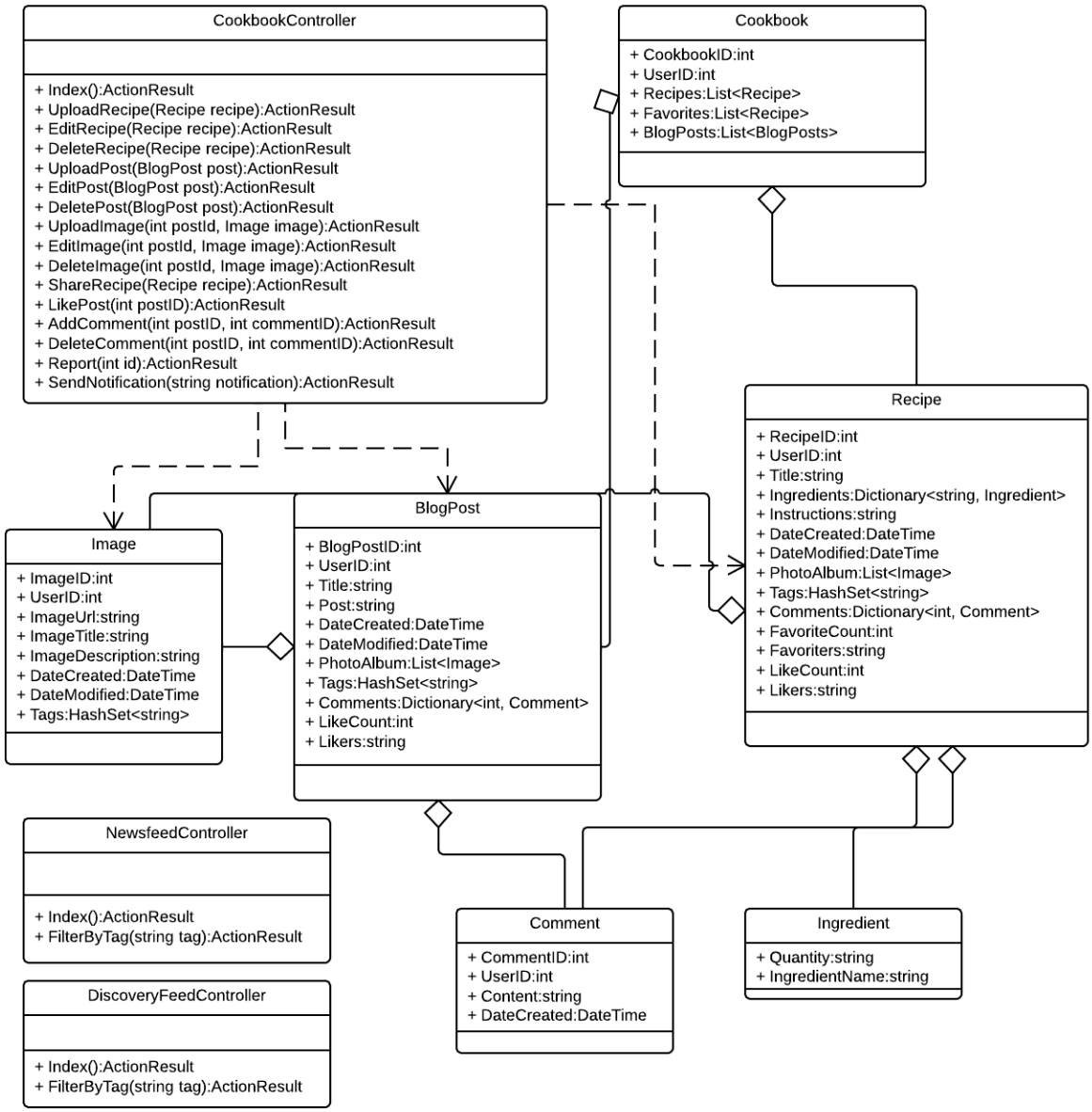
Lauren Ryan
Matthew Borowsky
Shahzore Qureshi
March 4, 2013

