

Daniel Mizrahi
Austin Shi
Eric Jung
Jennifer Anthony

Deployment Document

1. Open Eclipse
2. Import project folder "Recipedia"
3. Open the project folder
4. Navigate to index.jsp
5. Right click on index.jsp
6. Run As...
7. Run on Server
8. Copy and paste the URL
9. Open Google Chrome
10. Past the link
11. Have fun!

Testing Document

Create Recipes

Test #	1
Description	User wants to add in ingredients in textfield
Input	Textfield is clicked on
Expected Result	A cursor can be seen and the user can start typing
Actual Result	

Test #	2
Description	User needs to add in more ingredients
Input	Add Ingredient button is clicked
Expected Result	A new textbox/textfield dynamically appears without reloading the page
Actual Result	

Test #	3
Description	As user types in ingredients, user can see a list of possible ingredients to autocomplete
Input	User starts typing in an ingredient
Expected Result	A list of possible ingredients is seen
Actual Result	

Test #	4
Description	User uses autocomplete to fill in ingredient
Input	User presses enter or clicks on an ingredient
Expected Result	The added ingredient can be seen in the textbox

Actual Result	
---------------	--

Test #	5
Description	User wants to add in instructions for the recipe
Input	Textfield is clicked on
Expected Result	A cursor can be seen and the user can start typing
Actual Result	

Test #	6
Description	User adds an image for the recipe
Input	Add image button is clicked
Expected Result	User can choose an image
Actual Result	

Test #	7
Description	User chooses an image and sees that the picture has been chosen
Input	OK button is chosen
Expected Result	User can see the image that was chosen on the create recipe page
Actual Result	

Test #	8
Description	User clicks on the submit button without filling

	out all the inputs: ingredients, instructions, and image
Input	Submit button is clicked
Expected Result	Gives an error message and tells user to finish filling out the recipe
Actual Result	

Test #	9
Description	User clicks on submit after all the fields are filled in
Input	Submit button is clicked
Expected Result	A new recipe page is created and the user can see the new page
Actual Result	

Test #	10
Description	User decides not to create a recipe anymore
Input	Cancel button is clicked
Expected Result	User is taken back to the main feed page
Actual Result	

Recipes Page

Test #	11
Description	User wants to add a review
Input	Click on a textbox: Add Review
Expected Result	A cursor can be seen and the user can start typing a review
Actual Result	

Test #	12
Description	User is ready to submit a review
Input	Submit button is clicked
Expected Result	The review can seen in the recipe page now
Actual Result	

Test #	13
Description	User clicks on one of the recommended recipes
Input	The recipe picture is clicked
Expected Result	User is taken to the recipe that was clicked on
Actual Result	

Test #	14
Description	Refreshing the page would give a different list of recommended recipes
Input	Refresh the recipe page
Expected Result	The same recipe page is shown, but with different recommended recipes
Actual Result	

Test #	15
Description	Clicking the recipe image would enlarge as a popup window
Input	Click image button
Expected Result	The image would enlarge with a short description about the food/recipe

Actual Result	
---------------	--

Test #	16
Description	Clicking the close(x) button would close the popup window
Input	Click 'x' button or press esc button
Expected Result	The popup would close and would see the recipe page as it originally was
Actual Result	

Sign up Window

Test #	17
Description	Clicking "Register" with all required fields filled in will officially sign the user up on the website
Input	Click 'Register' button
Expected Result	User will be redirected to the main page as an official user rather than a guest user
Actual Result	

Test #	18
Description	Clicking "Register" with "First Name" field blank will refresh the page, due to not filling out the registration form properly
Input	Click 'Register' button
Expected Result	Page will refresh, displaying the error message and prompting the user to fill in the invalid field properly
Actual Result	

Test #	19
Description	Clicking "Register" with "Last Name" field blank will refresh the page, due to not filling out the registration form properly
Input	Click 'Register' button
Expected Result	Page will refresh, displaying the error message and prompting the user to fill in the invalid field properly
Actual Result	

Test #	20
Description	Clicking "Register" with "Username" field blank will refresh the page, due to not filling out the registration form properly
Input	Click 'Register' button
Expected Result	Page will refresh, displaying the error message and prompting the user to fill in the invalid field properly
Actual Result	

Test #	21
Description	Clicking "Register" with "Password" field blank will refresh the page, due to not filling out the registration form properly
Input	Click 'Register' button
Expected Result	Page will refresh, displaying the error message and prompting the user to fill in the invalid field properly
Actual Result	

