

Team Names: Sampurna Basu, Sagar Punhani, Kevin Wang, Matt Wenner

Product Name: Entree

Product Summary: Social recipe web app aimed at college students.

<i>Test #</i>	01
<i>Items Under Test</i>	Main Page Class, getMorePosts(), updateNewsFeed()
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	The main news feed is able to repopulate after the user creates a new recipe manually on their personal page
<i>Steps to run test</i>	<ol style="list-style-type: none">1. Create a new recipe in Profile class2. Add a post object to the main news feed of the user
<i>Expected Result</i>	The main news feed of the user updates in the GUI

<i>Test #</i>	02
<i>Items Under Test</i>	Personal Page Class
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	The user's personal news feed is able to repopulate after the user creates a new recipe manually on their personal page
<i>Steps to run test</i>	<ol style="list-style-type: none">1. Create a new recipe in Profile class2. Add a post object to the user's personal feed class
<i>Expected Result</i>	User's personal feed updates in the GUI

<i>Test #</i>	03
<i>Items Under Test</i>	Main Page Class
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	The user's main news feed is able to repopulate after the user reheats a post in the main feed
<i>Steps to run test</i>	<ol style="list-style-type: none">1. User shares a recipe in the main news feed/other user's

	personal page 2. The user's main news feed updates with post object
<i>Expected Result</i>	The user's main feed has their new post at the top

<i>Test #</i>	04
<i>Items Under Test</i>	Personal Page Class
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	The user's personal news feed is able to repopulate after the user shares a recipe from the main news feed
<i>Steps to run test</i>	1. The user hits the share button on main news feed/other user's personal page 2. Add a post object to the user's personal feed
<i>Expected Result</i>	The user's personal feed updates properly in the GUI

<i>Test #</i>	05
<i>Items Under Test</i>	Main Page Class
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	The user's followers' main news feeds are able to repopulate after the user manually creates a new recipe on their personal page
<i>Steps to run test</i>	1. The user creates a new recipe in their Profile Class 2. Add the post object to each follower's main feed
<i>Expected Result</i>	The followers main feed updates in the main feed

<i>Test #</i>	06
<i>Items Under Test</i>	Main Page Class
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	The user's followers' main news feeds are able to repopulate after the user shares a new recipe from their main news feed
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. The user shares a recipe from the main feed/other user's personal page 2. Post object is added to every follower's main news feed
<i>Expected Result</i>	The follower's main feed updates in the GUI

<i>Test #</i>	07
<i>Items Under Test</i>	Main Page Class
<i>Test Type</i>	White Box
<i>Test Description</i>	The user's main newsfeed is populated in the correct order (chronologically, most recent is at the top).
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Refresh page 2. Check if the main newsfeed is in the correct order (based on time of post)
<i>Expected Result</i>	Main news feed is populated by posts in chronological order

<i>Test #</i>	08
<i>Items Under Test</i>	Main Page Class, updateNewsFeed()
<i>Test Type</i>	Unit Test
<i>Test Description</i>	The updateNewsFeed() method updates the member variable main news feed class, which is a list of posts

<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Call function after a post is shared or created 2. Navigate to main news feed
<i>Expected Result</i>	Check if member variable updates in the right order

<i>Test #</i>	09
<i>Items Under Test</i>	Personal Page Class, Post Object, List of Posts, getMorePosts()
<i>Test Type</i>	Unit Test
<i>Test Description</i>	When a recipe is created, the recipe object is constructed and placed inside the recipe class (list of posts)
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. The user manually creates a recipe 2. Database is updated with new recipe
<i>Expected Result</i>	Check if the getMorePosts() method is called and properly updates the list of posts in the database

<i>Test #</i>	10
<i>Items Under Test</i>	Personal Page Class, Post Object, List of Posts, getMorePosts()
<i>Test Type</i>	Unit Testing
<i>Test Description</i>	When a recipe is shared, the profile class responds by calling getMorePosts()
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. The user shares a recipe from a feed 2. Personal page object getMorePosts() is invoked
<i>Expected Result</i>	Check if the getMorePosts() method is called and calls updateNewsFeed() in the profile class

