Design Use Cases

Version 1.1.0
Last updated: 2016-06-05

Fapp

Food Application

Team

POOP (People Order Our Programs)

Authors

RJ Dioneda, Vinson Gong, Andrew Han, Peter Han, Ketan Kelkar, David Le, Emma Li, Jessica Lin, Daniel Seong, Jonathan Shuai

Table of Contents

Category
Account [A]

Title	**Account Design Use Case #1 “Create A New Account” [A1]**
Description	This use case describes the process by which the user can create an account.
Desired Outcome	The user has an account they can log in with to access the “Fapp” application.
User Goals	The user wants to create an account to connect with friends and share photos of food with others.
Dependent Use Cases	N/A
Requirements	SR01
Priority Level	1
Status	Implemented
Pre-conditions	1. The user has downloaded and installed the application. 2. The user has a valid email.
Post-conditions	1. The user has a personal account. 2. The user has access to the application.
Trigger	The user wants to create an account to gain access to the application.
Workflow	1. login.js shall send data to the controller as a POST. 2. controller shall load signup.js which displays the register form. 3. signup.js shall verify that the form’s strings contain text while not being null. a. Form strings contain text that are null, and returns to signup.js. 4. The register form shall send submitted data to signup.js as parameters. 5. signup.js shall create a user object with given parameters. 6. signup.js shall query the database to verify that another user object does not have the same email or username. a. Duplicate exists and the controller sends the user back to signup.js. 7. signup.js shall make the user object persistent in the database. 8. The controller shall load login.js.

Title	**Account Design Use Case #2 “Log In” [A2]**
Description	This use case describes the process by which the user can log in to their account.
Desired Outcome	The user is logged in to the application with their personal account.
User Goals	The user wants access to their personalized content within the application.
Dependent Use Cases	A1
Requirements	SR02
Priority Level	1
Status	Implemented
Pre-conditions	1. The user has created an account.
Post-conditions	1. The user is logged in. 2. The user is redirected to the “Fapp” (home) screen.
Trigger	The user wants to access the application.
Workflow	1. login.js shall collect strings in username and password text fields. 2. login.js shall verify that username and password strings are not null. a. Username or password strings are null and the controller sends the user back to login.js. 3. login.js shall query the database to verify that username and password match. a. Username does not exist and the controller sends the user back to login.js. b. Username and password do not match and the controller sends the user back to login.js. 4. login.js shall retrieve the matching user object in database. 5. The controller shall load home.js.

Title	**Account Design Use Case #3 “Temporary Password” [A3]**
Description	This use case describes the process by which the user can regain access to their account using a temporary password.
Desired Outcome	The user has a temporary password.
User Goals	The user regains access to their account.
Dependent Use Cases	A2
Requirements	SR03
Priority Level	2
Status	Implemented
Pre-conditions	1. The user has created an account. 2. The user has access to the email account they signed up with.
Post-conditions	1. The user has a temporary 24 hour password for their account.
Trigger	The user cannot remember their password.
Workflow	1. login.js shall send data to the controller as a POST. 2. The controller shall load forgot-password.js which displays the form to enter email. 3. forgot-password.js shall verify that the form’s strings contain text while not being null. a. Form strings contain text that are null; system stays at forgot-password.js. 4. The form shall send submitted data to forgot-password.js as parameters. 5. forgot-password.js shall query the database to verify the email is already linked to an account. a. Email is not linked to an account, forgot-password.js alerts the user, the controller sends the user back to login.js. 6. forgot-password.js shall construct an email object. 7. forgot-password.js shall send an email to the user’s email address 8. forgot-password.js shall navigate to login.js 9. User shall input username and temporary password to log in

Title	**Account Design Use Case #4 “Change Password” [A4]**
Description	This use case describes the process by which the user can change their account password.
Desired Outcome	The user has a new account password.
User Goals	The user has a new account password.
Dependent Use Cases	A2
Requirements	SR04
Priority Level	2
Status	Implemented
Pre-conditions	1. The user is logged in to their account.
Post-conditions	1. The user has a new password for their account.
Trigger	The user wants to update their account password.
Workflow	1. setting.js shall send data to the controller as a POST. 2. The controller shall load change-password.js which displays the form to change password. 3. change-password.js shall verify that the form’s strings contain text while not being null. a. Form strings contain text that are null; system stays at change-password.js. 4. The register form shall send submitted data to change-password.js as parameters. 5. change-password.js shall query the database to verify that password differs from current password. a. Password is the same and the change-password.js alerts user and prompts user for a different password entry. 6. change-password.js chall change the user password in the database. 7. change-password.js shall notify user that password change was a success.