Test #	22
Description	Clicking "Register" with an already existing "Username" will refresh the page, due to

	entering in an already existing Username
Input	Click 'Register' button
Expected Result	Page will refresh, displaying the error message and prompting the user to enter in a different (and unique) Username
Actual Result	

Test #	23
Description	Clicking "Register" with the "Password" being less than 6 characters will refresh the page, due to entering a password that is too short
Input	Click 'Register' button
Expected Result	Page will refresh, displaying the error message and prompting the user to make a password that is at least 6 characters
Actual Result	

Test #	24
Description	Clicking "Register" with any additional optional info filled in + the required fields filled in properly will officially sign the user up on the website
Input	Click 'Register' button
Expected Result	User will be redirected to the main page as an official user rather than a guest user
Actual Result	

Sidebar

Test #	25
Description	Hovering the cursor on the left side of a page (where the right-side border of the sidebar is visible) will cause the sidebar to slide out

	from the left side
Input	Hover cursor on the left side of the site where the sidebar border is seen
Expected Result	Sidebar will slide out, displaying elements accessible through sidebar navigation
Actual Result	

Test #	26
Description	Clicking the profile icon on the sidebar will direct user to his/her profile page
Input	Click 'Profile' picture/icon
Expected Result	User will be directed to personal profile page
Actual Result	

Test #	27
Description	Entering a search term in the "Search Bar" field will pull up results relevant to the term that was entered
Input	Press "Search" button after making an entry
Expected Result	A results page will load that is relevant to the search term entered
Actual Result	

Test #	28
Description	Clicking the "Explore" option on the sidebar will direct user to a page that is loaded with random dishes stored on the database
Input	Click 'Explore' button
Expected Result	Page with random dishes/recipe combinations will load
Actual Result	

Test #	29
Description	Clicking the “Create Recipe” option on the sidebar will take the user to a page where he/she can create/upload/share a new recipe
Input	Click ‘Create Recipe’ button
Expected Result	User will be brought the the “Create Recipe” Page
Actual Result	

Test #	30
Description	Clicking the “Feed” or “Home” option on the sidebar will take the user to the feed/home page
Input	Click ‘Feed’ or ‘Home’ button
Expected Result	Main page/Personal feed will be displayed
Actual Result	

Test #	31
Description	Clicking the “Logout” option on the sidebar will Log the user out and reload the “Guest” mainpage
Input	Click ‘Logout’ Button
Expected Result	Login page or Guest main page will be loaded upon logging out
Actual Result	

Test #	32
Description	Hovering the cursor off the sidebar will cause the sidebar to slide back into the left and “hide” itself from full view
Input	Hover cursor off of the visible sidebar

Expected Result	Sidebar will slide back in, no longer being fully displayed on the left side
Actual Result	

Test #	33
Description	No user feed page loads generic recipes
Input	Right after user continues as guest
Expected Result	User is presented with top recipes
Actual Result	

Test #	34
Description	User feed page loads correct recipes
Input	Right after user logs in
Expected Result	User is presented with recipes from people they're fans of
Actual Result	

Test #	35
Description	Recipes display correctly
Input	When feed page is loaded
Expected Result	The recipes should show up in a square grid
Actual Result	

Test #	36
Description	New recipe
Input	A person the user is following adds a new recipe
Expected Result	The new recipe is shown as the first recipe in

	the feed page and all other recipes are moved down
Actual Result	

Test #	37
Description	Recipe image displays
Input	On loading main page
Expected Result	The correct image for the recipe displays in the recipe are
Actual Result	

Test #	38
Description	Liking recipe
Input	When the user clicks the like button on a recipe
Expected Result	The button gets checked off and the recipe gets added to the users liked recipes
Actual Result	

Test #	39
Description	Save recipe
Input	User clicks on save on a recipe
Expected Result	Recipe is added to user's saved recipes
Actual Result	

Test #	40
Description	Recipe clicked
Input	Users clicks on a recipe

Expected Result	The user is taken to that recipes page
Actual Result	

Test #	41
Description	Add to feed
Input	User clicks on recipes add to feed button
Expected Result	The recipe is added to the users feed
Actual Result	

Test #	42
Description	Sidebar
Input	Sidebar button clicked
Expected Result	The sidebar pops up
Actual Result	

Test #	43
Description	Sign up button appears
Input	When a user continues as guest
Expected Result	There should be a sign up button at the top of the page
Actual Result	

Test #	44
Description	Recipe unliked
Input	User clicks on an already liked recipe
Expected Result	The recipe is removed from users liked list, the liked button is unchecked
Actual Result	