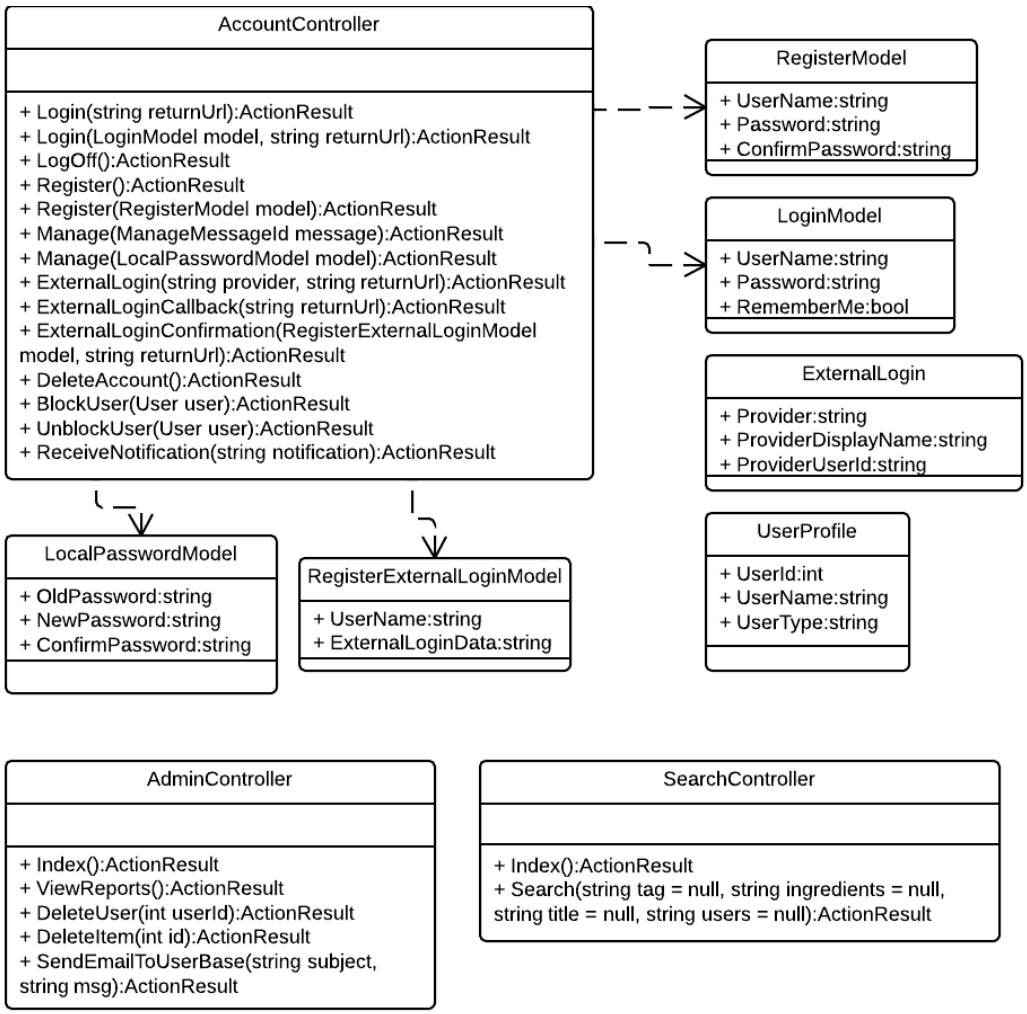
Wiki / Documents:
http://borowsky333.github.com/cookbook/

Design

Cookbook UML Diagram

Notes:
In ASP.NET, ActionResult (the return type for just about every function in the below controllers) is basically a catch all: these functions can return a view, partial view, or another Action (funtion).
Almost every function in these controllers will return a view or partial view. We are not picturing each view on the diagram, as they are not classes.
There are no getter/setters for any of these classes, because we are using autoproperties (which for simplicity are listed as attributes in the UML).



