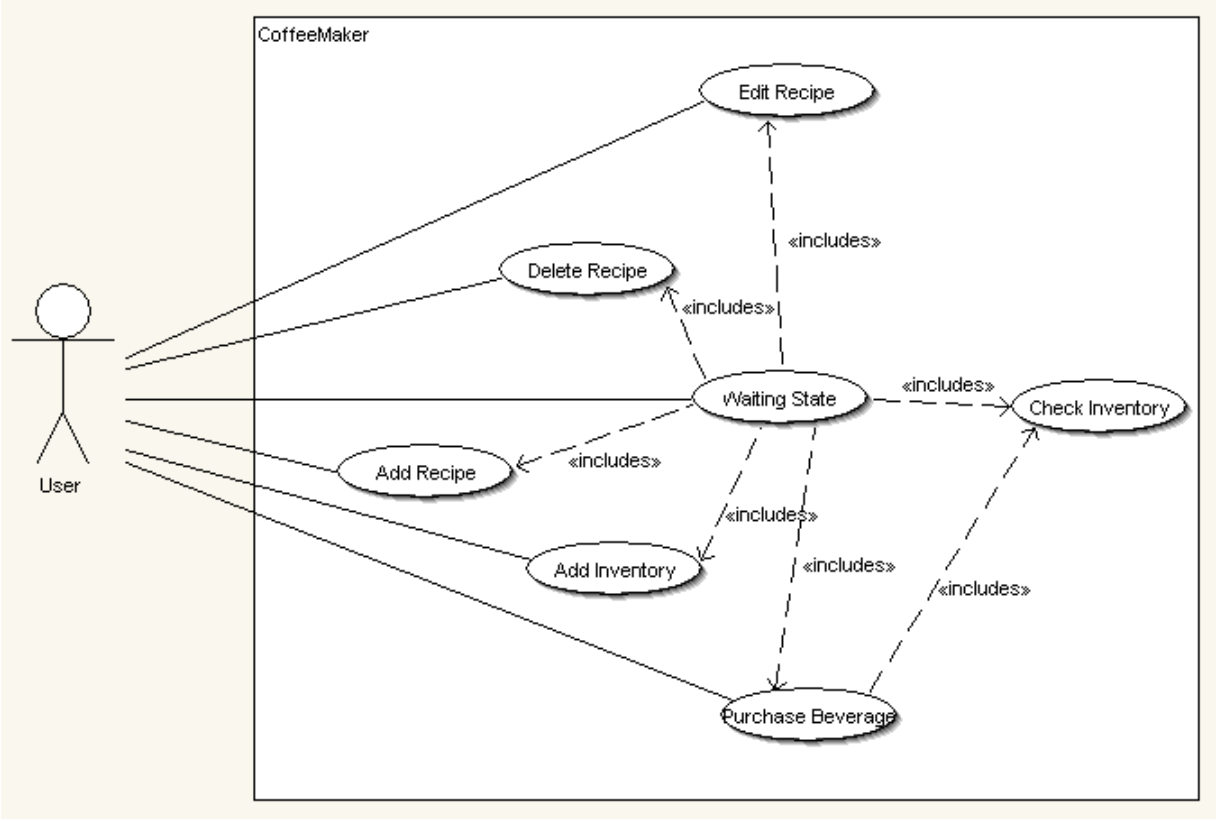
Title: Edit a Recipe		
AccTest: editRecipe1	Priority: 2	Story Points: 1
A recipe may be edited in the CoffeeMaker if it exists in the list of recipes in the CoffeeMaker. The recipes are listed by their name. After selecting a recipe to edit, the user will then enter the new recipe information. A recipe 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 inventory to make the beverage. Upon completion, the Coffee Maker displays a message about the purchase status and is returned to the main menu.		

Requirements - Use Cases



UC1: Flow of Events for the *Waiting State* Use Case
1.1 Preconditions: None
1.2 Main Flow: The Coffee Maker waits for user input. There are six options to chose from [E1]: 1) add recipe [UC2], 2) delete a recipe [UC3], 3) edit a recipe [UC4], 4) add inventory [UC5], 5) check inventory [UC6], and 6) purchase beverage [UC7].