Test #	45
Description	Recipe unsaved
Input	The user click on the unsave button on a recipe
Expected Result	The recipe goes from being saved to unsaved
Actual Result	

Test #	46
Description	Recipe removed from feed
Input	User clicks remove from feed button
Expected Result	The recipe no longer appears in the users feed
Actual Result	

Test #	47
Description	Recipe like toggled
Input	User clicks on recipe like button
Expected Result	The recipe changes from displaying liked to unliked or vice versa
Actual Result	

Test #	48
Description	Recipe save toggled
Input	User clicks on recipe save button
Expected Result	Recipe changes from displaying saved to not saved or vice versa
Actual Result	

Test #	49
Description	Recipe add to feed toggled
Input	Recipe add to feed button clicked
Expected Result	Recipes changes from displaying add to feed to remove from feed
Actual Result	

Login Window

Test #	50
Description	Nothing is entered
Input	No input
Expected Result	A textbox appears without reloading the page that says "Please enter a username and password"
Actual Result	

Test #	51
Description	Only the username is entered
Input	Enter the username field and leave the password field blank
Expected Result	A textbox appears without reloading the page that says "Please enter a password"
Actual Result	

Test #	52
Description	Only the password is entered
Input	Enter the password field and leave the

	username field blank
Expected Result	A textbox appears without reloading the page that says "Please enter a username"
Actual Result	

Test #	53
Description	Incorrect username
Input	User enters an invalid username
Expected Result	A textbox appears without reloading the page that says "Invalid username. Please try again"
Actual Result	

Test #	54
Description	Incorrect password
Input	User enters an invalid password
Expected Result	A textbox appears without reloading the page that says "Incorrect password. Please try again"
Actual Result	

Profile Page

Test #	55
Description	User wants to view the picture of whoever's profile it is
Input	Navigate to the profile page via several methods (Feed or Fans page)
Expected Result	Corresponding photo and name of the profile

	the user is trying to view
Actual Result	

Test #	56
Description	User wants to view another user's fans
Input	Navigate to the user's profile page
Expected Result	Corresponding photos and names of the profile's fans appear under the <i>Fans</i> heading
Actual Result	

Test #	57
Description	User wants to view a profile's fan's page
Input	Click the image of a user's fan
Expected Result	Navigate to user's fan's profile page
Actual Result	

Test #	58
Description	User wants to view all of the profile's fans
Input	Click view all under the <i>Fans</i> heading
Expected Result	Corresponding and complete list of the profile's fans appear
Actual Result	

Test #	59
Description	User wants to view another user's saved recipes
Input	A preview of the user's saved recipes appear upon navigation to the profile page

Expected Result	Corresponding recipes and images appear under the <i>Saved Recipes</i>
Actual Result	

Test #	60
Description	User wants to view a recipe listed under the <i>Saved Recipes</i> heading on the profile page
Input	Click an image under the <i>Saved Recipes</i> heading
Expected Result	Navigates to the corresponding recipe page
Actual Result	

Test #	61
Description	User wants to view all recipes listed under the <i>Saved Recipes</i> heading
Input	Click "View all" next to the <i>Saved Recipes</i> heading
Expected Result	Corresponding and complete list of that user's saved recipes appear
Actual Result	

Test #	62
Description	User wants to view another user's uploaded recipes
Input	A preview of the user's uploaded recipes appear upon navigation to the profile page
Expected Result	Corresponding recipes and images appear under the <i>Uploaded Recipes</i> page
Actual Result	

Test #	63
---------------	----

Description	User wants to view a recipe listed under the <i>Uploaded Recipes</i> heading on the profile page
Input	Click an image under the <i>Uploaded Recipes</i> heading
Expected Result	Navigates to the corresponding recipe page
Actual Result	

Test #	64
Description	User wants to view all recipes listed under the <i>Uploaded Recipes</i> heading
Input	Click “View all” next to the <i>Uploaded Recipes</i> heading
Expected Result	Corresponding and complete list of that user’s uploaded recipes appear
Actual Result	

Detailed Design Document

Hardware/Software Requirements:

Hardware:

Windows: Windows 8 (Desktop) Windows 7 Windows Vista SP2 RAM: 128MB

Disk Space: Recommended 512MB Processor: Minimum Pentium 2 266MHz processor

Mac OS X: Intel-based Mac running Mac OS X 10.8.3+, 10.9+

Linux: Oracle Linux 5.5+ Oracle Linux 6.x (32-bit), 6.x (64-bit) Oracle Linux 7.x (64-bit) Red Hat Enterprise Linux 5.5+ (32-bit), 6.x (64-bit) Ubuntu Linux 12.04 LTS, 13.x

