



## Stakeholder Goals

KITty will allow people and "intelligent" devices and to transfer recipes and extract the information they need from them.

## Needs Statement

The KITty product design team completed a competitive product analysis (over the course of the past three months) and then used four focus groups to arrive at this needs list.

### System Needs

- S-1 Request recipes from other devices (e.g., an oven needs to be able to request a recipe from the refrigerator).
- S-2 Extract information that it needs from the recipe (e.g., an oven needs to be able to extract the temperature from a recipe).

### User Needs

A user of KITty needs to be able to:

- U-1 Request a list of the ingredients in a recipe.
- U-2 Request a list of the utensils used in a recipe.
- U-3 Request just the instructions in a recipe.
- U-4 Request a complete recipe that includes the list of ingredients, the utensils, and the instructions.

# Use Case Descriptions for the Server

The use case descriptions below involve the following actors: WWWBrowser.

## UC1. WWWBrowser Requests Ingredients

*Actors:*

WWWBrowser

*Preconditions:*

The HttpServer is running.

The XML recipes do not contain references to any XSL-T programs.

*Flow:*

1. The WWWBrowser requests *recipe.ingredients*
2. The HttpServer inserts a reference to the XSL-T program named *ingredients.xml* in the appropriate place in the XML recipe.
3. The HttpServer responds with the "augmented" XML.
4. The WWWBrowser discovers it needs *ingredients.xml*
5. The WWWBrowser requests *ingredients.xml*
6. The HttpServer responds with *ingredients.xml*
7. The WWWBrowser applies the XSL-T program to the XML recipe and displays the resulting HTML

## UC2. WWWBrowser Requests Utensils

*Actors:*

WWWBrowser

*Preconditions:*

The HttpServer is running.

The XML recipes do not contain references to any XSL-T programs.

*Flow:*

1. The WWWBrowser requests *recipe.utensils*
2. The HttpServer inserts a reference to the XSL-T program named *utensils.xml* in the appropriate place in the XML recipe.
3. The HttpServer responds with the "augmented" XML.
4. The WWWBrowser discovers it needs *utensils.xml*
5. The WWWBrowser requests *utensils.xml*
6. The HttpServer responds with *utensils.xml*
7. The WWWBrowser applies the XSL-T program to the XML recipe and displays the resulting HTML

### **UC3. WWWBrowser Requests Instructions**

*Actors:*

WWWBrowser

*Preconditions:*

The HttpServer is running.

The XML recipes do not contain references to any XSL-T programs.

*Flow:*

1. The WWWBrowser requests *recipe.instructions*
2. The HttpServer inserts a reference to the XSL-T program named *instructions.xml* in the appropriate place in the XML recipe.
3. The HttpServer responds with the "augmented" XML.
4. The WWWBrowser discovers it needs *instructions.xml*
5. The WWWBrowser requests *instructions.xml*
6. The HttpServer responds with *instructions.xml*
7. The WWWBrowser applies the XSL-T program to the XML recipe and displays the resulting HTML

### **UC4. WWWBrowser Requests a Recipe in HTML:**

*Actors:*

WWWBrowser

*Preconditions:*

The HttpServer is running.

The XML recipes do not contain references to any XSL-T programs.

*Flow:*

1. The WWWBrowser requests *recipe.recipe*
2. The HttpServer applies *instructions.xml* to the recipe.
3. The HttpServer responds with the result of the transformation.

# Use Case Descriptions for the Client

The use case descriptions below involve the following actors: Device and HttpServer. Note that, since the “intelligent” devices don’t yet exist, for testing purposes, an operating system command shell will play the role of the Device.

## UC5. KIttyClient Requests a Recipe in XML

*Actors:*

Device and HttpServer

*Preconditions:*

The Device is ready to run the KIttyClient.

*Flow:*

1. The Device starts the KIttyClient with arguments that indicate the recipe to request and the HttpServer of interest.
2. The KIttyClient requests *recipe.xml* from the appropriate HttpServer (identifying itself in the header of the request).
3. The HttpServer responds with the recipe in XML.
4. The KIttyClient extracts the settings for the appliance of interest.
5. The KIttyClient displays the settings on the Device.

