Last Update	10/14/2019
Author	Andrew Munoz / Yuteng Zhang / Jorge Garcia / Fei Yao Li / Bohan Zhang
Email	amuno032@ucr.edu / yzhan399@ucr.edu / jgarc271@ucr.edu / fli034@ucr.edu / bzhan014@ucr.edu
Description	Our web app is a web platform for people to share recipes and pictures of what they have cooked. Each user is able to post their products and own recipes with pictures or video instructions and receive comments and ratings from other users who have tried it. The comments allow interaction among our users such as receiving advice and criticism. Each recipe will have classification tags so users can search for specific types of recipes. Additionally, users would be able to save favorite recipes and see featured recipes that are highly rated and relevant to their interests.
Summary / Requirements	This spreadsheet serves two purposes: (1) for your team to track progress during your sprints, and (2) as a record of what user stories were implemented, in which tasks, and by whom. For (1), if you prefer to use an alternate tool to track progress as you go (e.g., Trello), that is fine. However, for (2), it is critical that you keep the "Product Backlog" and "Archives" up to date, at least at the end of each sprint. Also, note that if the number of estimated points for a story ends up being very inaccurate (i.e., if the user story took much more or much less effort than expected), please update the story points estimate and give a comment explaining the adjustment.

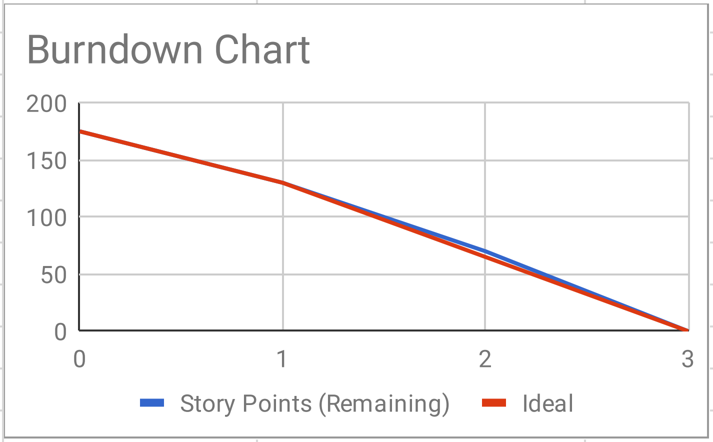
Tabs	**Notes**
"Product Backlog"	Should be initialized with the user stories and their estimated efforts (i.e., story points)
	Depending on the amount of story points, some stories may need to be split into smaller stories
	Add "Acceptance Criteria" to each user story by "insert note" to a spreadsheet cell (right click the cell)
	"Acceptance Criteria" define what it means to be "Done"
"Task Board"	"To Do" is initialized with tasks defined during the Sprint Planning Meeting (i.e., sprint backlog)
	Tasks are expected to have similar sizes, each with a time cost of roughly 3-4 hours
	Add "Details" and "Acceptance Criteria" to a task by "insert note" to a spreadsheet cell (right click the cell)
	Perform task assignments at the Sprint Planning Meeting
	During a sprint, members who become available can pull tasks from the "To Do" list by moving them to "In Progress" and attaching his/her name
	Inform your teammates know before you pull a task from "To Do" list
	"To Verify" should be pulled by a different team member; This is a good time to perform code review and unit testing
	A task in "Done" list should have its verifier's name appended
"Archives"	At the end of a Sprint, finished tasks are moved to "Archives"; Unfinished tasks are moved to the next Sprint Planning Meeting
"Burndown Chart"	At the end of a Sprint, update the "Burndown Chart"
	if "Story Points (Remaining)" is above the "Ideal", it would be an indicator for slow progress

To Do (Sprint Backlog)	In Progress	Testing	Done	
				[1]
[2]				
	[3]			

Sprint ID	Scrum Master	Tasks Completed	Tester
1	Andrew Munoz	Task 01: Retrieve recipe from database (Everyone)	Fei Yao
		Task 02: Display retrieved recipes in feed (Fei Yao)	Bohan
		Task 03: create account (Yuteng)	Andrew
		Task 04: login (Bohan)	Andrew
		Task 05: Format recipe firebase (Jorge)	Bohan
		Task 06: format account database	Yuteng
		Task 07: Connect React App to FireBase (Andrew)	Fei Yao
		Task 08: Set Up Homepage and Routes for the React App (Andrew)	Jorge
		Task 09: Upload Recipes to the DB (Andrew)	Yuteng
		Task 10: Create App NavBar (Andrew)	Jorge
		Task 11: Create recipe page (Jorge)	Fei Yao
2	Fei Yao		
		Task 23: Pull user object from database (Jorge & Andrew) [4]	Fei Yao
		Task 28: Create favorites/created recipe component (Jorge) [5]	Fei Yao
		Task 24: Create list item component (Jorge) [6]	Andrew
		Task 25: Create list component (Jorge) [7]	Fei Yao
		Task 26: Create bio component (Jorge) [8]	Andrew
		Task 27: Create user component (Jorge) [9]	Andrew
		Task 12: Create Frontend for User Profile (Jorge) [10]	Andrew
		Task 13: Add User Object to firebase (Andrew) [11]	Jorge
		Task 14: Sort feed by rating (Fei Yao) [12]	Andrew
		Task 15: Sort by Date (Fei Yao) [13]	Andrew
		Task 16: Create rating percentage in recipe DB (Andrew) [14]	Fei Yao
		Task 17: Thumbs up/down on recipe page (Fei Yao & Andrew) [15]	Jorge
		Task 18: Display rating on recipe/feed item (Fei Yao & Andrew) [16]	Jorge
		Task 19: Modify database on received rating (Fei Yao & Andrew) [17]	Jorge
		Task 22: Fix Recipe Page bug (Recipe Id not Found) - Andrew [18]	Fei Yao
		Task 29: Make Feed component into an abstract component (Andrew) [19]	Fei Yao
		Task 31: Favorite List on profile page (Jorge) [20]	Andrew
		Task 33: Go to recipe from favorite list (Andrew and Jorge)	Jorge
		Task 20: Favorite UI on recipe page (Yuteng) [21]	Bohan
		Task 21: Update user object to db with new favorite (Yuteng) [22]	Bohan