Title	**Account Design Use Case #5 “Change Email” [A5]**
Description	This use case describes the process by which the user can change their account email.
Desired Outcome	The user has a new account email.
User Goals	The user wants to change their email.
Dependent Use Cases	A2
Requirements	SR05
Priority Level	2
Status	Implemented
Pre-conditions	1. The user is logged in to their account.
Post-conditions	1. The user has a new email for their account.
Trigger	The user wants to update their account email.
Workflow	1. settings.js shall send data to the controller as a POST. 2. The controller shall load change-email.js which displays the form to change email. 3. change-email.js shall verify that the form’s strings contain text while not being null. a. Form strings contain text that are null; system stays at change-email.js. 4. The register form shall send submitted data to change-email.js as parameters. 5. change-email.js shall query the database to verify that email differs from current email. a. Email is the same and the change-email.js alerts user and prompts user for a different password entry. 6. change-email.js shall change the user password in the database. 7. Database shall tell change-email.js to notify user that password change was a success.

Title	**Account Design Use Case #6 “Deleting Account” [A6]**
Description	This use case describes the process the user takes to delete their account.
Desired Outcome	The user can delete their account to remove their profile from the application.
User Goals	The user wants to delete their account.
Dependent Use Cases	A2
Requirements	SR06
Priority Level	5
Status	Deferred
Pre-conditions	1. The user has an account. 2. The user is logged in.
Post-conditions	1. The system deletes the user's account.
Trigger	The user stopped using the application and does not want their profile on the application.
Workflow	

Title	**Account Design Use Case #7 “Change Profile Picture” [A7]**
Description	This use case describes the process by which the user can change their profile picture.
Desired Outcome	The user can change their profile picture.
User Goals	The user wants to change their profile picture.
Dependent Use Cases	A2
Requirements	SR07
Priority Level	3
Status	Implemented
Pre-conditions	1. The user has an account. 2. The user is logged in.
Post-conditions	1. The user’s profile picture is changed.
Trigger	The user wants a different profile picture
Workflow	1. settings.js shall send data to the controller as a POST. 2. The controller shall load change-profile-picture.js which displays the button to choose profile picture. 3. change-profile-picture.js shall verify that new profile picture chosen is not being null. a. Form contain new picture data that is null; system stays at change-profile-picture.js. 4. The register form shall send submitted data to change-profile-picture.js as parameters. 5. change-profile-picture.js shall query the database to verify that new profile picture differs from current profile picture. a. Profile picture is the same and the change-profile-picture.js alerts user and prompts user for a different profile picture entry. 6. change-profile-picture.js shall change the user profile picture in the database. 7. Database shall tell change-profile-picture.js to notify user that profile picture change was a success.

Category
Networking [N]

Title	**Networking Design Use Case #1 “Connect With Facebook” [N1]**
Description	This use case describes the option for the user to connect their account with Facebook.
Desired Outcome	The user can access information from their Facebook profile.
User Goals	The user will have access to network with other friends through links to their Facebook.
Dependent Use Cases	A2
Requirements	SR08
Priority Level	2
Status	Deferred
Pre-conditions	1. The user has access to Facebook. 2. The user has logged in.
Post-conditions	1. The user’s friends are imported from Facebook. 2. The user can also post images from their album on Facebook.
Trigger	The user wants to interact with their friends on Facebook through this application.
Workflow	1. tab-bar.js shall send a POST to the controller to open the settings page 2. Controller shall load settings.js, where the user can change their settings 3. Controller shall send the user’s Facebook information to settings.js 4. settings.js shall attempt to connect the user’s profile to the given Facebook information a. If it fails, it goes back to settings.js 5. settings.js shall update the database to store the user’s Facebook information. 6. On success, controller shall load home.js.

