

Cooking Special – Requirements Document

Product Description

Summary

Cooking Special is a web-based recipe and ingredient database accessible to the general public. It seeks to provide the user with a single portal to access a suite of **user-provided low-budget recipes** as well as a method by which the user can cross-reference the ingredients of those recipes with the websites of well-known food suppliers, allowing the user to **acquire those ingredients as cheaply as possible**. Moreover, it allows the user to search for price data on specific ingredients and then to reference recipes that have those ingredients, allowing the user to utilize the items he or she may already own.

Target Audience

The product is directly aimed at the **food consumer on a budget**. The product's ability to locate cheap and convenient recipes, as well as sales on ingredients for those recipes, is incredibly helpful for anyone attempting to survive on a small income. College students in particular may benefit greatly from this product, as many college students have the time to comparison shop or travel to a further store but must be frugal when buying food. They may then stock up on a particular item determined to be on sale, and then locate several recipes with which to use that item up. A consumer is not saving money if he or she can save a large amount of money on a given item, but can only cook so many variations of a given recipe before throwing the rest of an ingredient out.

The Problem and the Solution

Thus this product eliminates a question encountered by shoppers everywhere: "If I buy this, will I use it?" Food is no good if it won't be used. Most frequently a customer discovers that a given item is on sale by seeing it on the shelf, and thus doesn't have the ability to look up recipes containing the item. The customer thus runs the risk of purchasing something that will not be used. *However* if they had used Cooking Special to search for recipes before going out shopping, the user would have known (1) what they wanted to make, and (2) where to get it at a good price. This eliminates the problem of buying potentially low-performing foods.

As with most applications of technology there are less efficient means to the same end. A customer could go through the sales sheet of each of his or her favorite stores and look for good deals, then look up recipes in his or her own cookbooks that use those ingredients. However, this process is tedious and error prone. A user is limited to his or her own cookbooks and must attempt to remember which books have which recipes. Moreover, a user has to also check the prices of each of the other ingredients of a recipe in order to determine whether the entire recipe can be made cheaply. This is in itself another search through the sales sheets. This entire ordeal wastes the user's time.

Comparison to Other Products

Cooking Special is different because it grants instant knowledge to the user about what deals are currently available. The user can query the site for only the ingredients he or she is really interested in. Not only that, but the user is no longer limited to just the recipes he or she happens to have lying around. The user-created Cooking Special recipe database will be full of recipes that the user may not have known about, each of which has been submitted because of how inexpensive it is to make. Upon seeing an ingredient on sale the user can determine which recipes use that ingredient, or vice versa, without having to manually go through each sales sheet and cookbook.

Alternatives

The common alternatives include budget recipe websites and Albertson's website. One such budget recipes website is **allrecipes.com**. This website has an extensive set of user shared and proprietary recipes. Moreover, it has a special budget section on recipes whose ingredients can be bought under some fixed dollar amount. The users can save their favorite recipes for future use and also add all the ingredients of a recipe to their shopping list (which can be printed). The **Albertsons website** even goes further by collaborating with allrecipes.com where users can directly use their shopping list to check out items from a local grocery store. Another website, **cheapcooking.com**, provides users with a basic calculator that can calculate savings but requires physical coupons.

Competitor Strengths:

- Albertsons website does allow you to add ingredients locally.
- An extensive database for recipes is available.

Competitor Weaknesses:

- Albertsons specials are limited to only ONE brand store.
- Although Albertsons website does provide with a convenient way to add ingredients to shopping lists and check them out; there is no way interface for selecting a recipe based on cheap ingredients. The user still has to manually go through weekly savings and then search for recipes based on those ingredients.

Limitations

Granted, Cooking Special does have some limitations. It relies on user-generated content, which may be less reliable than traditional cookbooks. More importantly it doesn't have the ability to find coupons,

which can sometimes save much more money than the regular sale. This data simply isn't available in a form that can be used by the database, as each coupon website deliberately forms the data in a way that is impossible to parse without proprietary software. This may cause the very poorest of consumers to use traditional means to find food, but for most consumers this limitation is not considered to be too limiting.