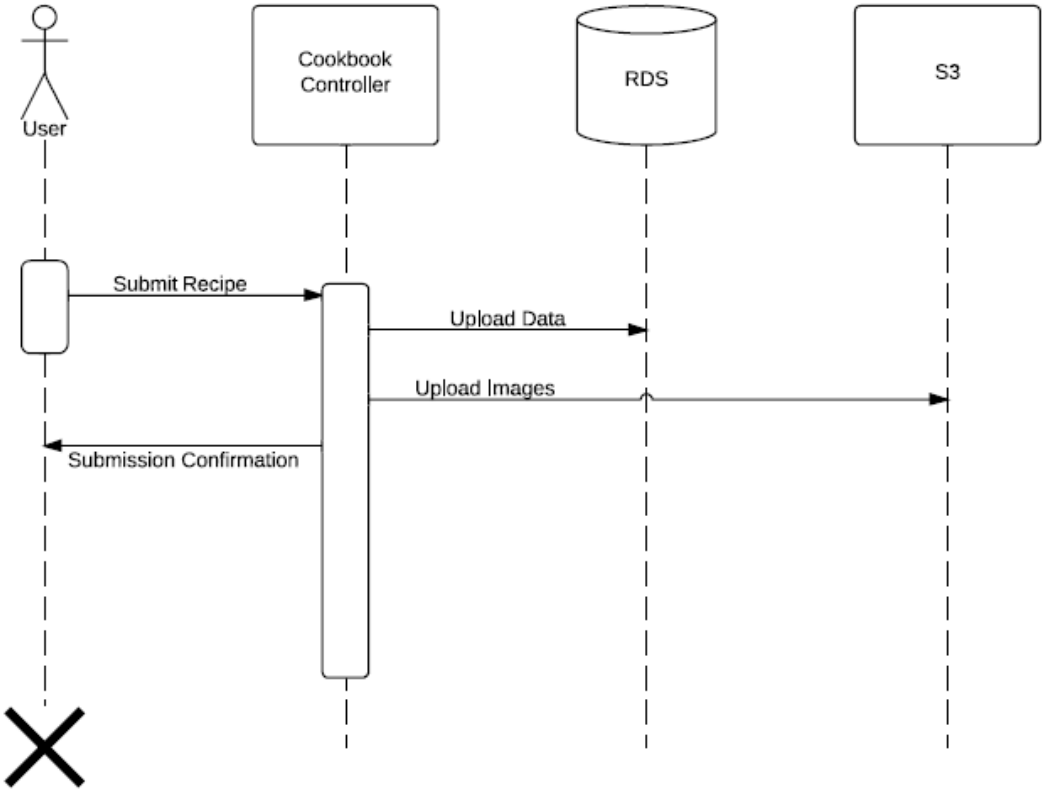

AccountController
+ Login(string returnUrl):ActionResult
+ Login(LoginModel model, string returnUrl):ActionResult
+ LogOff():ActionResult
+ Register():ActionResult
+ Register(RegisterModel model):ActionResult
+ Manage(ManageMessageId message):ActionResult
+ Manage(LocalPasswordModel model):ActionResult
+ ExternalLogin(string provider, string returnUrl):ActionResult
+ ExternalLoginCallback(string returnUrl):ActionResult
+ ExternalLoginConfirmation(RegisterExternalLoginModel model, string returnUrl):ActionResult
+ DeleteAccount():ActionResult
+ BlockUser(User user):ActionResult
+ UnblockUser(User user):ActionResult
+ ReceiveNotification(string notification):ActionResult
RegisterModel
+ UserName:string
+ Password:string
+ ConfirmPassword:string
LoginModel
+ UserName:string
+ Password:string
+ RememberMe:bool
ExternalLogin
+ Provider:string
+ ProviderDisplayName:string
+ ProviderUserId:string
UserProfile
+ UserId:int
+ UserName:string
+ UserType:string
LocalPasswordModel
+ OldPassword:string
+ NewPassword:string
+ ConfirmPassword:string
RegisterExternalLoginModel
+ UserName:string
+ ExternalLoginData:string

AdminController
+ Index():ActionResult
+ ViewReports():ActionResult
+ DeleteUser(int userId):ActionResult
+ DeleteItem(int id):ActionResult
+ SendEmailToUserBase(string subject, string msg):ActionResult

