

Cover Page

DePaul University

SE 430: Object-Oriented Modeling

Assignment:

Team Final Project

Team members:

Nirav Dedhia

Nicholas Kincaid

Sai Srikanth Nuthalapati

James Tyson

Athina Verroiopoulou

November 19, 2015

Project Overview Statement:

This project is to implement a Unified Smart Kitchen Application to enable users control of their kitchen tasks, appliances, and kitchen management from their hand-held devices.

Product Description:

The Unified Smart Kitchen Application (USKA) will provide users the control and management of multiple kitchen devices from their hand-held devices. Using Wi-Fi technology in new kitchen and home hardware, the USKA will be able to communicate with these devices and store different types of data and information. The USKA will have an Inventory System that will store all the food inventory information and this data can be accessed by other subsystems to create seamless cooking experience from deciding on a recipe to beginning of the prepping process. Using sensors placed throughout the kitchen, the USKA will have a Safety System which its purpose is to monitor the environment in the kitchen to alert and/or take action in the case of emergencies. The USKA will give users remote control over multiple appliance such as the oven, refrigerator, microwave, stovetop, etc. This will give the user the ability to do task such as remotely preheat the oven, control temperature of different compartments of your refrigerator, or control the heat levels on the stovetop. The appliances will be able to provide feedback to the user on their hand-held device about the task being performed or their status. The USKA has the ability to have its inventory system communicate with the recipe system to compile a list of possible recipes that can be made with the currently available products. This list of recipes also can be filtered based on preferences the user creates. When a recipe is picked, the USKA will provide cooking instructions to the user. Should the recipe require the use of an appliance, such as the oven or the stove top, the appliance will turn on and adjust to the cooking needs of the recipe.

Project Objective:

The primary objective of the Unified Smart Kitchen Application is to connect the wifi enabled kitchen devices and enable users to quickly accomplish kitchen tasks. The USKA will also enable users to quickly accomplish management of available foods in their kitchen while streamlining recipe choices and appliance preparation.

Cost and Benefits:

- Cost: Hardware (High)
- Cost: Application software (High)
- Cost: On-going software and system maintenance (Medium)
- Cost: Annual fee to fire department for safety 'on-call' feature (High)
- Benefits: Streamline experience in the Kitchen and ease of user experience (High)
- Benefits: Protection during emergencies that occur in the kitchen (High)

Project Scope:

The scope of the USKA is just the domain of the Kitchen. All the systems in this project only relate to the appliances and task related to the kitchen environment.

Subsystem Interfaces:

- Interface: Inventory System
 - Purpose: List of food products available in kitchen and other facts related to cost, quantity, expiration date, etc.
- Interface: Safety System
 - Purpose: Monitors equipment in the kitchen and in the case of emergencies, contacts authorities and activates safety mechanisms
- Interface: Appliance Control System
 - Purpose: Controls wifi connected kitchen appliances
- Interface: Recipe System
 - Purpose: Access to library of recipes and communicates with Inventory System and Appliance Control System to streamline recipe selection and preparation

Competitive Analysis:

This is a one of a kind system being available to your average consumer, thus no competition exists in this space. Other Smart Kitchen are custom jobs are only available to high paying consumers. Smart kitchen appliances are premium appliances that are designed for comfortable and efficient operation as compared to the traditional appliances. The most significant advantage of smart kitchen appliances is the high energy efficiency over traditional kitchen appliances. Smart kitchen appliances are equipped with different sensors that facilitate easy operation and safety for the users. In addition with the use of the USKA some of the smart kitchen appliances are designed with connectivity features that can connect to handheld devices such as smartphones, tablets and others. Thus, users can operate their kitchen appliances remotely from their handheld devices thereby increasing the ease of usage of these appliances.

Use cases:

UC1: Safety System: Stove top fire - Fire extinguisher response

Description: In reaction to a stove top fire, the Safety System keeps users alerted and takes actions to any potential stovetop fires. The Safety System monitors for potential fires and in case of an emergency, contacts the fire department and attempts to put out the fire automatically. The USKA will call the user and send notification box about the fire. Once the fire is under control, the user will be notified again that the stove top cover has been dropped, the fire has been suffocated, the stove top has been shut off, and the fire department has been contacted.

Prioritization:

This use case provides one of the most essential element of the smart kitchen system. The ability to track fire with the help of sensors and send emergency alerts to the user. The system has a huge impact as it deals with safety and can prevent fire spread. From the perspective of implementation it has a technological risk.

Actors: User, Fire Department

Priority: High

Risk: High

Type: Primary

Trigger: The stovetop caught fire and the sensors detect it .

Purpose or Goal: Alert the user in case of an emergency and distinguish the fire

Cross References: UC3 (Appliance control system) can be used to turn off the stovetop

Incoming information: Sensor information for fire detection, Fire department response

Preconditions:

- The user has set up and enable the sensors.

Postconditions:

- After the fire is distinguished the system get reset. .

Event flow:

1. The use case begins when the stovetop caught fire and the sensors detect it.
2. The Safety system starts the alarm and try distinguish the fire.
3. The fire-suppressing powder is automatically released onto the fire.
4. Drop stove top cover and turn off stove top.
5. The Safety system call the Fire Department to inform about the fire.
6. Fire department responds.
7. The Safety system send information about the location of the place the fire is on.
8. The Safety system notify the user by call and send him an notification box in his device to inform him that the stove top is on fire.
9. The use case ends when the system notify the user again that everything is under control and fire is extinguished.

Alternate Event Flow 1:

- 4) The automated mechanism for distinguishing the fire is unable to drop the stove top cover.
- 4.1.) The system call the Fire Department to inform about the fire.
- 4.2) Fire department responds.
- 4.3) The system send information about the location of the place the fire is on.
- 4.4) The system calls the user and sends a notification box to his device to inform him that the stove top is on fire and the automated mechanism unable to drop the stove top cover.
- 4.5). The use case ends when the system notify the user again.

UC2: Inventory System

Description: The user uses the Inventory System to log and maintain the food inventory in the kitchen. User has the ability to log the food inventory by scanning its barcode, then manually entering an expiration date and quantity of items via the handheld device. It will then automatically update the data about the items being stored in the kitchen. If the item does not have a barcode the user has the ability to input the data manually. User can keep track of inventory in the kitchen as it is stored in Inventory System database and also USKA suggests the user about the item whenever it reaches the expiration date or quantity of the item is done. The Inventory system communicates with the Recipe System to enable suggested recipes based on the available ingredients. USKA notifies the user with a shopping list required for the kitchen.

Prioritization: This use case is more crucial to achieve goals of the smart kitchen. This functionality allows user to keep track of inventory in the kitchen through the USKA and with the use of this data, user can get suggestions from Recipe System. This also can keep a running record of monthly expenses on inventory.

Actors: User

Priority : High

Risk: Medium

Type: Primary

Goal: User can able to keep track of Inventory.

Preconditions:

- 1) Inventory system should be connected to database and USKA.
- 2) Handheld device should enable camera for scanning attributes of items by taking pictures.

Postconditions:

- 1) Inventory Update Success when User selects Update Inventory option
- 2) Gets the information of the Item in the Inventory when the user selects Check Inventory option.
- 3) Gets the expenses of items when the user selects Check Expenses option.
- 4) Gets the list of items when the user selects Shopping cart option.

Event Flow:

- 1) The use case starts when the user selects an option from handheld device to open the inventory and select an option (Update Inventory, Check Inventory, Check Expenses, Shopping Cart).
- 2) The user scans the items through the camera of handheld device to update the item, quantity, cost, expiration date of the inventory.
- 3) User has an option to update manually if item specification is not clear.
- 4) User should select Update Inventory option whenever he uses the item in Inventory.

5)when quantity reaches to minimal or expiration date is nearby,Inventory sends information to the user.