Main features:

- The user will be able to search for a food or term in the recipe database, and be able to see results that match that query. This result page will list the recipe name and ingredients. Each ingredient in each recipe will have a link which is equivalent to a corresponding sale query (see below)
- The user will be able to search for an ingredient or term in the sale database, which will return a list of all ingredients which match that query. Each entry will have any sales information found in the database, as well as a link to search for that ingredient in the recipe database.
- The user will be able to log in and create a saved profile (see other features)
- The user will be able to keep a list of commonly used foods which has equivalent links to a recipe, allowing the user to easily search for commonly used foods. This page also includes a "search all" function, which searches for deals on every item in the list.
- The user will be able to specify a list of foods that are not desired. (e.g. foods that are not kosher)

Minor features:

- Users can 'bookmark' recipes that are used often
- Users can rate recipes posted by other users

Documentation for Users

What external documentation will you provide that will enable users to understand and use your product? This could take the form of help files, a written manual, integrated help text throughout the UI, etc.

The website has two main methods for finding food specials:

- Finding recipe(s) based on a query.
- Looking at the results of a recipe search, and browsing information on the specials associated with each of the recipes' ingredients.
- Searching the Specials database directly for a desired food or ingredient.

As these are the main functions of the website, there will be messages in appropriate places on the website indicating what the user should do to carry out their desired task. For instance:

- The recipe search box will be accompanied by a label such as: “Search for Recipe by Name or Ingredient”.
- Each recipe result will be accompanied by an arrow and the text: “4 Specials for this recipe”, which can be clicked in order to see more information on specials. Thus, it will be clear to the user how to get information on Specials for each recipe.
- The specials search box will be accompanied by a label such as: “Search for Specials by Food/Ingredient”.
- Each specials result will contain information on how to take advantage of the special, and the information will be complete enough to be self-explanatory.
- On the home page, there will be a “Get Started” section which indicates that the user can search for specials, or search for recipes. In other words, it must be visually clear that the user should initially progress through the site flow by searching for specials or recipes using a textual query.

Additionally, there will be a readily-accessible How To section which includes a walk-through of a sample usage of the site, including searching for deals on the ingredients of a given recipe.

Process

Software Toolset

Programming Languages:

- PHP and MySQL will be used for the website back-end.
- Perl will be used to parse weekly specials from online HTML sources.
- XHTML and CSS will be used for the website front-end.
- To import supplemental recipes to the site, we will use Perl if the source is unstructured data, or may take advantage of open source recipe databases, which provide their own interfaces.

Data Sources:

- Recipes will be supplied by the user, and by importing data from recipe sites and open source recipe databases.
- Weekly Specials will be parsed from the websites of companies such as Albertson’s and Safeway, which provide weekly specials in text form.

Version Control:

- We will use the Subversion provided by CSE 403 for source control for all code files.
- Documents other than code (such as requirements and specification) will also be submitted to source control.

Bug Tracking:

We will keep track of bugs using an online (Google Documents) spreadsheet with fields for bug category, description, submitted by, assigned to, submission date, and link(s) to relevant files such as images.

Tools:

- We will take advantage of frameworks for handling the website's back-end, such as a framework that takes care of MVC code structure, database interface, and templating interface.
- We will take advantage of IDEs for various tasks, such as Eclipse for PHP, an HTML/web design editor, etc. These will become clearer as the work pans out, but there is no need for highly specialized tools for this project, as the area of web development has been well-covered.
- We will require an image editor such as Adobe Photoshop in order to create images for the web site design.

Group Dynamics

Roles

Project Manager – This role will be filled by John-Paul. The main job for this role is to coordinate the actions of the rest of the team, and make sure that (1) the project is progressing on schedule, (2) manage risks as far as each development task is concerned, (3) adjust the time line, features, roles, and/or whatever else is needed to assure that the project is completed on time, and (4) make sure that all group members have no blocking issues, and are working towards specific and measurable goals.