<i>Test #</i>	11
<i>Items Under Test</i>	Personal Page Class, addManualRecipe
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	When the user input information to create a new recipe, the

	addManualRecipe() function is called and consequently calls the appropriate methods.
<i>Steps to run test</i>	1. Have a user input recipe information
<i>Expected Result</i>	the addManualRecipe function is called with calls getMorePosts() on the profile and updateNewsFeed() in the main news feed

<i>Test #</i>	12
<i>Items Under Test</i>	Personal Page Class, addManualRecipe
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	When the user input information to create a new recipe, the addManualRecipe() function is called and consequently calls the appropriate methods.
<i>Steps to run test</i>	1. Have a user input recipe information 2. Click on confirm button after inputting proper text
<i>Expected Result</i>	the addManualRecipe function is called with calls getMorePosts() on the profile and updateNewsFeed() in the main news feed

<i>Test #</i>	13
<i>Items Under Test</i>	Personal Page, Follow Button
<i>Test Type</i>	White Box
<i>Test Description</i>	When you click follow button, the database updates accordingly
<i>Steps to run test</i>	1. User clicks a follow button on other user's personal page 2. Check if user is notified that they are now following this person
<i>Expected Result</i>	The friend graph data structure will be updated with the new one way relationship

<i>Test #</i>	14
<i>Items Under Test</i>	Main Page Class, Personal Page Class
<i>Test Type</i>	Black Box, White Box

<i>Test Description</i>	Navigation to Main NewsFeed
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. User is currently on main news feed or user is currently in their personal page or user is currently viewing the personal page of another user 2. User clicks on the Entree logo at the top of the page 3. The page switches to the main news feed - updateMainPage() is invoked 4. Most updated version of main news feed is displayed
<i>Expected Result</i>	Click the main logo to return to the user's main news feed.

<i>Test #</i>	15
<i>Items Under Test</i>	Main Page Class, Personal Page Class
<i>Test Type</i>	Black Box, White Box
<i>Test Description</i>	Navigation to Personal Page
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. User is currently on main news feed or user is currently in their personal page or user is currently viewing the personal page of another user 2. User clicks on their name at the top of the page 3. The page switches to the main news feed - updatePersonalPage() is invoked 4. Most updated version of the user's personal page feed is displayed
<i>Expected Result</i>	Click the user's name to return to the user's personal page.

<i>Test #</i>	16
<i>Items Under Test</i>	Grocery List Class, Personal Page Class, updateNewsFeed()
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	When personal page is updated, items currently in the grocery list persist
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Call refreshPersonalPage() and check if grocery list remains unchanged 2. Search something from the personal page and check if

	grocery list remains unchanged 3. Add a recipe post to the to the personal page and check if the grocery list remains unchanged 4. Navigate to the main page and navigate back to the personal page and see if the grocery list is unchanged
<i>Expected Result</i>	When running operations on the personal page, nothing is added or removed from the grocery list

<i>Test #</i>	17
<i>Items Under Test</i>	Grocery List Class, addGroceryItem(GroceryItem), updateGroceryFeed()
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	When an item is manually added to the grocery list, the grocery list updates to include the title and description of the item added - GUI is updated as soon as user confirms information in text field
<i>Steps to run test</i>	1. Click "add to grocery list" 2. Add text to resulting text field and click confirm 3. Check to see if the new ingredient is now visible in the Grocery List 4. Check to see if the new ingredient is included in the grocery list items vector
<i>Expected Result</i>	Individual item is added to the grocery list

<i>Test #</i>	18
<i>Items Under Test</i>	Grocery List Class, addGroceryItem(GroceryItem), addRecipeToList(), updateGroceryFeed()
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	When the "add to grocery list" button is clicked from a recipe post on the user's personal page, the ingredients associated with that recipe are automatically added to the grocery list - GUI is updated as soon as user confirms information in text field