6)user has an option to check expenses on Inventory in different formats like monthly expenses,yearly expenses,Expenses on an item for a period etc.

7)USKA returns the information of the expenses.

8)User has a Shopping cart option where he can check the list of items required for shopping.

9)Use Case ends when the Options selected are successful.

UC3: Oven control in Appliance Control System

Description: The user uses the USKA to cook their food in the smart oven. The user selects an option from the Appliance Control System to start the oven and preheat it to their desired temperature for what they are cooking, and when that temperature has been met, they are alerted via notification box. The user has 5 minutes from the time the oven hit the preheated temperature to use another option to set the timer for how long they would like their food to be cooked. If the timer is not set, the oven will automatically turn off. The user can use a camera inside the oven to look at their food being cooked. User can use another option to raise or lower the internal racks to better position the food for optimal cooking position. The user will also be alerted via notification box when the timer has been reached and the oven will turn off automatically. The Appliance Control System has integration with the Recipe System to prepare the oven, once a recipe has been selected.

Prioritization: This use case is one of the high priority use cases for deployment. This functionality allows the user to control a major appliance from their device, while not having to be around the appliance, which is the essence of what the smart kitchen is all about.

Actors: User

Priority: High

Risk: High

Type: Primary

Trigger: User selects an option from the Appliance Control System to start their oven.

Purpose or Goal: User is able to cook their food in an oven while being in a remote location.

Preconditions:

- Oven needs to be connected to the USKA
- Oven is not being used for anything else

Postconditions:

- Food is cooked
- Oven is turned off

Event Flow:

1. The use case begins when the user selects an option from the Appliance Control System to start their oven.
2. The user selects on the pre-heat option.
3. The user sets their desired preheat temperature.
4. The oven sends a notification that the preheat temperature has been met.
5. The user sets their desired cooking time for their food.
6. The user uses the camera in the oven to look at their food being cooked.
7. The oven sends a notification that the timer has been reached.
8. The use case ends when the timer has been reached and the oven turns off automatically.

Alternate Event Flow:

5a. The user takes over 5 minutes without setting their desired temperature.

5.1 The use case ends with the oven turning off automatically.

UC4: Recipe System: Preparing a meal

Description: When the user wants to prepare a meal, the user can utilize the recipe system to understand what recipes that can be made with the existing inventory of items. In order to do this, the user navigates to the Recipe System of the USKA and then enters preferences based on types of meals or foods that the user wants to include in the meal. For instance, if the user wants to see recipes that utilize the foods that are about to expire, it will have the ability to suggest such meals. When the user is done selecting their preferences, the recipe system queries the foods that are in inventory and prioritizes them based on the input preferences. The recipe system then interfaces with an open source recipe list and its own list to present recipes that satisfy the input preferences. The user can then determine the final selection and can select that recipe in the USKA interface. Once the meal has been selected, the recipe system shows the steps to complete the preparation of the meal. If the recipe requires an oven preheat, it will send a request to the Appliance Control System to turn it on. Once the meal has been prepared, the user will then indicate to the recipe interface whether to automatically update the inventory of the items used in the recipe.

Upon completion of the recipe, the user can upload pictures of the resulting meal, along with ratings and recipe edit modifications. The recipe system will then save this data to its database so that it can be scanned in future queries.

Prioritization: This use case is mid priority. The implementation of the inventory system comes as a higher priority than this system because the inventory system is a dependency of the recipe system, but the recipe system is not a dependency of the inventory system.

Actors: User

Priority: Medium

Risk: Medium

Type: Primary

Trigger: The user wants to prepare a new meal.

Purpose or Goal: Provide the ability for the user to input meal type preferences, supply with available meal recipes to choose from, and provide a way to rate the meal in the end.

Cross References: Appliance control system to initialize the preheating of the oven. Inventory system to scan for existing ingredients. Open source recipe list to get possible recipes.

Incoming information: The user needs to set preferences and select the recipe to use.

Preconditions: None

Postconditions: The user has used a recipe from the list and has rated it in the system.

Event flow:

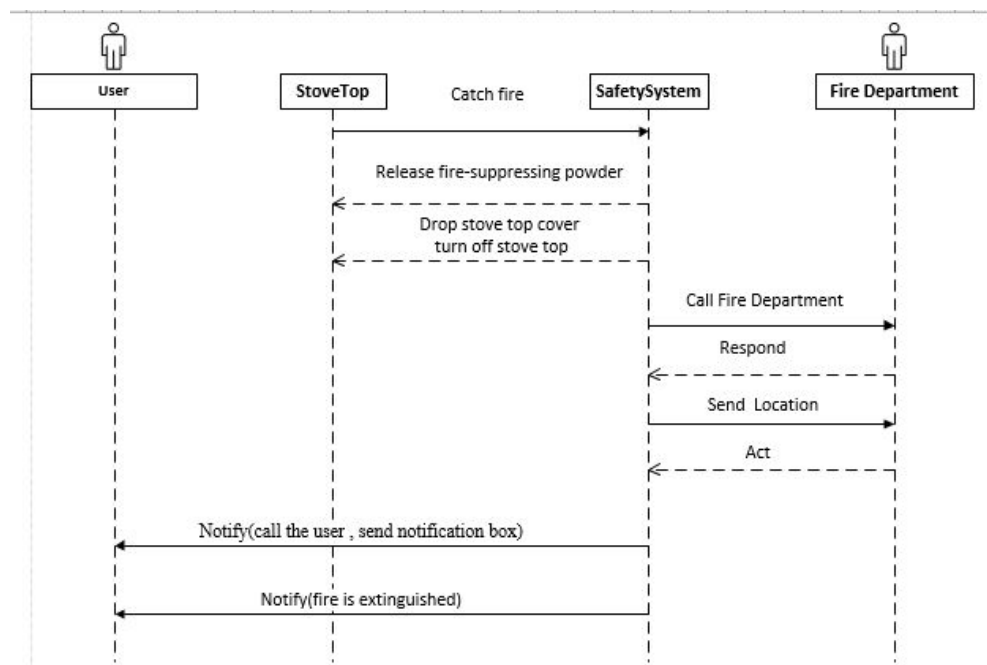
1. The use case begins when the user wants to prepare a meal and opens the recipe system
2. The user selects begin a new meal
3. The user selects the meal preferences
4. The recipe system looks for available ingredients in the inventory system
5. The recipe system looks for possible recipes to make with what the inventory contains
6. The recipe system presents the options to the user
7. The user selects which recipe to use
8. The recipe system sends a message to the oven to preheat in the case that baking is needed
9. The recipe system send a message to the user to begin preparing the meal
10. The user follows the recipe to prepare and cook the meal
11. The user tastes the meal
12. The user rates the meal
13. The use case ends when the user has tasted the meal and rated the recipe and closes the app.

Alternate Event Flow 1:

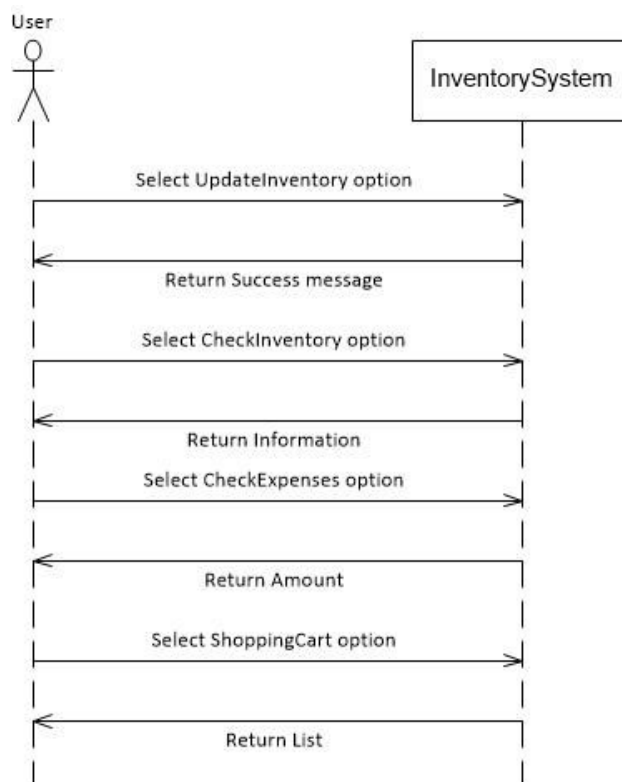
3. The user selects the option to look for meal recipes that contain ingredients that are not listed in the recipe system.
 - 3.1 The recipe system bypasses looking in the ingredients section
 - 3.2 The recipe system looks in the recipe list without the constraint of using ingredients from inventory
 - 3.3 Rejoin the main branch back at step 6.