UC2: Flow of Events for the *Add Recipe* Use Case
2.1 Preconditions: None
2.2 Main Flow: A user selects the menu option to add a recipe. The user enters the name, price [E2][E3][E4], units coffee [E2][E3][E4], units sugar [E2][E3][E4], units milk [E2][E3][E4], and units chocolate [E2][E3][E4] that make up the recipe. [E1][E5]
2.3 Subflows: None
2.4 Alternative Flows:
[E1] If there are already three recipes in the system, then a new recipe may not be added. The user is returned to the main menu.
[E2] The price must be a integer. If the price is not an integer then, a status message is printed, and the user is returned to the main menu.
[E3] The units of coffee, sugar, milk, and chocolate must be integers. If the unit value on any of the ingredients is not an integer, a status message is printed, and the user is returned to the main menu.
[E4] The price and units of ingredients must be positive. If any number is negative, a status message is printed, and the user is returned to the main menu.
[E5] If the name of the new recipe already exists in a recipe in the system, the new recipe will not be added.

UC3: Flow of Events for the *Delete Recipe* Use Case
3.1 Preconditions: None
3.2 Main Flow: The user will be shown a list of all recipes in the system, and asked to choose the recipe, by number, that they wish to delete. [S1][E1][E2][E3]
3.3 Subflows:
[S1] A message is printed describing the successful deletion of the selected recipe.
3.4 Alternative Flows:
[E1] If the user selects a number that is out of bounds of the number of recipes, the user is returned to the main menu.
[E2] If the user enters a alphabetic character, the user is returned to the main menu.
[E3] If the user selects an empty recipe to delete, the user is returned to the main menu.

UC4: Flow of Events for the *Edit Recipe* Use Case
4.1 Preconditions: None
4.2 Main Flow: The user will be shown a list of all recipes in the system, and asked to choose the recipe, by number, that they wish to edit. [E1][E2] The user enters the price [E3][E4][E5], units coffee [E3][E4][E5], units sugar [E3][E4][E5], units milk [E3][E4][E5], and units chocolate [E3][E4][E5] that make up the modified selected recipe. [E6]
4.3 Subflows: None
4.4 Alternative Flows:
[E1] If the user selects a number that is out of bounds of the number of recipes, the user is returned to the main menu.
[E2] If the user enters a alphabetic character, the user is returned to the main menu.
[E3] The price must be a integer. If the price is not a number then, a status message is printed, and the user is returned to the main menu.
[E4] The units of coffee, sugar, milk, and chocolate must be integers. If the unit value on any of the ingredients is not an integer, a status message is printed, and the user is returned to the main menu.
[E5] The price and units of ingredients must be positive. If any number is negative, a status message is printed, and the user is returned to the main menu.
[E6] If the user selects an empty recipe to edit, the user is returned to the main menu.

UC5: Flow of Events for the *Add Inventory* Use Case
5.1 Preconditions: None
5.2 Main Flow: The user will be prompted for the units of each of the four ingredients they wish to add to the inventory. [E1][E2]
5.3 Subflows: None
5.4 Alternative Flows:
[E1] If the user selects a number that is negative or a non-Integer, the user will be reprompted for the amount.
[E2] If the user enters a alphabetic character, the user will be reprompted for the amount.

UC6: Flow of Events for the *Check Inventory* Use Case
6.1 Preconditions: None
6.2 Main Flow: The user will be shown a listing of the inventory of ingredients in the CoffeeMaker
6.3 Subflows: None
6.4 Alternative Flows: None

UC7: Flow of Events for the *Purchase Beverage* Use Case
7.1 Preconditions: None
7.2 Main Flow: The user will select the beverage they wish to purchase. The user will deposit money to pay for the beverage. [S1][S2]
7.3 Subflows:
[S1] The CoffeeMaker will check if there are enough ingredients in the inventory to make the selected drink. [E1]
[S2] The CoffeeMaker will make sure enough money was deposited [E2], the beverage will be dispensed, and any extra change will be returned.
7.4 Alternative Flows:
[E1] If there is not enough inventory to make the beverage, a message will be displayed, the user's money will be returned, and the user will be returned to the main menu.
[E2] If the user does not enter enough money, their money will be returned, and the user will be returned to the main menu.

Requirements	Design	Test	Code
• User Stories • Use Cases • Use Case Diagram	• Class Diagram • Sequence Diagram	• Acceptance Tests	• CoffeeMaker_JUnit.zip - JUnit Tutorial • CoffeeMaker_FIT.zip - FIT Tutorial • CoffeeMaker_Web.zip - JSP Tutorial • CoffeeMaker_WebDB.zip - JDBC Tutorial • CoffeeMaker_WebTest.zip - HttpUnit Tutorial • CoffeeMaker_Coverage.zip - EclEmma, Jazz Test Coverage, djUnit, MuClipse

---

Back to Software Engineering Tutorials