Development – Our development will fall into several categories:

- **Website, Back-end development.** This includes code done in PHP using a MySQL database. It will do the job of retrieving information on user queries, and passing that information on in a predefined format to the templating system.
 - John-Paul will provide expertise in all aspects of this section, and will determine the framework and conventions that other members will follow when developing this section. Alex Phillips will code this section as he has experience in PHP and MySQL. Halil will also work on this section in collaboration with Alex Phillips.
- **Website, Front-end development.** This includes design and code done in XHTML and relevant technologies, and will need to communicate with the Back-end development section in order to determine (1) what data is needed by the front-end to display, and (2) how to submit data for search queries and user preferences changes.
 - John-Paul will provide expertise in this area. Paramjit will develop this section, and will call on Halil or Alex Phillips for help if/when needed.
- **Data gathering.** This includes code done in Perl. It will retrieve (1) Weekly Specials from sites such as Albertsons and Safeway, and (2) recipes from recipe sites. The data gathering effort may require the use of other programming languages in order to take advantage of open source recipe databases. This section will need to interface with the back-end development section in order to coordinate data formats and transfer.
 - Alex Eckerman and Ben will develop this section. Together they will work out which data sources to use, how they can best be parsed. They will communicate with website, back-end in order to determine how to transfer parsed data to the database.

Testing – Testing will be done by all members of the team. The testing effort will focus on the correct implementation of the website’s UI, correctness of the back-end’s user preferences and search results sections, and also on the integrity and consistency of weekly specials data.

Schedule/Timeline

- **All three sections will be developed simultaneously.**
- The data gathering section can proceed without the other two sections, except for deciding how to share the data. However, this is not time-sensitive as data is not needed during the development process.
- The website’s front-end and back-end will work together closely. However, while they will collaborate on the format of the data, the design of the UI forms and graphics may proceed independent of the back-end section.
- The **parsing section** should take three weeks to beta.
- The **website back end** should take three weeks to beta.
- The **website front end** should take three weeks to beta.
- After the three sections are in a beta state, further changes can be made as time permits.
- Because of the nature of the parsing section, if the parsing section is finished earlier than the website front and back ends, the members of the parsing section can help with testing and possibly developing (if ramp-up time is not too long) the website sections.
- Beta features include all of the Major Features (see Major Features section).
- Post-beta features include the Minor Features (see Minor Features section).

Risks

Major Risks

- **Parsing Data for the weekly specials**

Based on our preliminary research we have concluded that there isn’t any online fixed interface for retrieving information on weekly specials. However, QFC, Safeway, and Albertsons do have a HTML based web interface where users can browse through weekly specials. Therefore we have to rely on accessing these web interfaces and parsing the raw HTML to get the required data we want. This interface is slightly different for different stores. Moreover, these web interfaces use dynamic web pages with cookies to present weekly specials data for local stores. As such we will have to simulate storing and retrieving of these cookies based on the zip code. Since our major strength of this product is the weekly specials/coupon deals, if we are unable to retrieve weekly specials data then our product is simply unusable.

To mitigate this risk, we need to make a small script program that will retrieve the data from these web pages and see if can really parse them.

The timeline for this risk to be tested is right before we actually start to code and are still working on the requirements document.

Other Pertinent Risks

1) Ensure finality of UI design

It is important that we determine the actual impact of the major risk (parse-ability of weekly specials), so that any changes to the website UI can be made as early as possible.

Furthermore, we should scrutinize the UI as a group in order to identify any potential issues that may crop up and require change.

2) Ensure usability of web interface

Since this product is targeted towards relatively college students, who are majorly non computer science, we need to make sure the user interface is intuitive to them as well.

To mitigate this risk, after we have made a (usable) prototype of the user interface, we should definitely experiment it with all the team members to make sure it is intuitive, and also have a non technical person like a family member or a college friend test out the interface.

This should take place on two occasions: (1) as soon as there are accurate UI specifications, and (2) after the web development has been started, as soon as there is a working interface.