Sprint ID	Scrum Master	Tasks Completed	Tester
		Task 30: Enhance UI on Login & Signup page (Bohan) [23]	Yuteng
		Task 32: Add and remove from favorite list (Yuteng) [24]	Yuteng
3	Jorge	Task 34: Edit profile (fei yao) [25]	Andrew
		Task 35: "Create recipe" page (Jorge) [26]	Andrew
		Task 36: Upload new recipe (Jorge) [27]	Andrew
		Task 37: Search recipe by prefix based search (Andrew) [28]	Jorge & Fei
		Task 38: Search results page (Andrew) [29]	Jorge & Fei
		Task 39: Add search bar to homepage (Andrew) [30]	Jorge & Fei
		Task 45: Social Media Sharing(Yuteng & Bohan) [31]	Andrew
		Task 46: Recipe Page should pull info from Author (Andrew) [32]	Yuteng
		Task 48: User page should be pulling from user table (Andrew & Fei Yao) [33]	Bohan
		Task 49: Author on recipe page should link to Authors profile (Andrew) [34]	Bohan
		Task 50: Retrieve user object from database (Jorge) [35]	Andrew
		Task 51: Check possible errors in input (Jorge) [36]	Andrew & Fei
		Task 52: Push new recipe to "recipe-list" collection in database (Jorge & Andrew) [37]	Andrew & Fei
		Task 53: Push new recipe to user object's "created_recipes" array in database (Jorge & Andrew)	Andrew & Fei
		Task 54: Route user to newly posted recipe upon creation (Jorge) [39]	Andrew & Fei

Sprint ID	Story Points (Total)	Story Points (Completed)	Story Points (Accumulated)	Story Points (Remaining)	Ideal
0	175	0	0	175	175
1	175	45	45	130	130
2	175	60	105	70	65
3	175	70	175	0	0


Burndown Chart

ID	Story Name	Story (small features)	Story Points	Priority	Comments
1	Display homepage	As a user, after logging in, I want to see the latest recipe. [40]	15	-	
2	Display recipe informati	As a new user, I want to view a chosen recipe and the related information. [41]	10	-	
3	Create User Account	As a new user, I want to create my own account for sharing and exploring recipes and phot	10	-	
4	Login	As a user, I want to login to my account so I can work with my own stuff [43]	10	-	
5	Rating System	As a user, I want to see how well recieved a recipe is and be able to add my own input. [44	10	-	
6	Favorites List	As a user, I want to be able to save recipes I like and find them later. [45]	10	-	
7	Nice User Interface	As a user, I want the website to look better than just plain html. [46]	10	9	
8	Social Media Sharing	As a user, I want to share my recipes to my various social media channels. [47]	10	-	
9	Search recipes	As a user, I want to search the recipe database for a specific recipe. [48]	30	-	
10	Create Recipe	As a user, I want to create my own recipes and share them on the website. [49]	30	-	
11	User Profile	As a user, I want to see my user profile and information	20	-	
12	Sort Feed	As a user, I want to sort my feed by rating and data. [50]	15	-	
13	Edit Profile	As a user, I want to modify my bio, picture, and display name on my user profile [51]	15	-	
	Total		**195**		

[1] Design the login page, including the registration link

Acceptance Criteria:
1. XXX
2. XXX

[2] 1. Favorite list shows on profile page
2. Favorite list shows all of user's favorited recipes
3. Favorite list shows none of user's not favorited
4. User is able to remove favorites from list

[3] 1. Entire Recipe Should Be able to fit without scrolling (Unless ingredients or instructions are long)
2. Picture shouldnt take up more than 50% of vh

[4] 1. Should retrieve entire user object with associated favorites/created recipes list.

[5] 1. Should hold list component.
2. Should display appropriate list.

[6] 1. Displays an individual recipe item.

[7] 1. Iterates through all of the list elements.