Title	**Networking Design Use Case #2 “Sharing To Facebook” [N2]**
Description	This use case describes the process by which the user will share their activity to Facebook.
Desired Outcome	The user can publish a link on Facebook that will direct to this application
User Goals	The user can share activity to different sets of friends on Facebook..
Dependent Use Cases	A2, P1, N1
Requirements	SR09
Priority Level	2
Status	Deferred
Pre-conditions	1. The user is logged in. 2. The user has permission to publish a post on their Facebook.
Post-conditions	1. The user posts a link to the application on Facebook that directs others to the shared photo they have selected.
Trigger	The user wants to share their activity.
Workflow	1. post.js shall send POST with “share_media” flag to controller 2. Controller shall load post.js file which displays “Social Media Post” form 3. post.js shall create post on appropriate social media site 4. post.js shall return post status 5. post.js shall notify user of post status

Title	**Networking Design Use Case #3 “Create Events” [N3]**
Description	This use case describes the process by which the user will invite other people to an event.
Desired Outcome	The user can create events by clicking on a button which will send a notification to the user's friends.
User Goals	The user wants to eat and socialize with other users.
Dependent Use Cases	A2, U1
Requirements	SR10
Priority Level	2
Status	Implemented
Pre-conditions	1. The user is logged in. 2. The user has friends.
Post-conditions	1. The system sends a notification to the user’s friends and sets a reminder for the event for the user and the user’s friends.
Trigger	The user wants to socialize with friends.
Workflow	1. create-event.js shall display the form to create event. 2. create-event.js shall prompt form inputs. 3. create-event.js.js shall verify that the form’s strings contain text and the form’s clock fields contain time values. a. Form strings contain strings that are null or form numbers contain numbers that are null; create-event.js prompts alert to let the user to fill out the form completely. 4. create-event.js shall add event to the database. 5. create-event.js shall add event invitations to database. 6. notification.js shall notify invited friends.

Title	**Networking Design Use Case #4 “Create Groups” [N4]**
Description	This use case describes the process by which the user may group together people with similar interests.
Desired Outcome	The user can access a tab where they can connect with a group of people with similar interests.
User Goals	The user wants to organize a group with friends so they can chat or start organizing for an event.
Dependent Use Cases	U1, U2
Requirements	SR11
Priority Level	5
Status	Deferred
Pre-conditions	1. The user has friends they can add to the group. 2. The user is logged into their account.
Post-conditions	1. All specified members are added into a group where they can communicate and share photos, videos and recipes.
Trigger	The user wants to hangout with others, or receive and give information about certain food types or recipes.
Workflow	

Category
Photos [P]

Title	**Photos Design Use Case #1 “Create A Post” [P1]**
Description	This use case describes the process by which the user can upload photos of food into a post.
Desired Outcome	The user’s photo is posted.
User Goals	The user can share this photo with followers on their feed.
Dependent Use Cases	A2
Requirements	SR12
Priority Level	1
Status	Partially Implemented (Location and Recipe are not fully implemented)
Pre-conditions	1. The user is logged into their account. 2. The user’s phone has a camera or photo library.
Post-conditions	1. The user’s photo is saved to the database as a post and can be viewed by other users.
Trigger	The user wants to share photos of food.
Workflow	1. tab-bar.js shall send a POST to the controller to check whether the camera is activated within the application. a. The POST fails when the camera is disabled or broken and returns to the previous page. 2. controller shall load post-details.js, where the user can add an existing photo from the device. 3. controller shall send chosen photo along with the user’s post information to post-details.js. 4. post-details.js shall attempt to post the photo to the user’s page. a. On failure, the controller goes back to the camera. 5. User shall click screen 6. On success, post-details.js shall display the post and returns to the original page.

Title	**Photos Design Use Case #2 “Apply Filter To Photo” [P2]**
Description	This use case describes the ability for the user to be able to apply a filter to a photo prior to making a post.
Desired Outcome	Photo has filter applied to it.
User Goals	Make the photo more aesthetically appealing without using a 3rd party photo editing application.
Dependent Use Cases	A2, P1
Requirements	SR13
Priority Level	2
Status	Implemented
Pre-conditions	1. The user is logged into their account. 2. The user is on the “Camera” screen.
Post-conditions	1. The user can now post their edited photo.
Trigger	The user wants to edit their photo of food to make it look more appealing to followers.
Workflow	1. camera.js shall send data (original photo) to the controller as a POST. 2. controller shall display the original photo with filter options 3. controller shall pass data from camera.js, and the user’s input of filters applied to . 4. camera.js shall apply filter to the photo and returns the modified photo 5. camera.js shall display the filtered photo to the user.