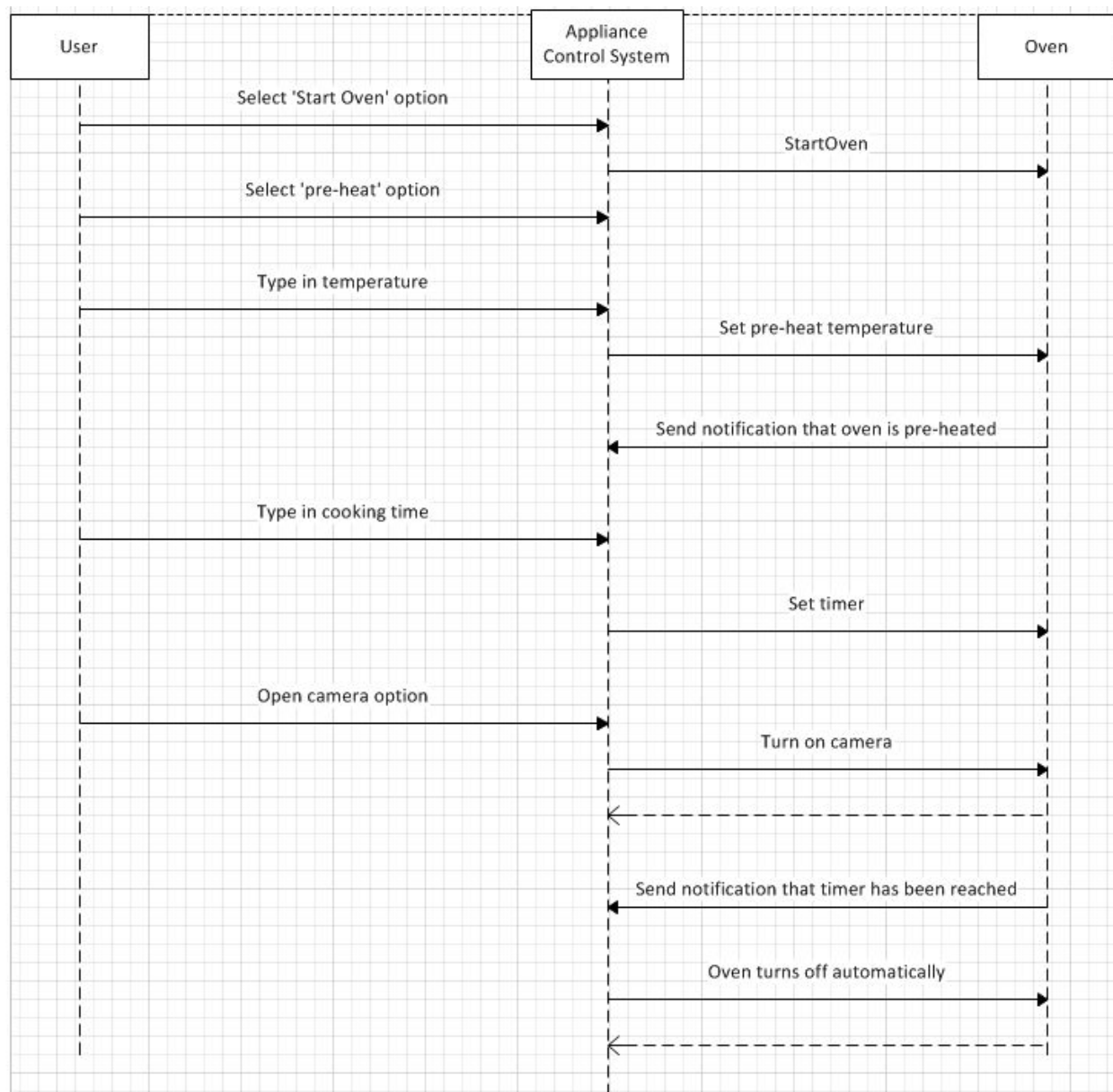
Sequence Diagram:

Safety System: Stove top fire - Fire extinguisher response

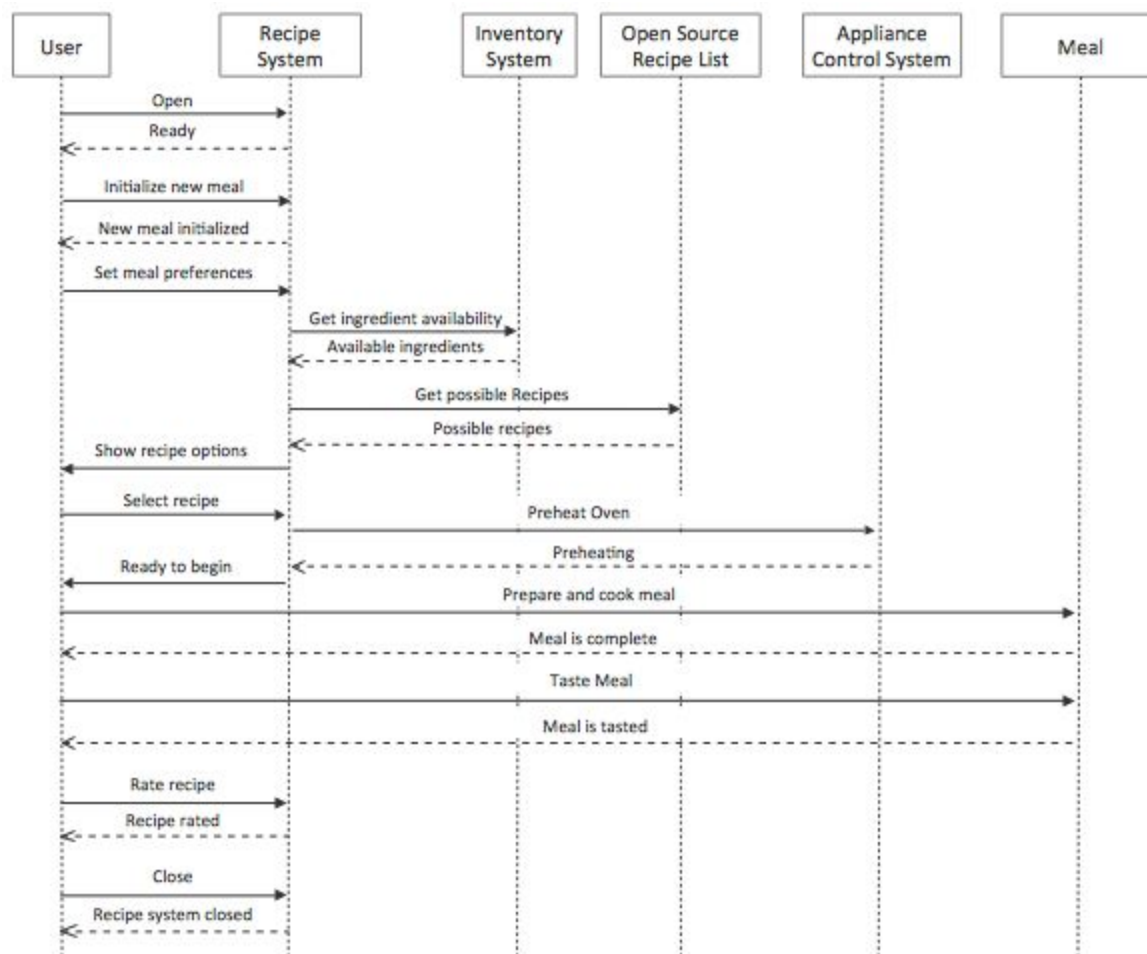


Inventory System

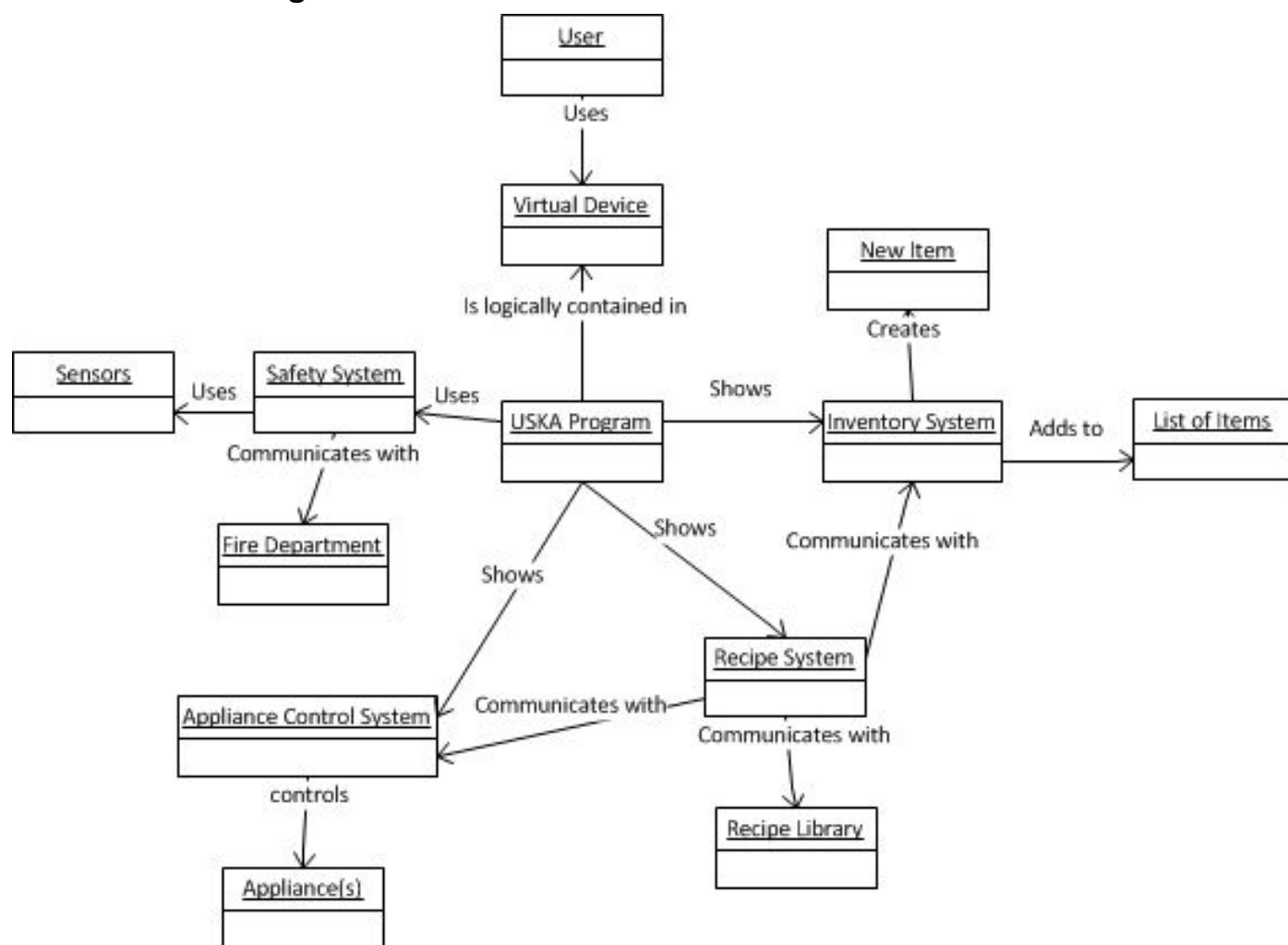


Oven control in Appliance Control System

Recipe System: Preparing a meal



Domain Model Diagram



Design Scenario:

Oven control in Appliance Control System

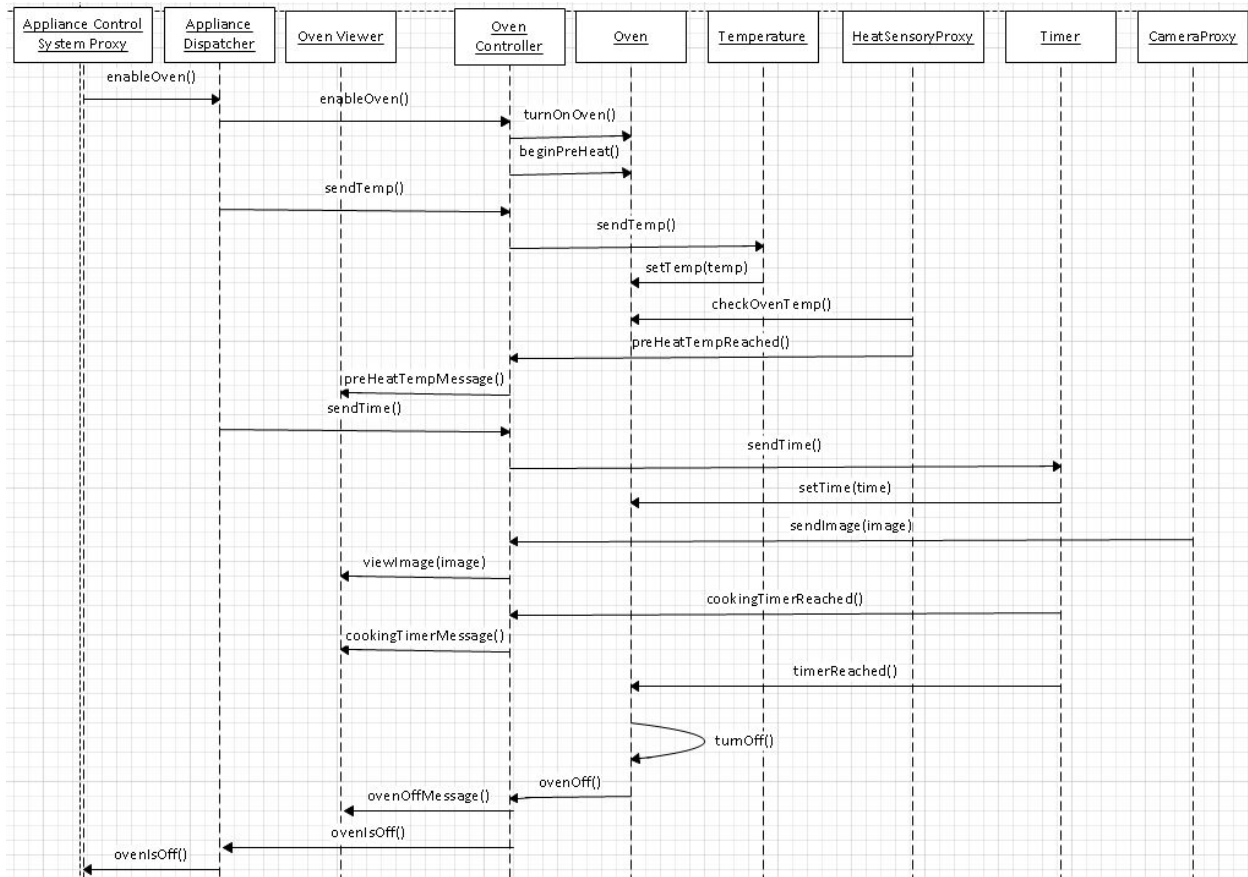
This use case is a perfect example for the MVC pattern