Software:

Java 8

Eclipse IDE for Java EE Developers (Neon)

MySQL

Chrome

Algorithms used:

```
matchIngredientsWithRecipe(){
    Backtracking algorithm that looks for the first ingredient listed
        If the recipe has that ingredient, keep it - if not, throw away
            From remaining recipes, look for next ingredient
                If the recipe has that ingredient, keep it - if not, throw away
                    etc.
}

matchTagWithRecipe(String tag){
    Loop through all recipes and search for the tag the user types in
        If tag is found
            Add recipe to list
        Return list
}
```

Database (schema):

User

- VAR CHAR(50) First name
- VAR CHAR(50) Last name
- VAR CHAR(50) Username
- VAR CHAR(50) Password
- VAR CHAR(300) Image
- PK Saved recipes

- PK Uploaded recipes
- PK My fans → needs method to figure out fans of the user

Saved Recipes

- FK User

Uploaded Recipes

- FK User

Recipe

- PK Ingredient
- INT(10) Quantity of ingredient
- PK Reviews
- PK Instructions
- Recommendations
- PK Tags

Tags

- FK Recipe
- Tag

Reviews

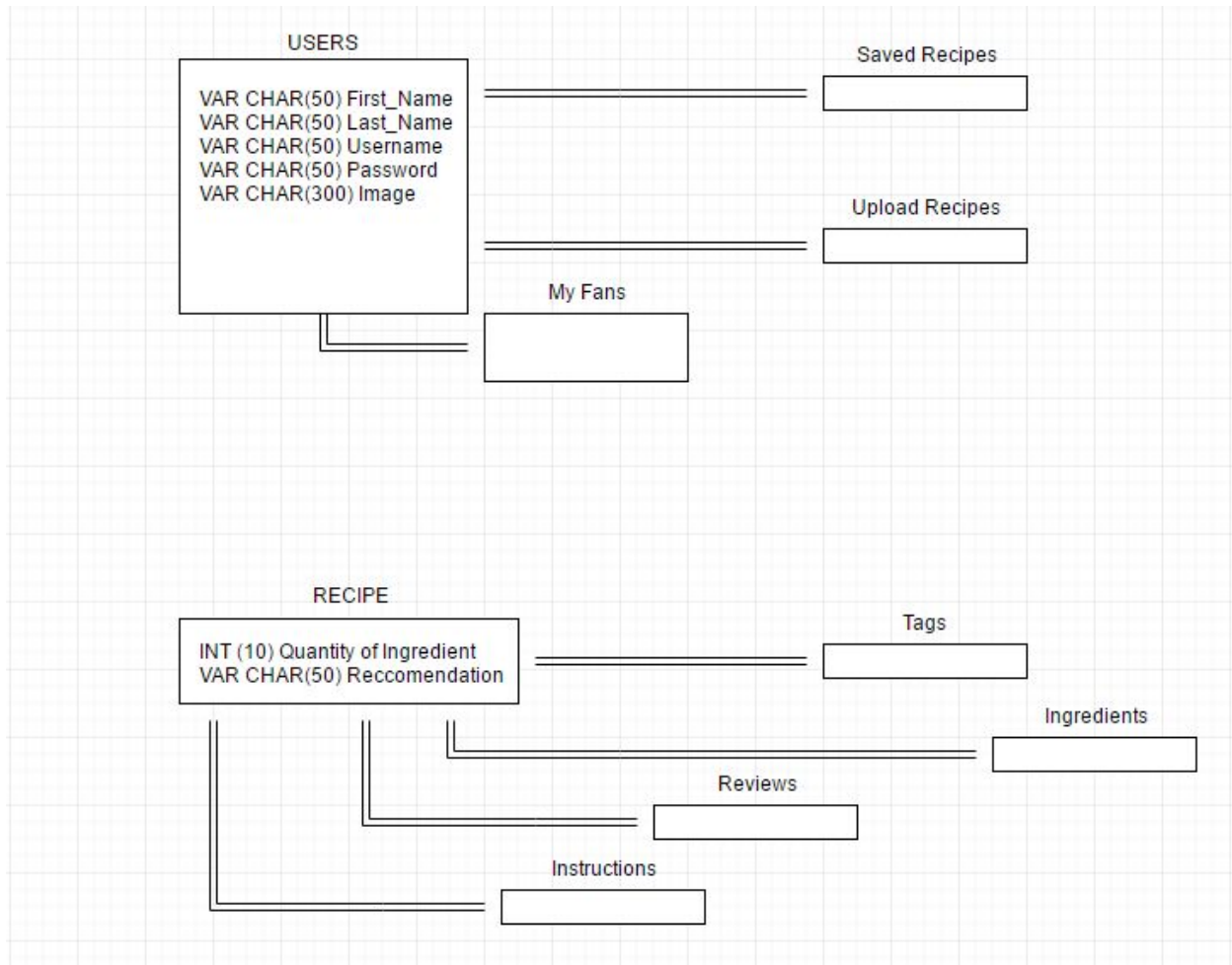
- FK Recipe

Instructions

- FK Recipe

Ingredients

- VAR CHAR(50) Name of ingredient



GUI (JSPs used):

Login Window

1 hr

<p>	id = "welcome-text"
<form> <input type ="text"> <input type ="text"> </form>	id = "login-textbox" name = "username" name = "password"
<p>	id = "continue-as-guest"

Profile Window

3 hr

(3 tables - one for saved, uploaded, and fans)

	id = "profile-image"
<table> <thead></thead> <tr> <td> <button>View All</button> </td> </tr> </table>	id = "saved-recipes-table"/"uploaded-recipes-table"/"fans-table" class = "recipe-image"/"fans-image"

Feed window

3 hr

https://www.w3schools.com/w3css/w3css_sidebar.asp

(3x3 squares with recipe images)

<button></button> <table> <tr> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr>

 onclick="openSidebar()" class = "recipe-container" class = "recipe-image" class = "recipe-image" class = "recipe-image" class = "recipe-image" class = "recipe-image" |

<pre> </td> <td> </td> <td> </td> </tr> </table> </pre>	class = "recipe-image" class = "recipe-image"
---	--

Create Recipe Window

2 hr

<h3>Step 1: Add Ingredients</h3> <form> <input type="text"/> <input type="text"/> <input type="button" value="Add ingredient"/> </form>	class="header" class="ingredient-input" class="ingredient-input" id="add-ingredient-button"
<h3>Step 2: Add instructions</h3> <form> <input type="text"/> </form>	class="header" Id="input-instructions"
<h3>Step 3: Add image</h3> <form> <input type="file" accept="file_extension image/*"/> </form>	class="header"