Title	**Photos Design Use Case #3 “Crop Photos” [P3]**
Description	This use case describes the ability for the user to crop photos.
Desired Outcome	Photo is cropped so only the desired section will be uploaded.
User Goals	Remove unnecessary background which distracts from the food.
Dependent Use Cases	A2, P1
Requirements	SR14
Priority Level	2
Status	Deferred
Pre-conditions	1. The user is logged in. 2. The user has taken a photo to upload.
Post-conditions	1. The photo is ready for further editing.
Trigger	The user wants to post only a portion of the image they have captured.
Workflow	1. camera.js shall send data (original photo) to the controller as a POST. 2. controller shall display the original photo with a crop box 3. camera.js shall accept the user’s input of the crop boundary. 4. camera.js shall crop the photo with given input and returns the modified photo 5. camera.js shall display the cropped photo to the user.

Title	**Photos Design Use Case #4 “Add Text To Photos” [P4]**
Description	This use case describes the ability for the user to add text to photos.
Desired Outcome	User’s post has a description along with their photo.
User Goals	The user wants to personalize the user’s photos with text.
Dependent Use Cases	P1
Requirements	SR15
Priority Level	3
Status	Deferred
Pre-conditions	1. The user has a photo to add text to. 2. The user has text the user wants to add.
Post-conditions	1. The photo has text overlay rendered on it.
Trigger	The user wants to provide more details about the photo directly on the photo.
Workflow	1. post-details.js shall send data to the controller. 2. The controller shall prompt a keyboard and textbox. 3. The user shall type any amount of text. 4. The user shall click the “Done” button. 5. The controller shall send the text to post-details.js as a parameter. 6. post-details.js shall display the text onto the original location that the user clicked. 7. The controller shall load post-details.js.

Title	**Photos Design Use Case #5 “Tag Photos” [P5]**
Description	This use case describes the ability for the user to add tags to their photos.
Desired Outcome	The user can tag their own photos.
User Goals	The photo is tagged.
Dependent Use Cases	P1
Requirements	SR16
Priority Level	1
Status	Implemented
Pre-conditions	1. The user takes a photo they want to tag.
Post-conditions	1. The user’s photo is associated with tags which can be searched.
Trigger	The user wants to categorize their food photos and make it easier to search for.
Workflow	1. post-details.js shall send data to the controller. 2. The controller shall prompt the area on the photo to tag. 3. The controller shall prompt the keyboard and textbox to enter a tag. 4. post-details.js shall verify that the textbox string contains text while not being null. a. Steps 2-4 are repeated for as many tags as the user inputs. 5. The controller shall send chosen photo along with the user’s post information to post-details.js. 6. post-details.js shall post the photo to the user’s page. a. Post fails, the controller goes back to the camera. 7. The controller shall load home.js.

Title	**Photos Design Use Case #6 “Remove Uploaded Pictures” [P6]**
Description	This use case describes the ability for the user to remove a previously published photo.
Desired Outcome	The user does not wish others to see their previously published photo.
User Goals	The user’s photo is removed from the application.
Dependent Use Cases	P1
Requirements	SR17
Priority Level	2
Status	Deferred
Pre-conditions	1. The user has a photo that has already been published. 2. The user wants to remove said photo.
Post-conditions	1. The photo is removed and no longer accessible.
Trigger	The user wants to delete their uploaded pictures.
Workflow	1. home.js shall send data to the controller as a POST. 2. The controller shall load delete-post.js which asks the user to confirm the delete action. 3. The user shall select an option a. User chose confirm delete i. delete-post.js deletes the object in the database. b. User chose cancel i. delete-post.js takes user back to home.js