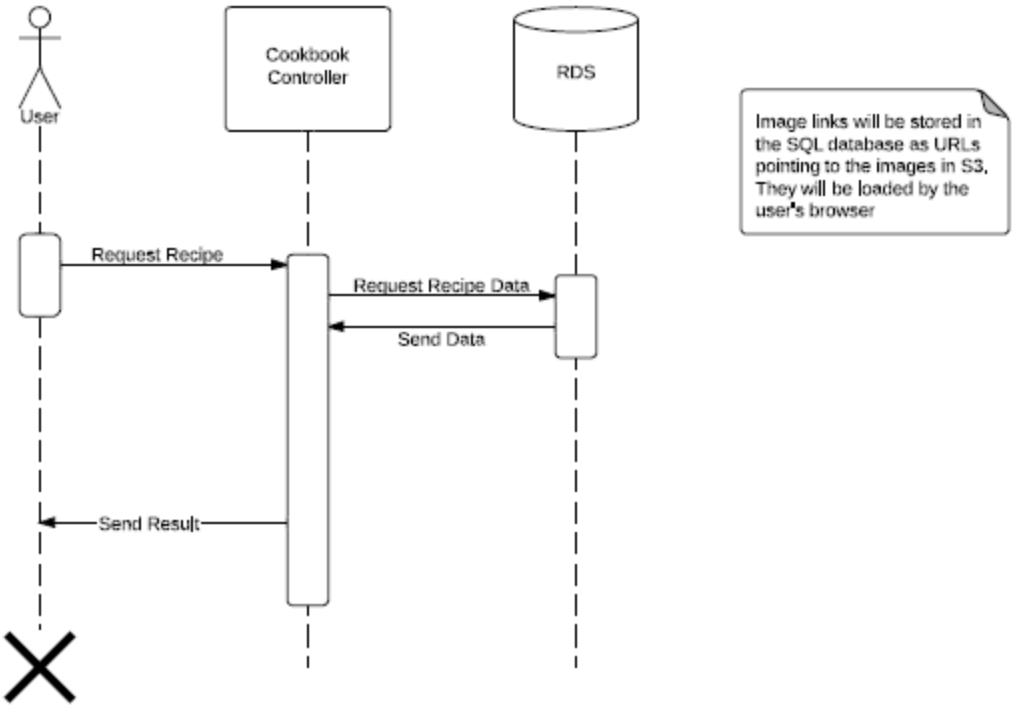
SearchController
+ Index():ActionResult
+ Search(string tag = null, string ingredients = null, string title = null, string users = null):ActionResult