Recipe window

2 hr

	id="recipe-image"
<h3>Ingredients</h3> <h3>Instructions</h3> 	class="header" class="ingredient" class="ingredient" class="header" class="instruction" class="instruction"

<pre> <h3>Reviews</h3> <p></p> </pre>	class="header"
<pre> <h3>Recommendations</h3> <table> <tr> <td> </td> </tr> </table> </pre>	id="rec-image"

Backend (Classes):

User

- String First name
- String Last name
- String Username
- String Password
- String Image
- Vector<Recipe> Saved recipes
- Vector<Recipe> Uploaded recipes
- Vector<String>My fans
- Void setFullName(String name)
- String getFullName()
- Void setFName(String name)
- String getFName()
- Void setLName(String name)
- String getLName()
- Void setUserName(String userName)
- String getUserName()
- Void setPassword()
- String getPassword()
- Void setImage()
- String getImage()
- Void setSavedRecipes(Vector<Recipe>)
- Vector<Recipe> getSavedRecipes()
- Void setUploadedRecipes()
- Vector<Recipe> getUploadedRecipes()
- Void setFans(Vector<String>)

- Vector<String> getFans()
- Void addFan(String)
- Void removeFan(String)
- Vector<String> getFansOf()

Recipe

- Vector<Ingredient> Ingredients
- Vector<String> Reviews
- Int likes
- String name
- Vector<String> Instructions
- Vector<String> Tags //holds the user searches for recipes (actual foods)
- Void setIngredients(Vector<Ingredient>)
- Void addIngredient(Ingredient)
- Void removeIngredient(Ingredient)
- Vector<Ingredient> getIngredients()
- Void setReviews(Vector<String>)
- Void addReview(String)
- Vector<String> getReviews()
- Void setLikes(int likes)
- Int getLikes()
- Void setInstructions(Vector<String>)
- Vector<String> getInstructions()
- Void addInstruction(String)
- Void deleteInstruction(String)
- Void setTags(Vector<String>)
- Vector<String> getTags()
- Void addTag()
- Void removeTag()
- Void setName(String name)
- String getName()

Ingredient

- String name
- String units
- Int quantity
- Void setName(String name)
- String getName()
- Void setUnits(String units)
- String getUnits()
- Void setQuantity(int quantity)
- Int getQuantity()
- String toString()

LoginServlet

Void service() //responds to front end saying if login was successful or not
If not successful responds with error

int CheckLogin() //checks to see if the login was successful. Returns 0 if yes, 1 if wrong user, 2 if wrong password

SignupServlet

Void service() //handles requests from the front end. If the user entered everything in and username doesn't exist already, creates new user and tells the front end it succeeded.