Title	**Photos Design Use Case #7 “Upload Video” [P7]**
Description	This use case describes the ability for the user to take videos of their food-making process.
Desired Outcome	The user’s video is recorded onto the application without having to exit the application to use the camera.
User Goals	The user wants to demonstrate to others how to create the food following the recipe.
Dependent Use Cases	A2
Requirements	SR18
Priority Level	4
Status	Deferred
Pre-conditions	1. The user is logged in. 2. The user’s phone has a camera.
Post-conditions	1. The user’s video is saved to the database.
Trigger	The user wants to provide a detailed tutorial for the recipe.
Workflow	1. tab-bar.js shall send a POST to the controller to check whether the camera is activated within the application. a. The POST fails when the camera is disabled or broken and returns to the previous page. 2. controller shall load post-details.js, where the user can add an existing video from the device. 3. controller shall send chosen video along with the user’s post information to post-details.js. 4. post-details.js shall attempt to post the video to the user’s page. a. On failure, the controller goes back to the camera. 5. On success, post-details.js shall display the post and then returns to the original page.

Category
User Interaction [U]

Title	**User Interaction Design Use Case #1 “Add Friend” [U1]**
Description	This use case describes the process by which the user adds another user as a friend.
Desired Outcome	The user becomes friends with another user.
User Goals	The user wants to share photos, communicate, and organize events with another they know.
Dependent Use Cases	A2
Requirements	SR19
Priority Level	1
Status	Implemented
Pre-conditions	1. The user’s friend has an account. 2. The user is logged into their account.
Post-conditions	1. The user and friend user are listed in each others’ “Friends” list.
Trigger	The user sees that their friend is also using the application and would like to share and receive posts from them.
Workflow	1. profile.js shall send current profile (friend to be added) as a POST to controller 2. Controller shall load profile.js 3. profile.js shall add the current profile to profile’s list of friends 4. profile.js shall update follow/friends list in the database 5. profile.js shall notify user of friend addition

Title	**User Interaction Design Use Case #2 “Remove Friend” [U2]**
Description	This use case describes the process by which a user removes a friend.
Desired Outcome	The user is no longer friends with another user.
User Goals	The user wishes to maintain privacy of their photos from their friend.
Dependent Use Cases	A2, U1
Requirements	SR20
Priority Level	1
Status	Implemented
Pre-conditions	1. The user is logged into their account. 2. The user is already friends with the user they wish to remove.
Post-conditions	1. The user is no longer friends with the other user.
Trigger	The user no longer wants to be associated with the other user.
Workflow	1. profile.js shall send current profile (friend to be removed) as a POST to controller 2. profile.js shall remove current profile from my_profile’s list of friends 3. profile.js shall update follow/friends list in the database 4. profile.js shall notify user of friend removal

Title	**User Interaction Design Use Case #3 “Block User” [U3]**
Description	This use case describes the process by which a user blocks another user.
Desired Outcome	The user’s content can no longer be seen by the blocked user.
User Goals	The user wishes to maintain complete privacy and eliminate interaction with the blocked user.
Dependent Use Cases	A2
Requirements	SR21
Priority Level	3
Status	Deferred
Pre-conditions	1. The user shall be logged into their account.
Post-conditions	2. The user’s content is no longer available from the blocked user’s end.
Trigger	The user and the blocked user have personal strife with each other.
Workflow	1. profile.js shall send current profile (friend to be removed) as a POST to controller 2. profile.js shall remove current profile from list of friends 3. profile.js shall update follow/friends list in the database 4. profile.js shall notify user of friend removal

Title	**User Interaction Design Use Case #4 “View Friends’ Food” [U4]**
Description	This use case describes the process of how a user views posts made by their friends.
Desired Outcome	The user can view pictures of foods uploaded by their friends.
User Goals	The user is up to date on the activities of their friends.
Dependent Use Cases	A2, U1
Requirements	SR22
Priority Level	1
Status	Implemented
Pre-conditions	1. The user is logged into their account. 2. The user is friends with the user.
Post-conditions	1. The user sees posts made by their friends.
Trigger	User wants to view friends’ food.
Workflow	1. home.js shall access the database to find updates to friends’ posts. 2. home.js shall return all changed posts to home.js 3. home.js shall display the newly updated friend’s feed.