<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Choose a post on the personal recipe feed 2. Click “add to grocery list” 3. Check to see if all the ingredients associated with that specific post are added to the grocery list 4. Check to see if all new ingredients are included in the grocery list items vector
<i>Expected Result</i>	Multiple items are added to the grocery list

<i>Test #</i>	19
<i>Items Under Test</i>	Grocery List Class, addGroceryItem(GroceryItem), addRecipeToList(), updateGroceryFeed()
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	If the user attempts to populate the grocery list with a grocery list item that already exists, that item should not be added to the grocery list
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Manually add something to the grocery list that is identical to a grocery list item that already exists in the grocery list 2. Check to see that the item is not added to the grocery list 3. Click “add to grocery list” on a post in the personal feed that has at least 1 ingredient already present in the grocery list 4. Check to see that the specific item is posted on the grocery list
<i>Expected Result</i>	No repetitions occur in the grocery list

<i>Test #</i>	20
<i>Items Under Test</i>	Grocery List Class, addGroceryItem(GroceryItem), addRecipeToList(), updateGroceryFeed()
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	A recipe post that has already been added to the grocery list should have the “add to grocery list” button disabled

<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Click the “add to grocery list” button on a post in the personal page 2. Check to make sure the grocery list is populated accordingly 3. Click the “add to grocery list” button on the same post in the personal page 4. Check to make sure that no changes occur in the grocery list
<i>Expected Result</i>	When the “add to grocery list” button is clicked in on a post that has already been added to the grocery list, the grocery list should not update

<i>Test #</i>	21
<i>Items Under Test</i>	Grocery List Class, removeGroceryItem(GroceryItem), updateGroceryFeed()
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	When the user “removes” an individual grocery list item, that item no longer appears in the GUI and is deleted from the GroceryList vector.
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Make sure the grocery list is populated with at least one grocery list item 2. Choose an ingredient to remove 3. Click “remove” on that grocery list item 4. Check if that item has been removed from the GUI and the vector of grocery list items
<i>Expected Result</i>	Individual item is removed from the grocery list

<i>Test #</i>	22
<i>Items Under Test</i>	Grocery List Class, doneShopping(), updateGroceryFeed()
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	When the user clicks the “Done Shopping” button, all items are removed from GUI and the GroceryList vector has a size of 0.

<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Populate the grocery list with at least one item 2. Click “done shopping” 3. Check to see that all items are removed from the grocery list GUI and the grocery list item vector
<i>Expected Result</i>	All items that were previously in the grocery list are now removed from the grocery list

<i>Test #</i>	23
<i>Items Under Test</i>	Grocery List Class, Personal Page Class
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	When the user views their own personal page, they are able to see their current grocery list but when the user views another user’s personal page, they are not able to see that user’s grocery list
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Navigate to personal page 2. Ensure that grocery list is visible 3. Search for another user on the service 4. Click on the user after search results are populated 5. Ensure that the searched user’s grocery list is not visible when viewing their personal page
<i>Expected Result</i>	See Test Description

<i>Test #</i>	24
<i>Items Under Test</i>	Search Bar Class on main page, Main Class News Page
<i>Test Type</i>	White Box
<i>Test Description</i>	User searches food on the main screen and the results populate in the list of Objects Class
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Search a food on the main page
<i>Expected Result</i>	Check if the list of post objects on the Main Class News Page updates with the appropriate objects

<i>Test #</i>	25
<i>Items Under Test</i>	Search Bar Class on Main Page, Main News Page GUI
<i>Test Type</i>	Black Box
<i>Test Description</i>	User searches food on the main screen and the results are shown in the main news page
<i>Steps to run test</i>	1. Search a food on the main page
<i>Expected Result</i>	Check if the gui updates with the appropriate search results

<i>Test #</i>	26
<i>Items Under Test</i>	Search Bar Class on Personal Page, Personal Page Class
<i>Test Type</i>	White Box
<i>Test Description</i>	User searches food on the personal page and the results are shown in the personal page
<i>Steps to run test</i>	1. Search a food on the personal page
<i>Expected Result</i>	Check if the list of objects in the personal page class update with appropriate results