Otherwise responds with appropriate error

String parseFirstName(String name) //returns first name in textfield user entered

String parseLastName(String name) //returns last name

Boolean userExists(String userName) // checks to see if the user exists already

Void createUser(String userName, String password, String fName, String lName, String image)

CreateRecipeServlet

Void service() //handles requests from the front end. Checks to see if all fields are filled in and valid, if they are, creates recipe responds with success, otherwise responds with error

Void createRecipe()

SearchRecipeServlet

Void service() //handles requests from the front end

Recipe getRecipe(Recipe recipe, int searchType) //returns recipe if it exists, null otherwise. Int specifies if there's a different search type (useful in the future if we want to search by different parameters than just recipe name)

Vector<String> getPossibleRecipes(Vector<Ingredients>) //returns recipes that have ingredients in common with the ones the user entered

Frontend:

The frontend is going to mostly based on JSPs while using javascript functions and servlets.

Script tags would include methods that would call the servlet or do a response to a form. These are methods that are caused by an onsubmit or an onclick.

Login window

The login window is going to have a form where the user is going to input their username and password. When the user clicks on a submit button, it would call a validate() method where it would make a AJAX call to the servlet and validate if the user puts in the correct input.

Additionally, there would be a onclick function if the user decides to continue as a guest.

Signup window

The signup window essentially does the same thing as the login window, but there would be different inputs where the user can enter in their information such as name, username, and password. Like the login window, the signup window is also going to have a onsubmit action where it would call a Javascript function (validate()) inside a script tag where it would do a validation to see if the user exists, if password does not meet the criteria, or any other form invalidations.

Feed window

Depending whether or not if the user is a guest or a logged in user, there would be different functionalities that would affect what methods would be needed in the frontend in the script tags. If the user chose to be a guest, there is going to be a onclick function where the user can click the sidebar in which the user can choose want to do such as explore or look for recipes. However, due to the limited functionalities for the guest, the frontend would mostly redirect to different JSP pages.

If the user is logged in, there would be more functionalities and more ways for the user to interact in the feed page. Each recipe shown in the feed would have a button where the user can like, rate, save, and add to the user's feed. Each of these would be a button and have an onclick function where it would add to corresponding data structures and change the information of each recipe info and user's profile. Additionally, the sidebar navigation would also include full functionalities, which like the the guest's sidebar would mostly redirect to JSP pages.

Profile window

The profile window is going to display all of the information about the user that was clicked on. The page includes a profile picture and the feed of the user: saved/uploaded recipes and how the user reacted and rated to certain recipes. Here there would be an onclick function where if the user clicks on 'view All' to see all the user's fans, we would use an AJAX function to show more fans rather than reloading the entire page.

Create recipe window

As the user adds in ingredients and instructions when creating a recipe, there is going to a button where it would say add ingredient or add instruction where it would call an onclick function where it would dynamically add in a textbox using AJAX, so that we would not need to reload the entire page.

Recipe window

The frontend for the recipe window would be based on the information on the recipe; it would list the ingredients and the steps to make the recipe. Additionally, the frontend is going to include a recommended recipes portion in which the backend would send information to the frontend to give users more options on similar recipes on the current recipe that the user is on.

High Level Specs

Front-End/Design

When someone first gets to the website, they're greeted by a log-in page, with the option to continue as guest. From there, if they choose to sign in they're taken to the main page. This page is a feed. Here the user sees what recipes the people he's/she's following have posted recently, displayed in a list style. From this feed via the top navigation bar, a user can go to a recipe creator page. This page allows the user to add ingredients, quantities of each ingredient, and then type in steps to create the recipe. For adding ingredients, the user sees an area where they can type in the ingredient, another to type in the quantity, and a button to submit this. Once the ingredient is submitted, it appears below this input form in a list style. Below this is another input area to type in the steps. This is simply a textbox with a submit button, which once submitted is displayed below the input form, like with ingredients. There is a submit button at the bottom for when the user finishes the recipe. Additionally, there's a recipe search page, where a user can search for a recipe by name. The recipes that pop up are clickable, and show the user that create them on the side. There is also a browse section, where recipes can be searched by category. In addition, there's another recipe search feature, except for this one, the user can type in the ingredients that he/she has, and then can showed recipes that can be made using those ingredients. The ingredients here are typed in, in a similar fashion to in the recipe creator page. When looking at a recipe, below the actual recipe, related recipes can be displayed to users.