Title	**User Interaction Design Use Case #5 “User Page Visibility” [U5]**
Description	This use case describes a system that allows the user to choose who can view their profile.
Desired Outcome	The user can choose who their profile is visible to.
User Goals	The user wishes to have control over their profile privacy.
Dependent Use Cases	U3, U4
Requirements	SR23
Priority Level	2
Status	Deferred
Pre-conditions	1. The user is logged into their account.
Post-conditions	1. Users that are not friends are are blocked from viewing the user's profile.
Trigger	The user does not want everyone to see their profile.
Workflow	1. profile.js shall send POST to controller with a flag to change visibility 2. controller shall display a page with a list of friends for the user to select specific visibilities for 3. controller shall send the selected friend(s) to profile.js 4. Profile.js shall disable selected friend(s) from accessing the user’s page 5. profile.js shall display a confirmation message saying the specified friend(s) are now unable to view the user’s page

Title	**User Interaction Design Use Case #6 “Save Favorite Pictures” [U6]**
Description	This use case describes the ability for users to save a list of their favorite pictures.
Desired Outcome	The user can save posts that they like into a maintained personal list.
User Goals	The user would like to save certain pictures into a personal list that they can view anytime.
Dependent Use Cases	A2, P1
Requirements	SR24
Priority Level	2
Status	Deferred
Pre-conditions	1. The user is logged into their account.
Post-conditions	1. The selected photo is saved to the user’s favorite photos list.
Trigger	The user wants to save pictures that they like to view at a later time.
Workflow	1. post.js shall send POST to controller with a flag to save the current photo 2. profile.js shall receive the photo and creates a save photo action. 3. The action shall update the database to add the photo to the user’s favorite photos list. 4. The current view shall notify user that the photo has successfully been saved.

Category
Post Details [D]

Title	**Post Details Design Use Case #1 “Rate A Post” [D1]**
Description	This use case describes the process by which a user can rate their friends’ posts.
Desired Outcome	The user’s rating is displayed.
User Goals	The user wants to give feedback to their friends’ post.
Dependent Use Cases	A2, U1
Requirements	SR25
Priority Level	1
Status	Implemented
Pre-conditions	1. The user shall be logged into their account.
Post-conditions	1. The user’s rating will be displayed under their friends’ photo.
Trigger	The user wants to give feedback to their friends.
Workflow	1. post.js shall send data (the post ID) to the controller as a POST. 2. The controller shall retrieve requested threads within the database. 3. post.js shall add user rating to list of posts under specified post. 4. post.js shall update the display with the new rating on the post

Title	**Post Details Design Use Case #2 “Comment On A Post” [D2]**
Description	This use case describes the process by which a user may add comments on posts.
Desired Outcome	The user’s comment is displayed on other user’s posts.
User Goals	The user wants to give feedback to other user’s content.
Dependent Use Cases	A2, U1, P1
Requirements	SR26
Priority Level	1
Status	Implemented
Pre-conditions	2. The user shall be logged into their account.
Post-conditions	2. The user’s comment will be displayed under a post.
Trigger	The user wants to give feedback to other users.
Workflow	1. post.js shall send data (the post ID) to the controller as a POST. 2. The controller shall receive requested threads within the database. 3. post.js shall add user comment to list of posts under specified post. 4. post.js shall update the display with the new comment on the post

Title	**Post Details Design Use Case #3 “View More Details” [D3]**
Description	This use case describes the process by which the user can view comments, ratings, and other post details.
Desired Outcome	The user’s comments and ratings are displayed, as well as a description with tags and an option to view a larger photo of the post.
User Goals	The user wants to view more details about their post, give comments or feedback, or learn from a recipe.
Dependent Use Cases	A2, P1, D1
Requirements	SR27
Priority Level	1
Status	Implemented
Pre-conditions	1. The user is logged in. 2. The friend has a post with a photo. 3. The friend has a post with a recipe video and description. 4. There are comments and ratings on the friend’s photos.
Post-conditions	1. The user sees the photo with description.
Trigger	The user sees a photo that they is interested in and would like to know more
Workflow	1. post.js shall send data (view more clicked, and the original post ID) to the controller as a POST. 2. The controller shall receive requested threads within the database. 3. post.js shall return the list of threads and constructs an ordered list of posts with the threads in a Hashmap object, 4. Controller shall send hashmap object to post.js. 5. post.jsl shall display the detailed post to the user.