System Sequence Diagrams

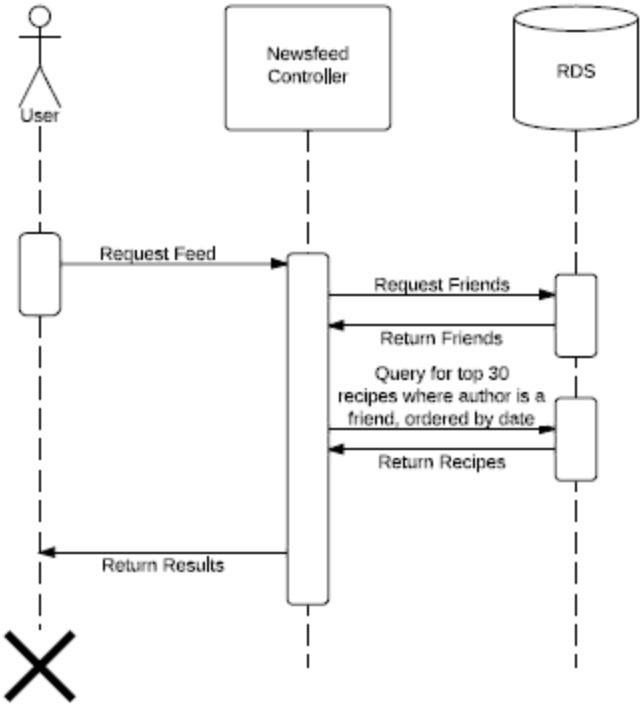
Submit Recipe



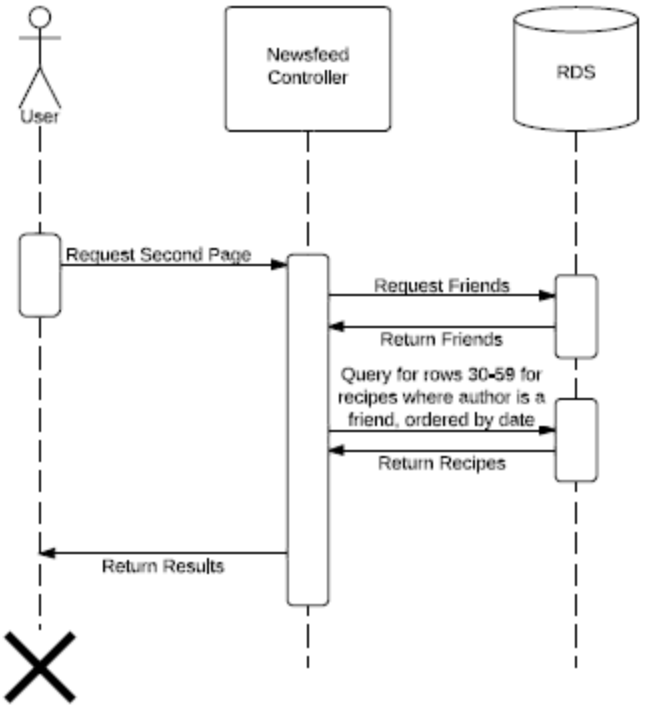
View Full Recipe



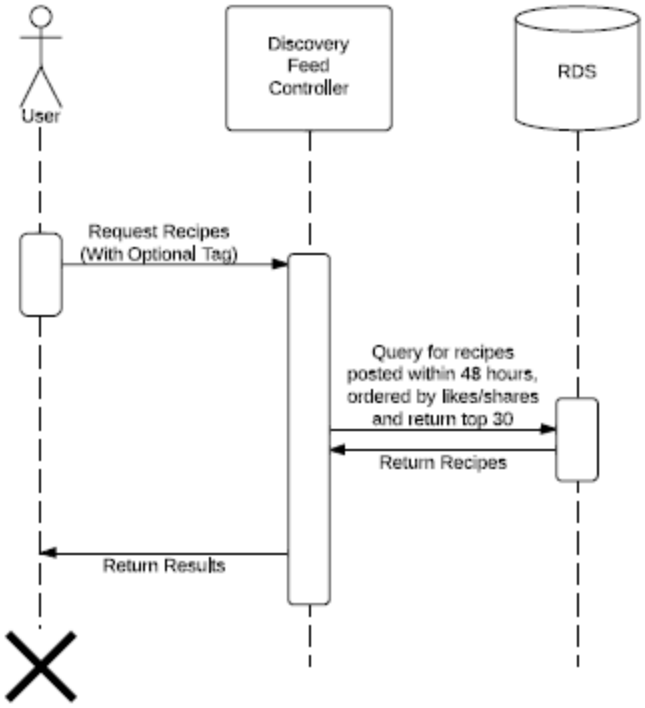
View Recipe Feed



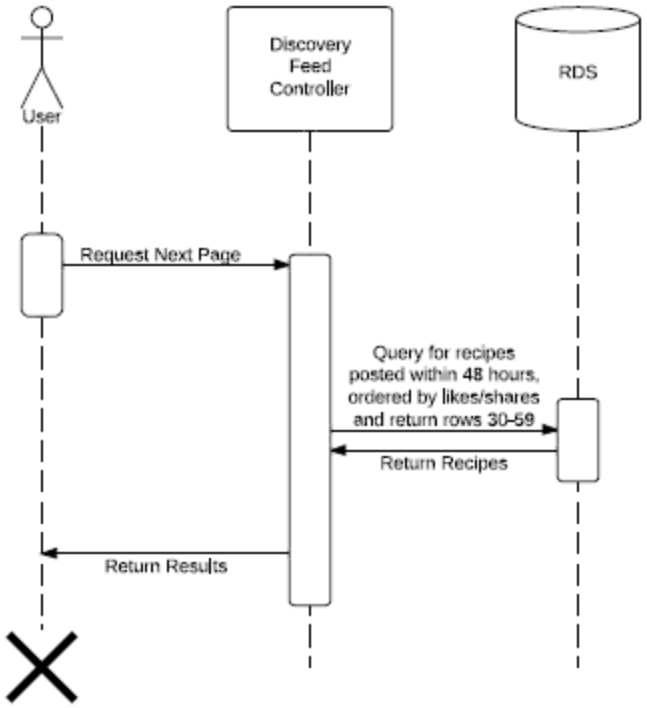
View Recipe Feed (Second Page)