Functionality

A user with an account should be able to do the following:

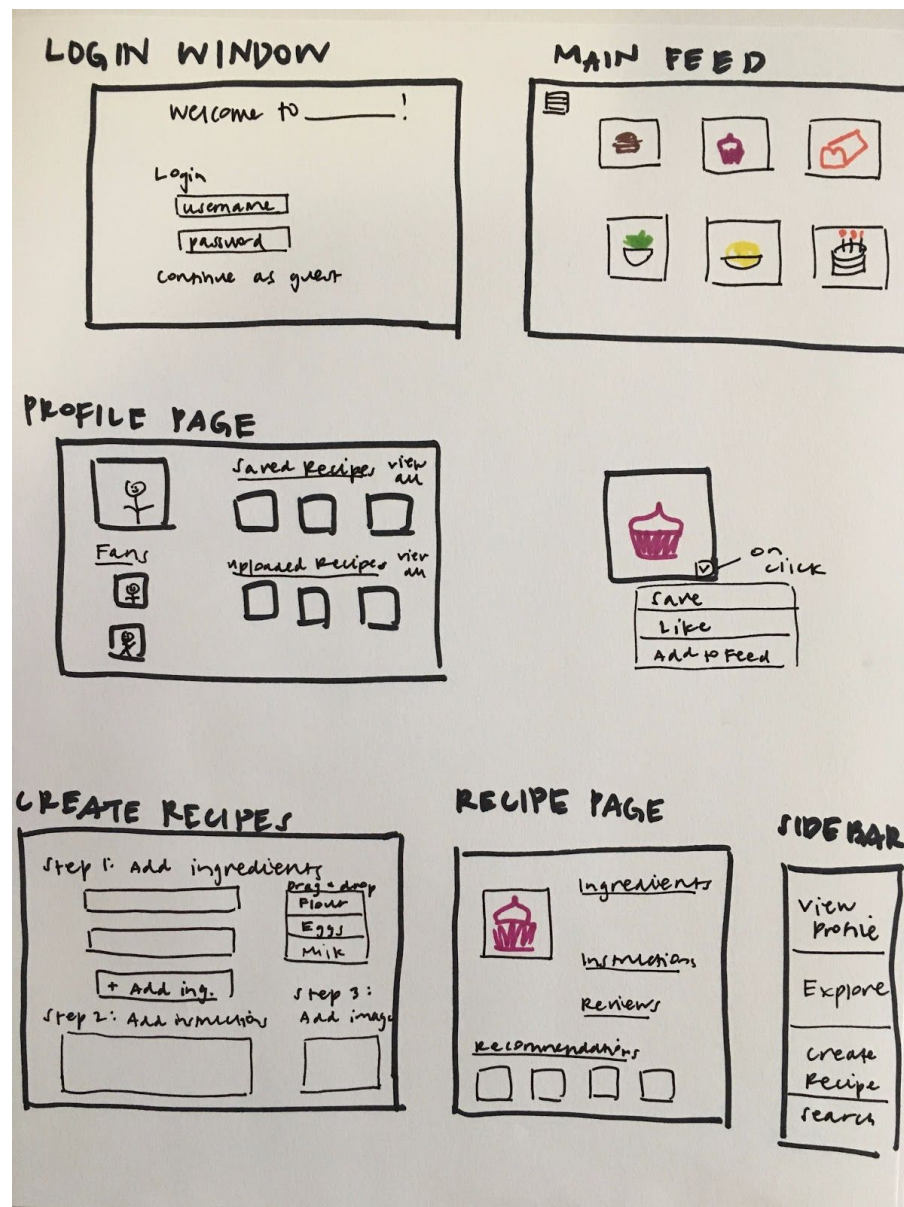
1. Login and access pinned recipes (from other users they follow and their own recipes)
2. Save and create their own recipes; whether through discovery, other users, or via external online sources
3. Users can comment, like, and post pictures of their own renditions of the recipe
4. The recipe creator page will allow users to add ingredients, the quantities of each ingredient, and a form for step-by-step instructions
5. Users can put their recipes on several backgrounds with different fonts, borders, and designs for printing if they would like
6. The users can share their own recipes or pinned recipes to other social media platforms such as Facebook, Instagram, and Twitter
7. Users can view similar recipes at the bottom of the page based on past interests or the current recipe they are looking at
8. Users can search for recipes given a list of ingredients, category, or occasion