Title	**Post Details Design Use Case #4 “View Others’ Recipes” [D4]**
Description	This use case describes the process of how a user views recipes of other approved users (friends)
Desired Outcome	The user can view other user’s recipes
User Goals	The user has instructions to make something without having to ask
Dependent Use Cases	A2, D3
Requirements	SR28
Priority Level	2
Status	Deferred
Pre-conditions	1. The user has an account. 2. The user is logged in. 3. Other users have uploaded recipes.
Post-conditions	1. Recipe page/pane displayed.
Trigger	The user wants to view others’ recipes.
Workflow	1. post.js shal send data (view recipe clicked, and the original post ID) to the controller as a POST. 2. The post.js shall retrieve requested recipe within the database. 3. post.js shall displaythe detailed recipe to the user.

Title	**Post Details Design Use Case #5 “Tag Friends in Photo” [D5]**
Description	This use case describes the way the user can tag friends in a photo.
Desired Outcome	Photos have tags specifying which users were present or relevant to the photo.
User Goals	The user can notify friends who the user ate with so that those friends can easily see the picture and other users can see who ate the food.
Dependent Use Cases	U1, P1
Requirements	SR29
Priority Level	2
Status	Deferred
Pre-conditions	1. The user is logged in. 2. The user posted a photo. 3. The user has friends.
Post-conditions	1. The photo has the specified friends tagged onto the post. 2. The tagged friends receive a notification of the tag action. 3. Other users can see which friends are tagged on the post.
Trigger	The user wants to share a photo with his followers to specify who the user ate with.
Workflow	1. post.js shall send data to controller as a POST 2. Controller shall display“Tag Friends” form 3. Valid form is sent as parameter to tag-friends.js 4. tag-friends.js shall update posts“tagged-friends” list in database 5. post.js shall notify user of success

Category
Browsing (B)

Title	**Browsing Design Use Case #1 “Search By Tag” [B1]**
Description	This use case describes the process by which the user can search for photos by tag.
Desired Outcome	The user can filter photos to match a certain type of food.
User Goals	The user is able to explore foods that match certain type of taste palate or dietary restriction.
Dependent Use Cases	P5
Requirements	SR30
Priority Level	1
Status	Implemented
Pre-conditions	1. The photos have tags associated with them.
Post-conditions	1. Photos matching the given tags are displayed.
Trigger	The user wants to explore a certain type of food.
Workflow	1. search-bar.js shall send submitted data (searchByTags flag and user’s specified tags) to the controller. 2. controller shall implement onChangeText with data from search-bar.js. 3. onChangeText shall retrieve requested Posts within the database. 4. onChangeText shall return the list of posts in order of highest relevance to user’s specified tags 5. home.js shall display the returned posts to the user.

Title	**Browsing Design Use Case #2 “Search by Rating” [B2]**
Description	This use case describes a search function that allows users to search for food based on the food’s ratings.
Desired Outcome	The user can filter food posts from a certain area by order of rating.
User Goals	The user would like to organize food posts by their ratings.
Dependent Use Cases	P5
Requirements	SR31
Priority Level	1
Status	Deferred
Pre-conditions	1. The user is logged in. 2. The food posts within the area have ratings or otherwise considered low priority.
Post-conditions	1. Photos are sorted by ratings in decreasing order.
Trigger	The user wants to see food posts listed in priority of their ratings.
Workflow	1. search-bar.js shall send submitted data (searchByRating flag and user’s specified rating [1-5]) to the controller. 2. controller shall implement onChangeText with data from search-bar.js. 3. onChangeText shall retrieve requested Posts within the database. 4. onChangeText shall return the list of posts in order of highest rating 5. home.js shall display the returned posts to the user.

Title	**Browsing Design Use Case #3 “Search by Popularity” [B3]**
Description	This use case describes the process by which the user can search trending posts within a certain timeframe.
Desired Outcome	The user is able to view posts by popularity.
User Goals	The user is able to stay on top of the latest trends in food culture.
Dependent Use Cases	P5
Requirements	SR32
Priority Level	2
Status	Deferred
Pre-conditions	1. The user is logged in. 2. The timeframe is selected
Post-conditions	1. Photos of trending food within the specified timeframe are displayed.
Trigger	The user wants to know trending food.
Workflow	1. search-bar.js shall send submitted data (searchByPopularity flag and user’s specified timeframe (day, week, month, etc.)) to the controller. 2. controller shall shall retrieve requested Posts within the database. 3. onChangeText shall return the list of posts in order of highest popularity. 4. home.js shall display the returned posts to the user.