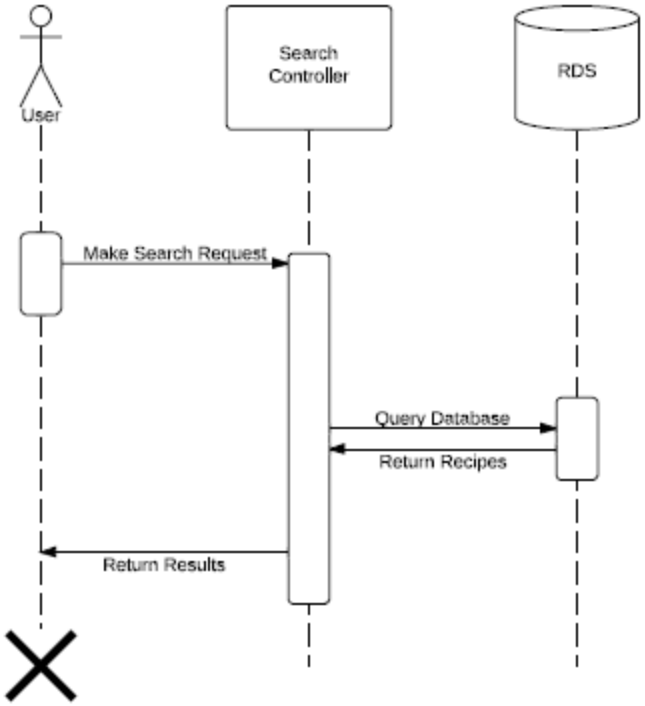
Discover Recipes



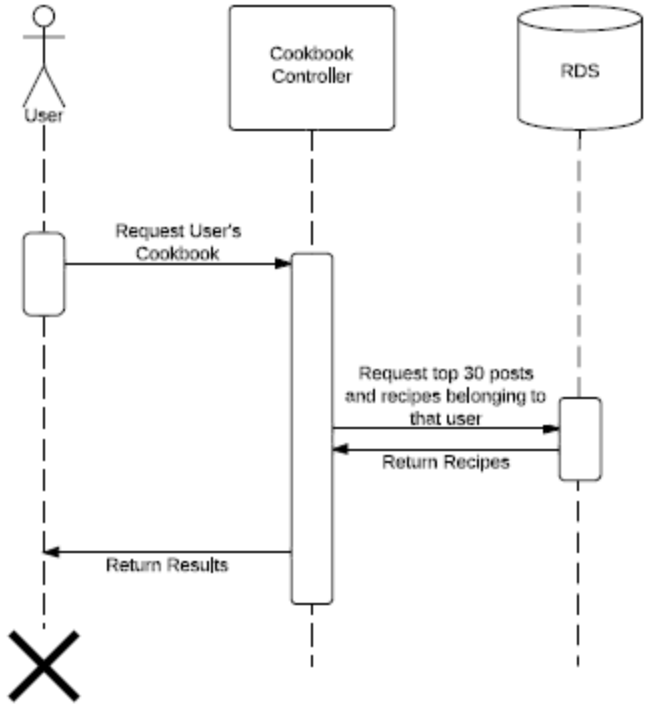
Discover Recipes (Page Two)



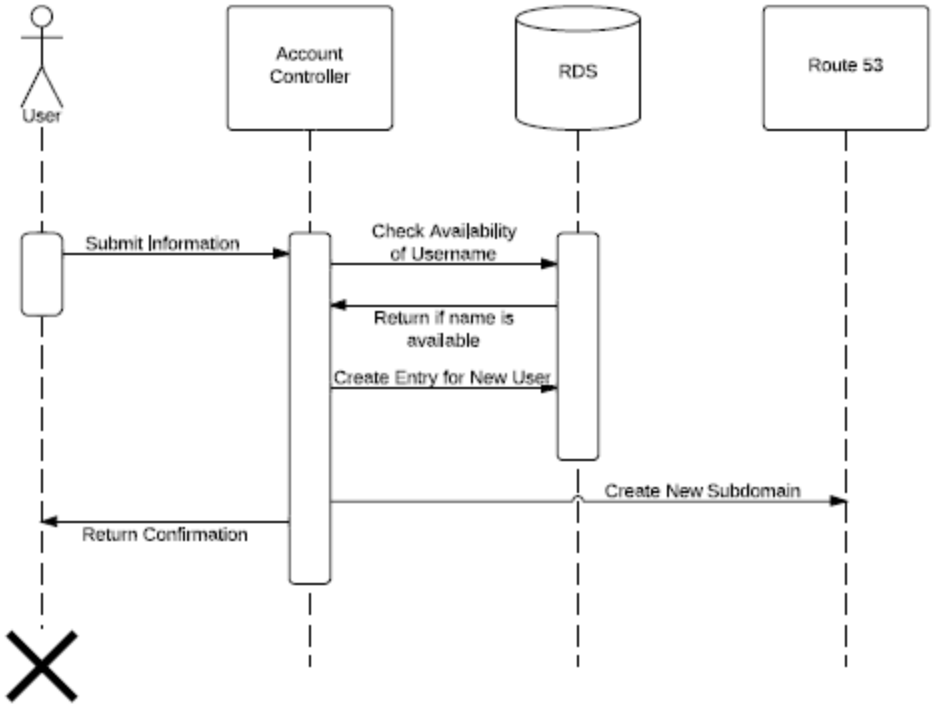
Search Recipes



View Cookbook



Register New User



User Interface Design

The user interface of The Cookbook strives for simplicity and overall user satisfaction with the product by implementing Shneiderman's "Eight Golden Rules of Interface Design".