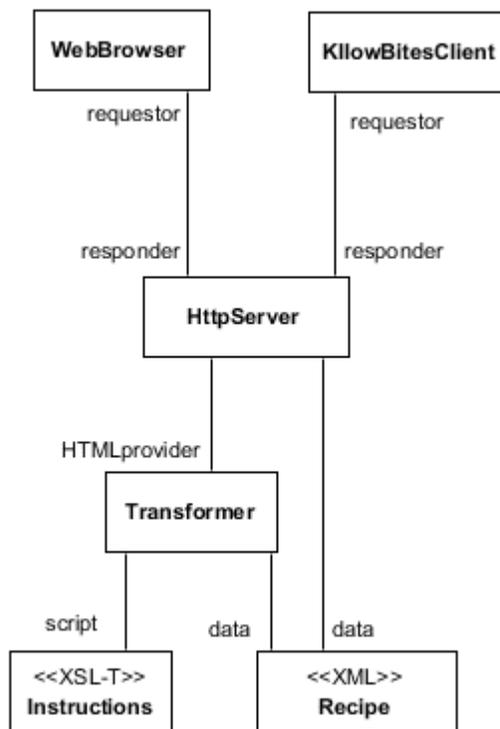
# Requirements Specification

## Abbreviations and Acronyms

HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
WWW	World Wide Web
XML	Extensible Markup Language
XSL-T	Extensible Stylesheet Language Transformations

## Conceptual Model

The following conceptual model was developed early in the design process:



It illustrates, at a fairly high level, the important concepts in the system (note that the WWW browser is not part of the system). It is not intended as an engineering design. That is, the software components that need to be designed and developed may or may not correspond to these concepts.

## Physical Requirements for the Server

The server must satisfy the following physical requirements:

- PR-1** The system must have an XSL-T program named `ingredients.xsl` that can be used to transform an XML recipe into an HTML file containing only the list of ingredients in the recipe.
- PR-2** The system must have an XSL-T program named `utensils.xsl` that can be used to transform an XML recipe into an HTML file containing only the list of utensils used in the recipe.
- PR-3** The system must have an XSL-T program named `instructions.xsl` that can be used to transform an XML recipe into an HTML file containing only the list of instructions in the recipe.

## Operational Requirements for the Server

The server must satisfy the following operational requirements:

- OR-1** The HTTP server must store recipes only in XML.
- OR-2** The HTTP server must be able to handle requests (from WWW browsers that support XML and XSL-T) for ingredients. Specifically, it must insert a reference to `ingredients.xsl` into the the XML recipe and then respond to the client. (see Needs U-1)
- OR-3** The HTTP server must be able to handle requests (from WWW browsers that support XML and XSL-T) for utensils. Specifically, it must insert a reference to `utensils.xsl` into the the XML recipe and then respond to the client. (see Needs U-2)
- OR-4** The HTTP server must be able to handle requests (from WWW browsers that support XML and XSL-T) for instructions. Specifically, it must insert a reference to `instructions.xsl` into the the XML recipe and then respond to the client. (see Needs U-3)
- OR-5** The HTTP server must be able to handle requests (from all WWW browsers) for instructions in HTML format. Specifically, it must process the XML using `instructions.xsl` and deliver the result. (see Needs U-4)
- OR-6** The HTTP server must be able to handle requests (from KIttyClients) for recipes in XML format. Specifically, it must include the XML recipe (without modification) in the response. (see Needs S-1)

## Physical Requirements for the Client

The client must satisfy the following physical requirements:

- PR-6** The system must use HTTP. (see Needs S-1)

## Operational Requirements for the Client

The client must satisfy the following operational requirements:

- OR-7** The system must be able to request recipes in XML format from an HttpServer running HTTP.  
(see Needs S-1)
- OR-8** The system must be able to extract the information that is relevant to a particular appliance from a recipe in XML format. (see Needs S-2)