[8] 1. Displays a bio title.
2. Displays a bio body.

[9] 1. Displays user profile image.
2. Displays user stats.

[10] 1. Profile image.
2. Bio.
3. Tabs for favorited and created recipes.
4. Pull user from database.

[11] 1. User Object has These attributes:
a. Task created List
b. Favorited List
c. Voted List

2. User Object should be created in Login

[12] 1. Highest Rated
2. Lowest Rated

[13] 1. Most recently added
2. Most old

[14] 1. Rating percentage = Upvotes / Total Votes
2. Rating percentage should be an added attribute of the rate object

[15] 1. Thumb up increments both total ratings and likes in DB
2. User must be signed in to rate
3. Non logged in user doesn't see ability to rate

[16] 1.Shows total votes and total liked votes
2. Displays on both recipe and feed pages

[17] Acceptance Criteria:
1. has to be logged in
2. add to user's rated list
3. won't be able to vote twice on a recipe

[18] 1. If a user an invalid recipe id (i.e: /recipe/invalid) the page should redirect to the 404 page

[19] 1. RecipeFeed should use any recipe data passed into it
2. RecipeFeed shouldn't have any firebase calls
3. FeedItems should be using props, not a set variable

[20] 1. Favorite list shows on profile page
2. Favorite list shows all of user's favorited recipes
3. Favorite list shows none of user's not favorited recipes
4. User is able to remove favorites from list

[21] 1. Show button to favorite recipe
2. Button should display on both recipe, feed, and favorite list
3. Should show if recipe already favorited
4. Same functionality regardless of location

[22] Acceptance Criteria:
1. has to be logged in
2. add to user's fav list

[23] 1. create a uniform layout for both pages.
2. use icons from Material UI
3.(optional) use background picture

[24] 1. If favorite is in list, remove it
2. if favorite is not in the list, add to it

[25] Parameters:

bio
picture
user_name

To Pass:
1. Have prompts to edit those 3 parameters.
2. If user leaves prompt blank, keep what they had before
3. Limit bio to 100 characters

[26] 1. Able to input all information for recipe: ingredients, instructions, etc.
2. Can only be accessed when logged in.

[27] 1. Recipe is uploaded successfully to firebase
2. Recipe is assigned an owner in the DB

[28] 1. Match based on exact prefix   query
2. Should be case sensitive

[29] 1. Page use "RecipeFeed" component to display results
2. If no results, don't load feed. Show "Search not found, try again"
3. If multiple results, use scrollbar.
4. Search based on search/:id

[30] 1. Able to search and redirect to search results page from homepage
2. Search bar is in the taskbar

3. Pressing enter with valid text redirects to search page
4. Pressing enter when theres no text in searchbar doesn't link to search page

[31] 1. Share via URL
2. Share via Facebook
3. Share via Twitter

[32] 1. If it was created by a user in the userbase, it should pull their Display name, and Picture

2. If wasn't created by a user, simply show "Source: " and a text of what the source is

[33] Pulls these from DB:
1. Bio
2. name
3. image
4. Date created

[34] If a recipe was created by an author, link to their profile page.

[35] Must check that a valid user is returned from database.

[36] Must warn user when error in input.

[37] Must check that recipe was successfully added.
Must warn user if recipe was not added to database.

[38] Must check that recipe was successfully added.
Must warn user if recipe was not added to database.

[39] When user successfully uploads recipe, they are routed to that new recipe's page.

[40] Acceptance Criteria:
1. Retrieve latest recipes from db
2. Display list of recipes on homepage
3. Unaccessible before login

[41] 1. Be able to select recipe from feed
2. When selected display all information from recipe correctly
3. Inform user when a recipe does not exist

[42] Acceptance Criteria:
1. XXX
2. XXX
3. XXX
4. XXX

[43] Acceptance Criteria:
1. When username and password are correct, able to log in.
2. When either username or password are incorrect, unable to log in.
3. Does not say if username or password is incorrect to protect users' information.

[44] Acceptance Criteria:
1. Display total number of ratings.
2. Display number of liked ratings.
3. User be able to give a 'like' or 'dislike' rating.

[45] Acceptance Criteria:
1. Save recipe from recipe page.
2. Store a user's list of favorite recipes.
3. Display user's favorited recipes on specific page.

[46] Acceptance Criteria:
1. Plan to integrate MaterialUI into webapp.
2. Standardize theme across different features and pages.

[47] Acceptance Criteria:
1. Be able to share with url
2. Be able to share with facebook
3. Be able to share with twitter

[48] Acceptance Criteria:
1. Basic prefix based search
2. Searchbar integrated in navbar
3. Links to a search result page
Reach Goals:
1. Better word based search

[49] Acceptance Criteria:
1. Create recipe with ingredients, instructions, etc.
2. Allow image upload for certain size

[50] Acceptance Criteria:
1. Sort feed by highest rating
2. Sort feed by oldest recipe
3. Sort feed by newest recipe

[51] On profile page:
1. Edit Bio
2. Edit attatched picture
3. Edit display name