Event Flow:

- 1 The use case begins when the user selects an option from the Appliance Control System to start their oven.
 - 1.1 Appliance Control System Proxy invokes the Oven Controller that the oven will be in use. [This is a use of the proxy pattern.]
 - 1.1.1 Implement the enableOven() method.
 - 1.2 The Oven Controller sends a call to the oven class to turn on. [Controller pattern]
 - 1.2.1 Implement the turnOnOven() method.
- 2 The user selects on the pre-heat option.
 - 2.1 Oven Controller sends a call to the oven class to begin pre-heating.
 - 2.1.1 Implement the beginPreHeat() method.
- 3 The user sets their desired preheat temperature.
 - 3.1 The temperature class will receive a temperature from the Oven Controller
 - 3.1.1 Implement the sendTemp() method.
 - 3.2 The temperature class will set the oven at the desired pre-heat temperature
 - 3.2.1 Implement the setTemp(temp) method.
- 4 The oven sends a notification that the preheat temperature has been met.
 - 4.1 The heatSensoryProxy class is looping to check the oven to find out what temperature it has met.
 - 4.1.1 Implement the checkOvenTemp() method.
 - 4.2 The heatSensoryProxy class will send a message to the Oven Controller that the pre-heat temperature has been met.
 - 4.2.1 Implement the preHeatTempReached() method.
 - 4.3 The Oven Controller sends a message to the Oven Viewer that the pre-heat temp has been met. [Viewer Pattern]
 - 4.3.1 Implement the send preHeatTempMessage() method.
- 5 The user sets their desired cooking time for their food.
 - 5.1 The Appliance Control System Proxy sends the desired cooking time to the Oven Controller.
 - 5.1.1 Implement the sendTime() method.
 - 5.2 The timer class will receive a cooking time from the Oven Controller
 - 5.2.1 Implement the sendTime() method.

- 5.3 The timer class will set the oven to cook for the desired time.
 - 5.3.1 Implement the setTime(time) method.
- 6 The user uses the camera in the oven to look at their food being cooked.
 - 6.1 The camera class will send video of the food to the Oven Controller
 - 6.1.1 Implement the sendImage(image) method.
 - 6.2 The Oven Controller passes images to the Oven Viewer
 - 6.2.1 Implement the viewImage(image) method.
- 7 The oven sends a notification that the timer has been reached.
 - 7.1 The timer class sends a message to the Oven Controller that the timer has been met.
 - 7.1.1 Implement the cookingTimerReached() method.
 - 7.2 The Oven Controller sends a message to the Oven Viewer that the timer has been met.
 - 7.2.1 Implement the cookingTimerMessage() method.
- 8 The use case ends when the timer has been reached and the oven turns off automatically.
 - 8.1 The timer class sends a call to the oven class, letting it know that the time has been reached.
 - 8.1.1 Implement the timerReached() method.
 - 8.2 The oven class sends itself a call to turn off.
 - 8.2.1 Implement the turnoff() method.
 - 8.3 The oven class sends a message to the Oven Controller that the oven has been turned off.
 - 8.3.1 Implement the ovenOff() method.
 - 8.4 The Oven Controller lets the Oven Viewer that the oven has been turned off.
 - 8.4.1 Implement the ovenOffMessage() method.
 - 8.5 The Oven Controller lets the Appliance Control System Proxy that the oven has been turned off.
 - 8.5.1 Implement the ovenIsOff() method.

Use-case realization Diagram for UC3: Oven control in Appliance Control System



Design Scenario for UC2: Inventory System:

Event Flow:

1. The Usecase begins when the user calls to execute Inventory system from the Appliance Control System to open operations

1.1 user invokes InventoryDispatcher to open Inventory system operations

1.1.1. Implements the ExecuteInventory() method.

2. InventoryDispatcher needs to be notified whenever it activates or select options

2.1 InventoryDispatcher invokes a call to InventoryMonitor that user have an option to activate the inventory in the kitchen.

2.1.1. Implements the Update(activate) method.

2.2 InventoryDispatcher invokes a call to InventoryMonitor that user have an option to select the options required for the inventory in the kitchen.

2.2.1. Implements the Update(option) method.

2.3 InventoryDispatcher invokes a call to InventoryMonitor that user have an option to deactivate the inventory in the kitchen.

2.3.1 Implements the Update(deactivate)

3. InventoryController needs to notified whenever user select options like add item,check item,check expenses,check shopping cart

3.1 InventoryMonitor invokes a call to InventoryController that user have an option to add items in inventory in the kitchen.

3.1.1. Implements Update(addItem) method.

3.2 InventoryMonitor invokes a call to InventoryController that user have an option to check items in inventory in the kitchen.

3.2.1. Implements Update(checkItem) method.

3.3 InventoryMonitor invokes a call to InventoryController that user have an option to check expenses of inventory in the kitchen.

3.3.1 Implements Update(checkExpenses) method.

3.4 InventoryMonitor invokes a call to InventoryController that user have an option to check the shopping cart to know list of item required for inventory in the kitchen.

3.4.1 Implements Update(ShowCart) method.

4. User selects the Add Items option

4.1 The InventoryController sends a call to InventoryLog to add the item in the Inventory.

4.1.1. Implements Update(item,cost,quant,expdate) method.

4.2 The InventoryLog sends back a call to InventoryController that updating item in the inventory is success.

4.2.1. Implements Success(item).

5. user selects the Check Items option

5.1 The InventoryController sends a call to InventoryLog to check the items in the Inventory.

5.1.1 Implements check(item) method.

5.2 The InventoryLog sends back a call to InventoryController about the information of the item.

5.2.1 Implements show(item) method.

6. user selects the Check Expenses option

6.1 The InventoryController sends a call to ExpenseCaluculator to check the expenses of the inventory in kitchen.

6.1.1 Implements ShowExpenses(item,date) method.

6.2 The ExpenseCaluculator sends back a call to InventoryController about the expenses of Inventory in the kitchen.

6.2.1 Implements ReturnExpenses() method.

7.ShoppingCart gets notified with the information of items which reaches expiration date or minimal quantity

7.1 InventoryLog invokes the ShoppingCart whenever item in the inventory reaches expiration date.

7.1.1 Implements Expire(item) method.

7.2 InventoryLog invokes the ShoppingCart whenever item in the inventory reaches minimal quantity of the item

7.2.1 Implements MinimumQuantity(item) method.

8.user selects an option to check Shopping Cart

8.1 The InventoryController sends a call to ShoppingCart to check the shopping list required for the inventory in kitchen.