3) Development Falls Behind schedule

It is the Project Manager's job to manage this risk. Actions taken could include feature cuts, adjusting group roles/dynamics, time schedule, and reduction on testing.

4) Recipe data store is not available

Since we are relying on an offline recipe database, we might not be able to find such a database. Therefore we can no longer present any recipes to the user just basic weekly specials. A simple experiment to confirm this will be to find an offline recipe database that might have even a small amount of recipes.

This risk should be investigated early on in the project, as it has potential impacts on the web site's front-end design.

Finally if we are not able to get a recipe data store, then we will come up with our own recipes and store them, enabling us to move forward. Additionally we can rely on our users to add their custom recipes which can help populate our recipe database.

Use Cases

Actors

Website and System actors will be further expounded in the technical specification.

- **User** (logged-in | new-user) – refers to the
- **Website** – refers to the graphical user interface with which the user uses Cooking Special's various features. This is accessed through a web browser.
- **System** – The Cooking Special servers. This entity serves data to the user's browser, and also accepts input from the user/browser.

Cases

User searches for a recipe [See Figure 1 and Figure 2]

- **Precondition:** the user has the website loaded on their browser.
- **Actors:** Computer, User
- **Success guarantee:** the user finds a recipe and determines if any ingredients are on sale
- **Flow:**
 - The user begins at the website.
 - The user then types in a recipe name and presses enter.
 - The user is presented with a list of recipes
 - Clicking on the name of the first recipe brings up another page, which shows the ingredients as being "Cherries" and "Cheesecake". "Cherries" is listed as being on sale at Albertsons.
- **Alternate conditions:**
 - No recipes are found – Show "0 results found" instead of result list.
 - No items are on sale – Recipes are shown, but no link to specials.

User searches for an ingredient [See Figure 3]

- **Precondition:** the user has the website loaded on their browser.
- **Actors:** Computer, User
- **Success guarantee:** the user finds a recipe and determines if any ingredients are on sale
- **Flow:**
 - The user begins at the website.
 - The user clicks on the "Ingredients" tab
 - The user then types in an ingredient name and presses enter.
 - The user is presented with a list of all items that are on sale that match the query, followed by all recipes that have that ingredient
 - Clicking on the name of the first recipe brings up another page, which shows the ingredients as being "Cherries" and "Cheesecake". "Cherries" is listed as being on sale at Albertsons.
- **Alternate conditions:**

- No sales are found – Show “0 specials found”.
- No recipes are found – Show “0 recipes found”.
- No sales or recipes are found – Combine previous two messages.

User logs in

- **Precondition:** the user has the website loaded on their browser.
- **Actors:** Computer, User
- **Success guarantee:** the user is logged into the site for future actions
- **Flow:**
 - The user begins at the website.
 - The user changes the focus to the login box
 - The user then types in his/her username and password
 - The user presses enter
 - The logged in status is indicated in the login box
 - The user is now logged in for future actions
- **Alternate conditions:**
 - The password does not match the username. The user will have to reattempt login.
 - The username is not in use. The user will be redirect to a registration page.

Figure 1 – Recipe Search with 1 result:

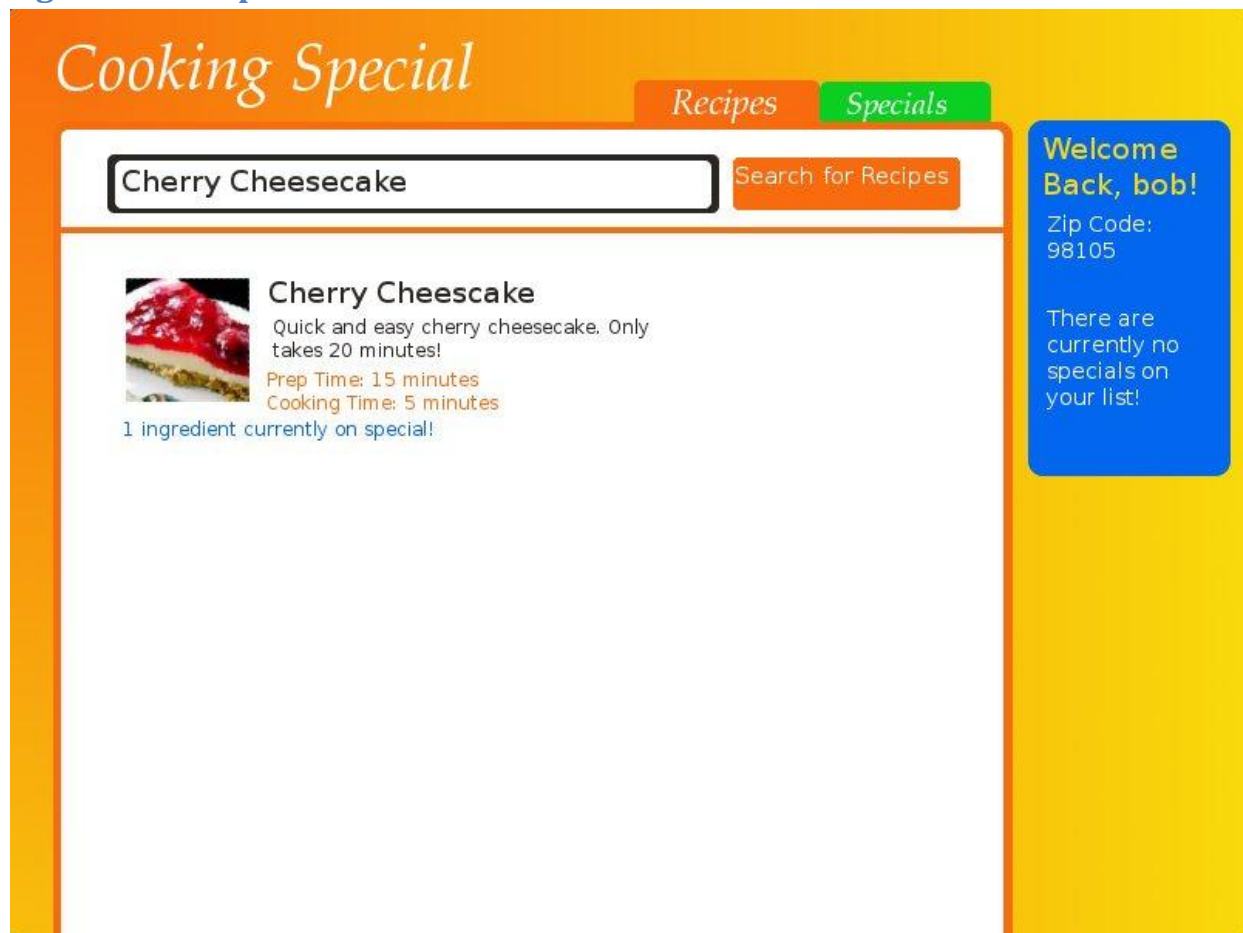



Figure 2 – Viewing Recipe:

Cooking Special

[Recipes](#)[Specials](#)

[Search for Recipes](#)



Cherry Cheesecake

Quick and easy cherry cheesecake. Only takes 20 minutes!

Prep Time: 15 minutes
Cooking Time: 5 minutes

Ingredients:

- 1 Cheesecake
- 1 lb Cherries

Cherries are \$1 per lb at Albertons!

Directions:

1. Preheat oven to 375.
2. Cook cheesecake until firm (~5 min)
3. Place cheesecake on a level surface and put cherries on top.
4. Refrigerate.

Welcome Back, bob!

Zip Code: 98105

There are currently no specials on your list!

Figure 3 – Ingredient/Special Search with

Cooking Special

Recipes


Specials

98105

Cherries

Search for Specials

Cherries are on sale at Albertsons!



Cherry Cheesecake

Quick and easy cherry cheesecake. Only takes 20 minutes!

Prep Time: 15 minutes

Cooking Time: 5 minutes