<i>Test #</i>	27
<i>Items Under Test</i>	Search Bar Class on Personal Page, Personal Page Class
<i>Test Type</i>	Black Box
<i>Test Description</i>	User searches food on the personal page and the results are shown in the personal page
<i>Steps to run test</i>	1. Search a food on the personal page
<i>Expected Result</i>	Check if personal page GUI properly matches the list of post objects recently updated

<i>Test #</i>	28
<i>Items Under Test</i>	Search Bar Class on Personal Page, Personal Page Class
<i>Test Type</i>	White Box
<i>Test Description</i>	User searches person on the page with “@” and the appropriate usernames are filled in the dropdown object during live search
<i>Steps to run test</i>	1. Search a person’s username and check result character by character
<i>Expected Result</i>	All matches are updating in the dropdown’s object list of users while searching

<i>Test #</i>	29
<i>Items Under Test</i>	Search Bar Class on Personal Page, Personal Page Class
<i>Test Type</i>	Blox Box
<i>Test Description</i>	User searches person on the page with “@” and the appropriate usernames are filled in the dropdown GUI during live search
<i>Steps to run test</i>	Search a person’s username and check result character by character
<i>Expected Result</i>	All matches are updating in the GUI while searching

<i>Test #</i>	30
<i>Items Under Test</i>	Posts Class, getters
<i>Test Type</i>	Black Box, Unit Test
<i>Test Description</i>	Timestamp, name, picture, username, instructions, number of reheats, and ingredients are correctly displayed
<i>Steps to run test</i>	1. Create a new post 2. Check that the post is constructed with the proper member variables
<i>Expected Result</i>	The timestamp should be the time that the user submitted the

	recipe and the username should be the current user's username. Everything else should be filled out by the user
--	---

<i>Test #</i>	31
<i>Items Under Test</i>	Posts Class
<i>Test Type</i>	White Box
<i>Test Description</i>	The submit button must check if all forms are completed when creating a new post
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Create a new post and leave certain forms blank 2. Press submit
<i>Expected Result</i>	The form should not go through, and there should be some type of notification or alert

<i>Test #</i>	32
<i>Items Under Test</i>	Posts Class
<i>Test Type</i>	White Box
<i>Test Description</i>	New posts and shared posts are added to the proper recipe list
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Create a new recipe 2. Iterate through the lists and see if the new recipes are added (print it out)
<i>Expected Result</i>	Posts should be added to the lists

<i>Test #</i>	33
<i>Items Under Test</i>	Posts
<i>Test Type</i>	Black Box
<i>Test Description</i>	Posts are formatted properly and all buttons function properly - Featured posts, User post, and Follower posts are all different

<i>Steps to run test</i>	1. Check that all the posts have their proper layouts
<i>Expected Result</i>	Featured posts, user posts, and follower posts should have different layouts

<i>Test #</i>	34
<i>Items Under Test</i>	Posts, Search Map
<i>Test Type</i>	White Box
<i>Test Description</i>	Search Map is properly populated when posts are created (and not when they are reheated)
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Create a new post 2. Check if the post is placed in the search map by iterating through the map or using the contains() function
<i>Expected Result</i>	The contains function should return a 1 for the object placed in the map

<i>Test #</i>	35
<i>Items Under Test</i>	Posts, rehear()
<i>Test Type</i>	White Box
<i>Test Description</i>	Reheating from the newsfeed repopulates the news feed properly
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Reheat a post from the newsfeed 2. Check if post populates the feed for other users
<i>Expected Result</i>	When a post is reheated, the other users' newsfeeds will automatically update

<i>Test #</i>	36
<i>Items Under Test</i>	Posts, rehear()
<i>Test Type</i>	White Box
<i>Test Description</i>	Reheat from other profiles populates the news feed properly

<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Reheat a post from another profile 2. Check if post populates the feed for other users
<i>Expected Result</i>	When a post is reheated, the other users' newsfeeds will automatically update

<i>Test #</i>	37
<i>Items Under Test</i>	Login Database
<i>Test Type</i>	White Box
<i>Test Description</i>	Test login functionality by specifying a username that doesn't exist and a password that doesn't exist. Once the user presses the "login" button, a message box should appear that says "Invalid login". After the message box is exited, the user should still be on the login page.
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Enter a username that doesn't exist in the username text field 2. Enter a password that doesn't exist in the password text field 3. Press login
<i>Expected Result</i>	A message box should appear that says "Invalid login". After exiting the message box, the user is still on the login page.