8.1.1 Implements ShowList() method.

8.2 ShoppingCart sends backs a call to InventoryController with the list of shopping items required for the kitchen.

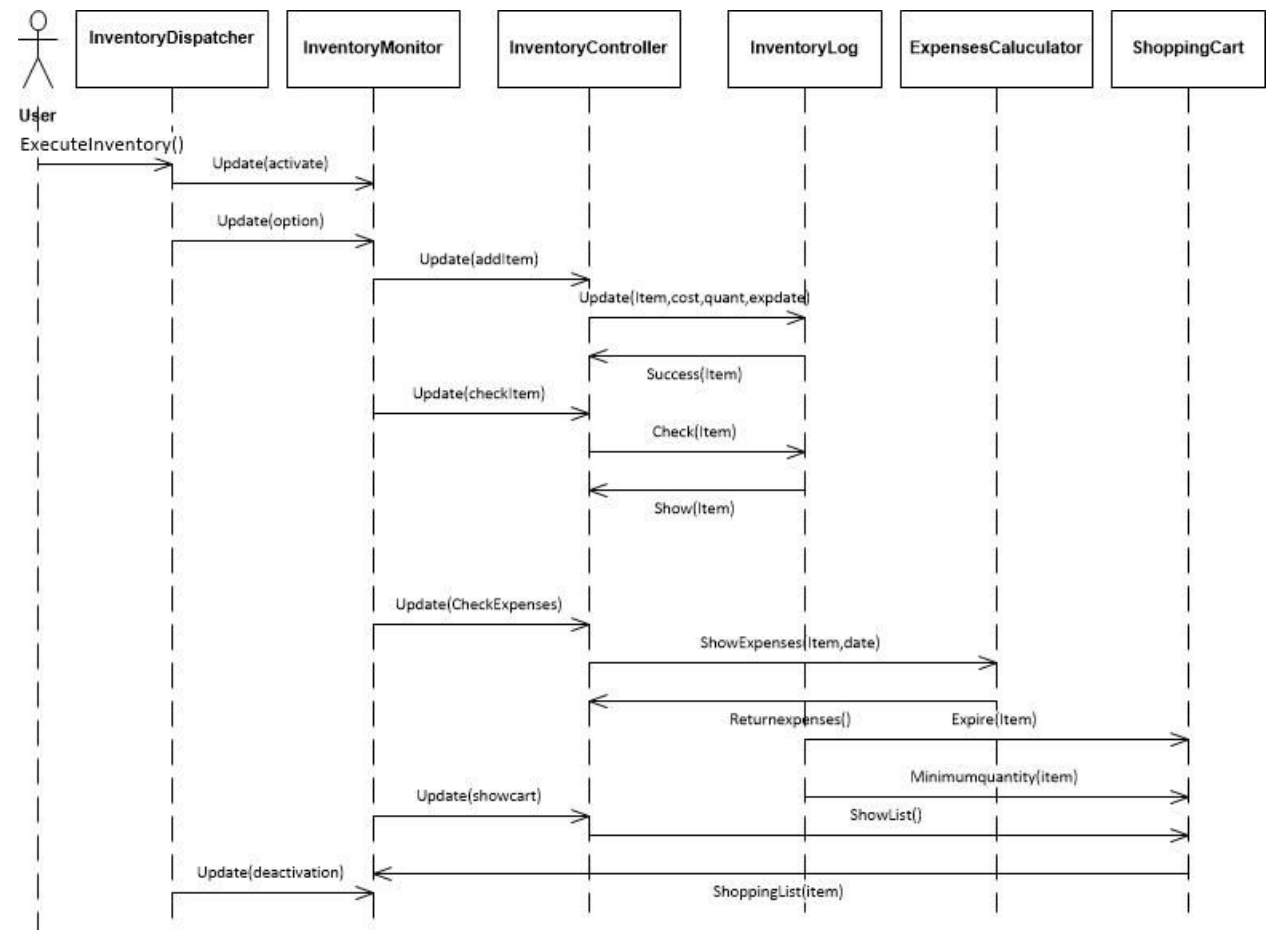
8.2.1 Implements ShoppingList(item) method.

9.usecase ends when InventoryDispatcher notifies InventoryMonitor with deactivation message

9.1 InventoryDispatcher invokes on InventoryMonitor with deactivation message.

9.1.1 Implements Update(deactivate) method.

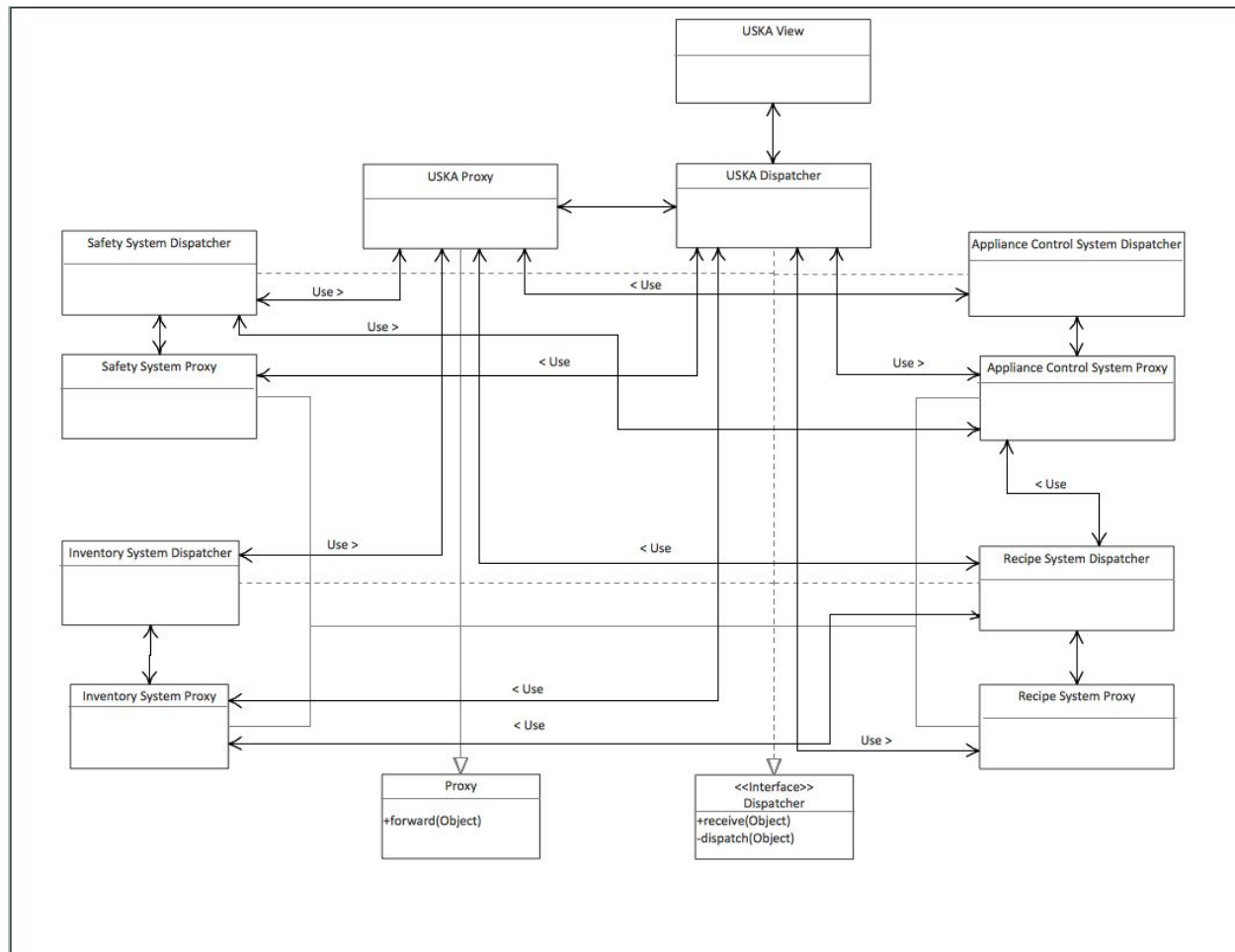
Use-case realization Diagram for UC2: Inventory System:



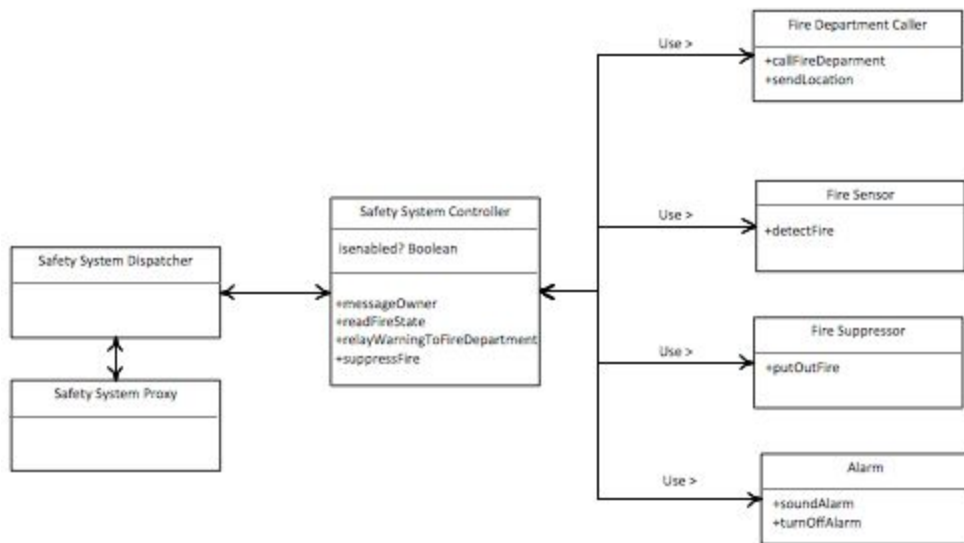
Design Class Diagram

The design class diagram is split up into the high level view of how the subsystems interact and the low-level views of how the individual subsystems operate within themselves:

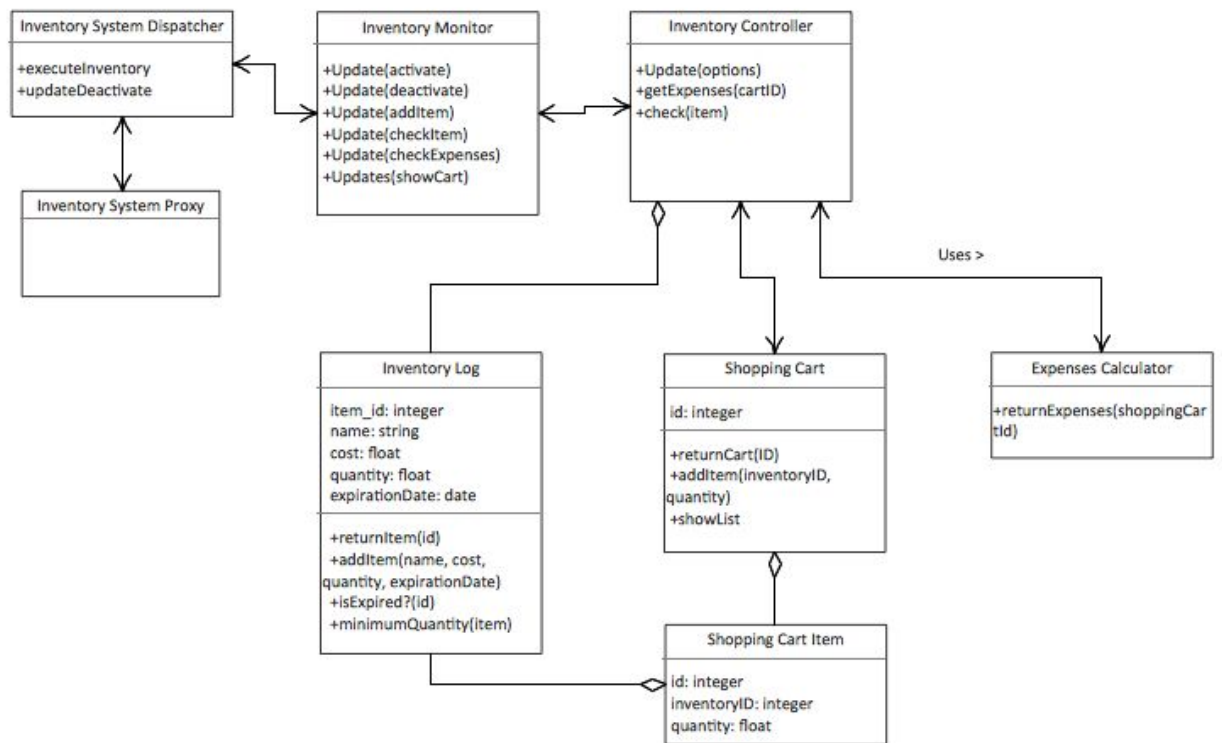
High Level Design Class Diagram:



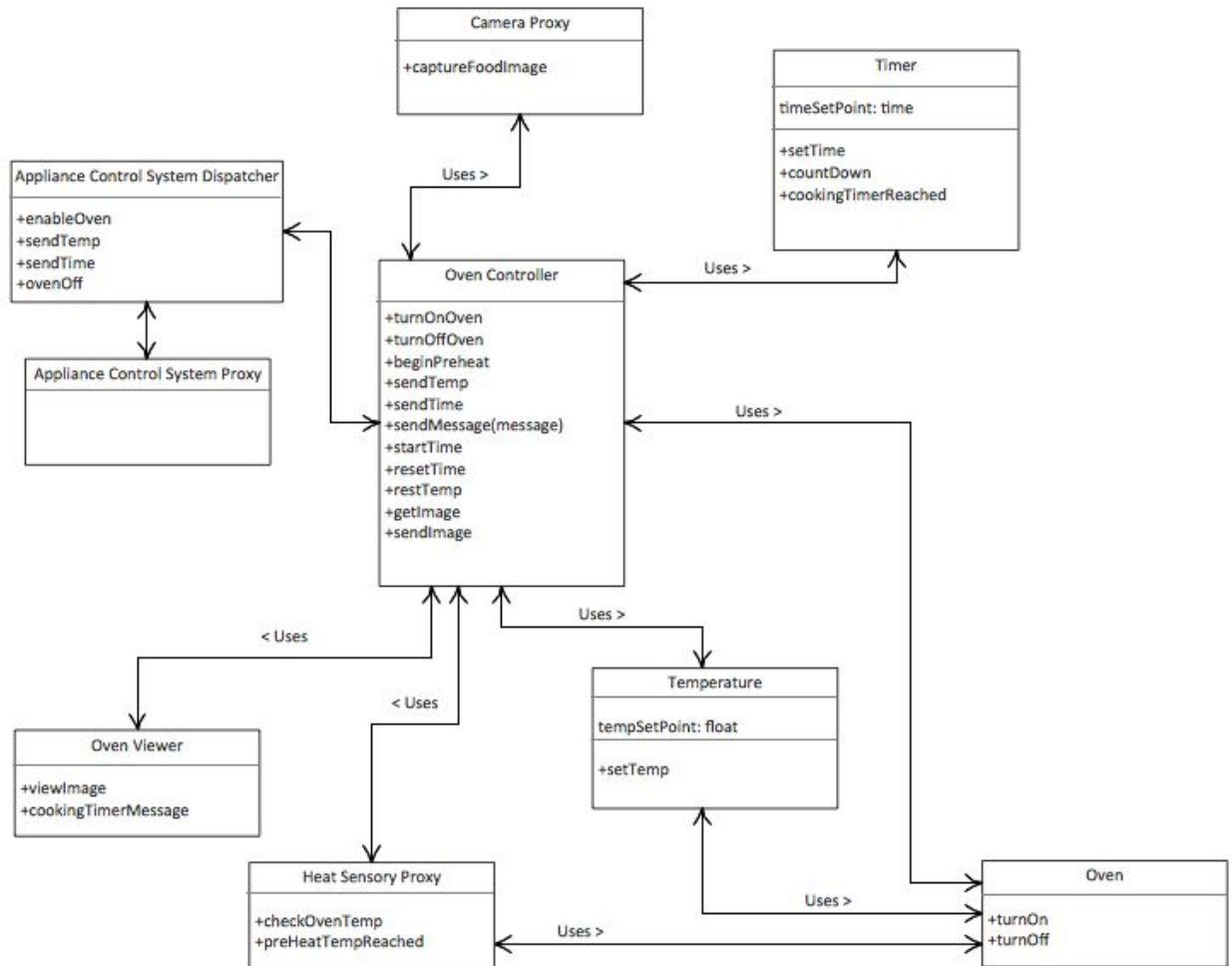
Safety System Sub-System Design Class Diagram:



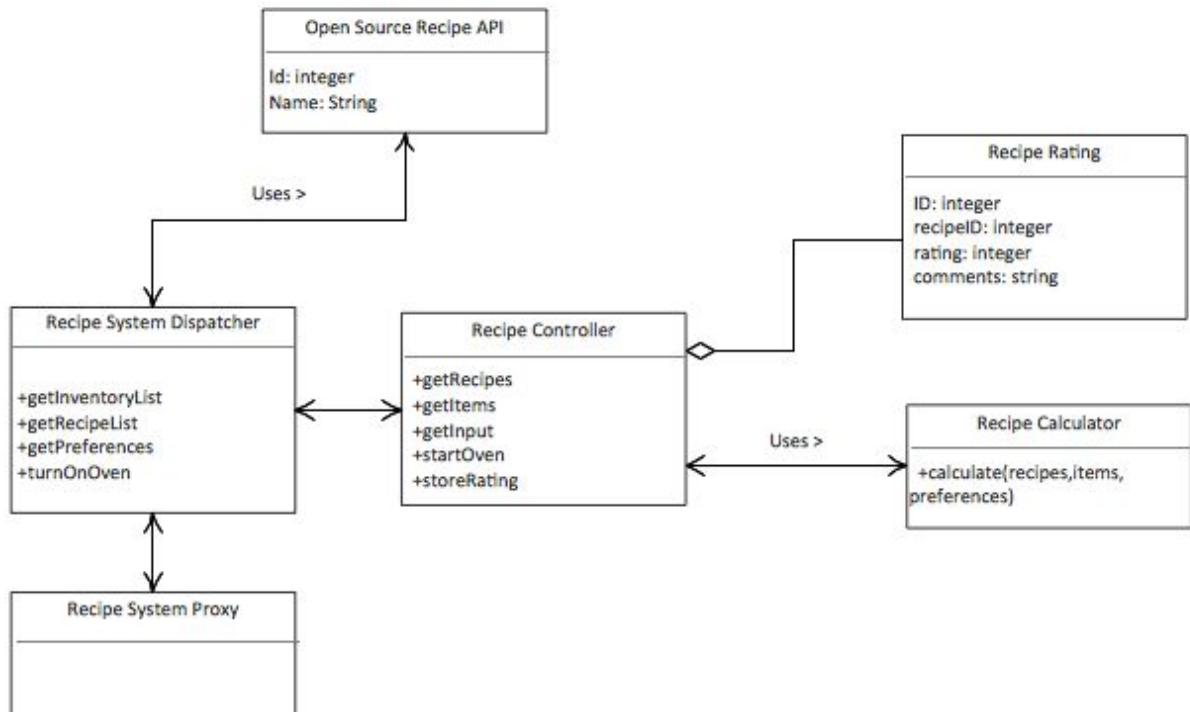
Inventory System Sub-System Design Class Diagram:



Appliance Control Sub-System Design Class Diagram:



Recipe System Sub-System Design Class Diagram:



Glossary:

Fire extinguisher: Alerts the user if a fire start in the kitchen stove and is responsible for extinguisher the fire.

Control information: Enable user to set cooking temperature and time and alert him.

Visual content display: With the use of a projecting cooking mechanism it shows the food process and alert the user in any case.

Application: It's a software application that can be used from any hand-held device to select any option from the kitchen menu.

Shelf: Same as rack.

Barcode: used to capture and read information contained in a item.

StoveTop: the kitchen stove top that is detected for potential fire. In case of an emergency actions will be invoked to distinguish a fire in it

USKA: The Unified Smart Kitchen Application that will enable a user to control and manage multiple kitchen devices , see the food item inventory and based on that see suggested recipes and be alerted in case of fire.

Use cases:

UC1: Safety System: Stove top fire - Fire extinguisher response:

Users notified in reaction to a stovetop fire. Actions are emerged to distinguish a potential fire.

UC2: Inventory System:

The users can update or check the existence of their food inventory. Items that are stored inside are viewed by this system. Based on them we can calculate expenses , check for existing recipes or notified when an item reaches the expiration date or quantity of the item is done.

UC3: Oven control in Appliance Control System:

The users can control how to cook their food in the smart oven and are notified when their selected actions are done. They can select the preheat temperature , set the timer for how long they would like their food to be cooked and raise or lower the internal racks. We can set oven function based on a recipe from recipe system.

UC4: Recipe System: Preparing a meal:

When users want to prepare a meal they can have access to recipes that can be made with the existing inventory of items. Users enters their food preferences and see a recipe list based on that. The users selects a recipe and see the the steps to complete the preparation of the meal. If the recipe requires an oven preheat, it will send a request to the Appliance Control System to turn it on. Once the meal has been prepared, the user will then indicate to the recipe interface whether to automatically update the inventory of the items used in the recipe. Upon completion of the recipe, the user can upload pictures of the resulting meal, along with ratings and recipe edit modifications.

Domain Model - Use-case realization Diagram and Design diagram Glossary:

User: the person who uses USKA application

Virtual Device : the handheld device that a user have to use the USKA application

USKA Program: the application program that includes and controls all the different system interaction with the user.

Safety System: The system that is responsible for notify the user in case of stovetop fire.

Fire Department: the department of a local or municipal authority in charge of preventing and fighting fires.

Sensors: A flame detector that is designed to detect and respond to the presence of a flame or fire. It is installed for the stove top safety.

Appliance Control System:

Inventory System: A system through which all the Inventory interactions in the kitchen is managed.

Appliance control system: A system in which all of the appliances in the smart kitchen can be controlled remotely, thru a handheld device. The appliances include fridge, oven and microwave.

Appliances: the devices that appliance control system controls such as oven.

New Item: is the new items that we add in our inventory (manually or with a barcode) in order to update it.

List of items: is the existing items that the inventory has inside and the user can check them for calculate his expenses or find a recipe that matches with what exist in the inventory.

Recipe System: it is the system that search for a new recipe based on the inventory ingredient or the preferences of the user meal.

Open Source Recipe Library: it is the database where the recipes are stored in order to view and rated by the user.

Design scenario glossary:

CameraProxy: A proxy class for a camera subsystem that captures images of the food being cooked and sends the images to the Oven Controller.

Heat Sensory Proxy: A proxy class for a heat sensory subsystem that measures the temperature of the oven and sends the result to the Oven Controller.

Timer: A class that measures the amount of time the oven has been cooking.

Temperature: A class that sets the temperature for the oven class.

Oven: A class that sets the cooking temp and cooking time and turns off when the allotted cooking time has been reached.

Oven Controller: A controller class that receives the data for the oven class.

Oven Viewer: A viewer class that the user can use to get messages or images.

Appliance Control System Proxy: A proxy class for the Appliance Control System

Inventory Dispatcher: Manages processing to the Inventory system.

Inventory Monitor: observer class that monitors Inventory classe for activity

Inventory Controller: use case controller class receives event information from Inventory Monitor and passes details to the Inventory Log, it passes Inventory Log to Recipe Proxy for processing and find a recipe , it send items to the update inventory and it sends items information to expenses calculator.

Inventory Log:holds the activity of items that user consume.

Expenses Calculator: a class that calculates the expenses of items that are in the inventory log.

ShoppingCart:A class that makes the list of items required for the shopping.