The first golden rule is striving for consistency. Interface design should make use of the same terminology and buttons for every action or sequence of actions. Cookbook follows this rule by implementing a navigation bar that allows the user to browse through the entire website via clicking navigation bar buttons and links. This bar also stays in the user's view when the user scrolls down any of the site pages. It is always located at the top of the screen.

The second golden rule is creating shortcuts that allows an experienced user to easily jump back and forth between different sections of the product. The Cookbook site allows users to create custom news feed filters that allow them to narrow their news feed to posts and recipes that include certain keywords and hashtags. At the top of the news feed page, there exists a row of custom filters where each filter changes the content of the news feed when they are clicked on. To expedite the process of altering the news feed content, Cookbook can start building a content cache for each filter as soon as the user logs in. By implementing such a system, users do not need to search for the same hashtags and keywords every time they visit the site.

The third golden rule of Shneiderman's list of interface design rules is offering informative feedback. To follow this rule, Cookbook has successful, unsuccessful, and confirmation messages that are displayed to the user whenever an action is taken. For example, when a potential user attempts to create an account with a username that is already taken, Cookbook alerts the user via highlighting the username field in red and displaying the error message next to it. Also, when a user is unable to create a post or recipe on the server due to network connectivity issues, a timeout error is printed out to the user.

The fourth golden rule is walking through multi-step actions and providing feedback at the end of each step. In general, the focus of the Cookbook interface is to minimize the number of steps necessary to complete any action. Most actions only require one step to complete. However, by necessity, Cookbook has a few multi-step actions, such as two-step account verification via email. In such a case, the website lets the user know that an email was sent to the user's email address and asks the user to open the verification link in the email in order to complete the account verification action.

The fifth golden rule of interface design is simplifying error handling. Cookbook offers informative feedback to the user for any type of error that occurs. In the cases where the user is at fault, the website provides detailed information about the error and a possible solution to that error. For example, if a user tries to access the site via an Internet browser that is outdated and that is not supported, Cookbook informs the user with the error statement and a solution, such as a download link to a supported Internet browser. Another example.

The sixth golden rule is providing an easy way to undo actions. The website allows users to

create and delete their own comments that are made on other people's posts and recipes. Cookbook does not allow users to edit their comments because it adds unnecessary complexity to the entire system. If a user does not like a previous comment, the user can delete it and write another one. Cookbook also allows users to create, edit, and delete their photos, recipes, and posts. If the user feels that a recipe or a post is too complex or poorly worded, he or she can go back and revise it.

The seventh golden rule is making the user feel in control of the website. Cookbook allows members of the site to create all of the content that powers the entire website. Without users, there are no recipes or posts. Cookbook is simply a tool that allows users to create and share food recipes and discoveries with friends, family, and the rest of the Cookbook network. Members have full control over what they create and what they see.

The eighth and final golden rule of Shneiderman's list of interface design rules is reducing the amount of short-term memory needed to use the product. Cookbook follows this rule by minimizing the amount of steps (and web pages) needed to complete any action. Creating a recipe only requires one webpage. Making modifications to an existing post or recipe only requires one page. Querying the database for recipes and posts requires only one page. Viewing the news feed requires only one page. Both queries and the news feed refresh and load new data dynamically as the user scrolls down those web pages. This eliminates the need for additional web pages for query results and news feed items. The only action that requires more than one web page and step is the account verification, which is handled via email. Cookbook needs to use two steps for account verification in order to maintain the integrity of posted data and the security of the website. All other website actions require just one step. Also, the implementation of a persistent navigation bar makes Cookbook much easier to browse through and explore. The user does not need to remember where he or she is on the website. Instead, that person can simply take a glance at the navigation bar to find his or her current location and choose another destination by clicking on a different button on the bar.