<i>Test #</i>	38
<i>Items Under Test</i>	Login Database
<i>Test Type</i>	White Box
<i>Test Description</i>	Test login functionality by specifying a username that doesn't exist and a password that does exist. Once the user presses the "login" button, a message box should appear that says "Invalid login". After the message box is exited, the user should still be on the login page.
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Enter a username that doesn't exist in the username text field 2. Enter a password that exists in the password text field

	3. Press login
<i>Expected Result</i>	A message box should appear that says “Invalid login”. After exiting the message box, the user is still on the login page.

<i>Test #</i>	39
<i>Items Under Test</i>	Login Database
<i>Test Type</i>	White Box
<i>Test Description</i>	Test login functionality by specifying a username that does exist and and password that doesn’t match. Once the user presses the “login” button, a message box should appear that says “Invalid login”. After the message box is exited, the user should still be on the login page.
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Enter a username that exists in the username text field 2. Enter a password that doesn’t match in the password text field 3. Press login
<i>Expected Result</i>	A message box should appear that says “Invalid login”. After exiting the message box, the user is still on the login page.

<i>Test #</i>	40
<i>Items Under Test</i>	Login Database
<i>Test Type</i>	White Box
<i>Test Description</i>	Test login functionality by specifying a username that does exist and and password that matches Once the user presses the “login” button, the user should be taken to their main news feed
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Enter a username that doesn’t exist in the username text field 2. Enter a password that exists in the password text field 3. Press login
<i>Expected Result</i>	The user is taken to their main news feed

<i>Test #</i>	41
<i>Items Under Test</i>	Guest Login Functionality
<i>Test Type</i>	Black Box
<i>Test Description</i>	Test guest functionality by pressing “guest” on login page. Guest User should
<i>Steps to run test</i>	1. Press “Guest” button on login page.
<i>Expected Result</i>	The user is taken to their main news feed

<i>Test #</i>	42
<i>Items Under Test</i>	Guest Capability
<i>Test Type</i>	Black Box
<i>Test Description</i>	Guest should lack the capability to follow people, add recipes, create recipes, go to main page, and go to personal page. Basically they should only be able to view the featured news feed.
<i>Steps to run test</i>	1. Log in as guest. 2. Check that expected result is consistent.
<i>Expected Result</i>	Guest can only view featured news feed and cannot follow people, add recipes, create recipes, go to main page, or go to personal page.

<i>Test #</i>	43
<i>Items Under Test</i>	GUI
<i>Test Type</i>	Black Box
<i>Test Description</i>	Test the browser compatibility of HTML/CSS elements and formatting
<i>Steps to run test</i>	1. Open the web page on multiple browsers on different devices

	<ul style="list-style-type: none"> a. Mac, PC, mobile b. Google Chrome, Safari, Internet Explorer, Firefox
<i>Expected Result</i>	Properly using CSS to format would mean that the general format of the web pages is the same across multiple platforms.

<i>Test #</i>	44
<i>Items Under Test</i>	Featured Recipe Class
<i>Test Type</i>	White Box, Black Box
<i>Test Description</i>	Every 24 hours the featured recipe object's list of post objects updates with new recipes
<i>Steps to run test</i>	<ol style="list-style-type: none"> 1. Check if every 24 hours, the appropriate object updates with a fresh new list 2. Make sure all the new recipes are different from the previous ones 3. Make sure GUI updates
<i>Expected Result</i>	The featured recipes update on time and with a fresh new list