Title	**Browsing Design Use Case #4 “Search by Location” [B4]**
Description	This use case describes the process by which the user can search posts from a specific location.
Desired Outcome	Posts from the location that the user specified are displayed.
User Goals	The user wants to see food/ places to eat in his or her neighborhood or a certain region.
Dependent Use Cases	P5
Requirements	SR33
Priority Level	2
Status	Deferred
Pre-conditions	1. The user is logged in.
Post-conditions	1. Posts from the specified location will be selected from the server and are displayed on the user’s wall.
Trigger	The user wants to explore food/places to eat.
Workflow	1. search-bar.js shall send submitted data (searchByLocation flag and user input of location) to the controller. 2. The controller shall implement onChangeText with data from search-bar.js. 3. onChangeText shall retrieve requested POST within the database. 4. onChangeText shall return the list of posts and constructs an ordered list of posts with the threads in a Hashmap object, then sends the object to home.js. 5. home.js shall display the returned posts to the user.

Title	**Browsing Design Use Case #5 “Search by Upload Time” [B5]**
Description	The user shall be able to search photos by their upload times.
Desired Outcome	Photos are sorted with the most recent upload first.
User Goals	The user wants to see food uploaded near the time they are searching.
Dependent Use Cases	P5
Requirements	SR34
Priority Level	2
Status	Deferred
Pre-conditions	1. The user has logged in.
Post-conditions	1. The system displays posts most recently uploaded to the user’s news feed.
Trigger	The user wants to see food people eat around the time they are searching.
Workflow	1. home.js shall send data to the controller as a POST. 2. The controller shall load search-bar.js which displays a textbox and keyboard. 3. The user shall click the “Most Recent” button. 4. search-bar.js shall send data to the controller. 5. The user shall type in any text. 6. The user shall click the “Search” button. 7. The controller shall pass data including the text as a string parameter to onChangeText. a. The string is null and the controller sends the user back to search-bar.js. 8. onChangeText shall query the database to find all posts that match the search criteria. 9. The controller shall load search-bar.js with all the objects found by onChangeText.

Title	**Browsing Design Use Case #6 “Advanced Browsing” [B6]**
Description	This use case describes the way the advanced search works.
Desired Outcome	Combined search options to filter posts with greater precision.
User Goals	The user can search with precision for photos they want to view
Dependent Use Cases	P5
Requirements	SR35
Priority Level	2
Status	Deferred
Pre-conditions	1. The user is logged into their account. 2. The user has friends or a recipe book they would like to perform search through
Post-conditions	The user can combine search features B1-B5 into one search and get photos that match the query.
Trigger	The user wants to look for photos
Workflow	1. search-bar.js shall send data to the controller as a POST. 2. The controller shall load adv-search-bar.js. 3. adv-search-bar.js shall send submitted data to the controller. 4. The controller shall implement advOnChangeText with data from adv-search-bar.js. 5. advOnChangeText shall retrieve requested post objects from the database. 6. advOnChangeText shall return the list of posts and constructs an ordered list of posts with the threads in a Hashmap object. 7. home.js shall display the returned posts to the user.

Title	**Browsing Design Use Design Case #7 “Refresh Home Screen” [B7]**
Description	The user shall be able to get new posts that have been made while they are browsing the home screen.
Desired Outcome	The user shall see new posts that have been made after refreshing.
User Goals	The user can view new uploads by other users.
Dependent Use Cases	P1
Requirements	SR36
Priority Level	2
Status	Implemented
Pre-conditions	1. The user has logged into their account.
Post-conditions	1. The system displays the latest posts in the home screen.
Trigger	The user wants to view the latest posts that have been made.
Workflow	1. home.js shall notice that a refresh action has been requested in a POST. 2. home.js shall access the database to find newer posts. 3. queryDataAll shall return new posts to home.js 4. home.js shall display the newest posts.