A user without an account should be able to do the following:

1. Access a generic feed with top recipes
2. Search for recipes given a category or occasion

3. Will not have the ability to save/bookmark specific pages or recipes through a guest account
4. Access other miscellaneous content unrelated to unique user benefits, but will unable to take full advantage of such functions

Technical Specifications



LOGIN WINDOW (4 hours)

The user is first directed to a sign-in page. They can choose to log-in, sign-up or continue as guest. If they choose to sign-up, they're taken to a sign-up page where they're asked to enter a username and password. If they choose to log-in and provide valid credentials, they're taken to the main page, otherwise an error is displayed. The backend handles the form validation. If they choose to sign in as guest, they're taken to this same main page, but with guest access only.

MAIN FEED (Guest) (6 hours)

The main feed for the guest will have a sign-in button at the top (to encourage them to sign in). It will display top recipes of users. They can't like, comment, or share these posts, but they can see the recipes. If they click on the sidebar, it slides over the screen. The sidebar displayed is a guest version of the sidebar with limited features. The backend determines which posts are to be shown on the front end based on the most popular recipes.

MAIN FEED(Signed in) (4 hours)

It's similar to the normal main feed, but there isn't a log-in button at the top now (the log-out button will be in the sidebar). In addition, the user now sees recent recipes from people they're fans of instead of top-posts, but there's a bar at the top that lets them switch. The can now fully interact with the recipe squares. Opening the side-bar now opens the sign-in version of the side-bar with all of the functionality. The backend determines which posts are shown on the front end based on the what mode the user has selected. For the fans version of the feed, it looks at the people this user is a fan of to compile this list of recipes they see.

RECIPE SQUARE (2 hours)

The user will be able to click on a drop down menu in the bottom corner of each recipe square on the feed and choose to save it for the recipe to show up on their profile, like it to increase the recipe's rating, or add it to their feed so that it will consistently appear on the feed. The backend handles any action that occurs in this square and updates the database.

PROFILE PAGE (4 hours)

The user will see their profile picture on the top left corner of the page. To the right, there will be a preview of saved recipe and a preview of uploaded recipes. A user can click to view all, where they will be directed to a page with all of their saved or uploaded recipes. On the left, there will be a preview of the user's "fans", which can also be clicked to view all. The user will be able to click on the profile image or a recipe image to go to the profile page or recipe page (described below). The backend provides all of the saved/ uploaded recipes, and who the persons fans are.

CREATE RECIPES (8 hours)

If a user decides to create a recipe, they will be taken to a page where they would be writing out the instructions on how to make the recipe. A user would fill in textboxes to add in ingredients with a button functionality, giving the user more options to add in ingredients. Another feature that may be added would be a drag and drop option for ingredients. For example, if a user wants to create a recipe on cupcakes, there would be a sidebar of common ingredients in cupcakes such as flour, eggs, sugar, etc. Additionally, as the user types in the ingredients, there would be a feature where it would autocomplete the ingredient. The next thing the user would do is to add the step by step instructions, where they would be adding this to a textbox. Finally, the user is also able to upload an image, so that other users can see how the final product looks like. The backend processes this form, validates it, and updates the database appropriately.

SIDEBAR NAVIGATION (6 Hours)

Users will have the ability to navigate the website through a sidebar, providing links to pages such as the Profile Page, Explore Page, and other useful subpages that cover every important functionality the website has to offer. This sidebar will either be implemented as a pop out that will slide into full view on hover or on click. The sidebar will also have a search function where users can enter in a username or miscellaneous keywords to search up (such as dishes, ingredients, etc). Users will also be able to sign out through the sidebar by adding a separate action or clicking on their profile picture displayed on the sidebar. The backend determines which actions are available to the user depending on whether they're signed in or not.

RECIPE PAGE (8 hours)

When a user clicks on a recipe, they're taken to a recipe page where they're shown how to create this recipe. On the left, there's an image of the final dish that they'll be preparing. To the right of that, they can see the ingredients that are required to make the recipe, along with how much of each one they need to make it. Below that, they can see all of the steps they need to take to create the recipe. Finally, at the bottom, there's a place where users can leave reviews or see other users reviews. The user must be signed in to leave a review. Below that, there's a recommended recipes page that shows the user recipes that are similar to this recipe(using food and recipe API to obtain these recommendations). The backend will use this api to deliver the recommended recipes to the front end, and it will send the recipe data to the front end from the database.