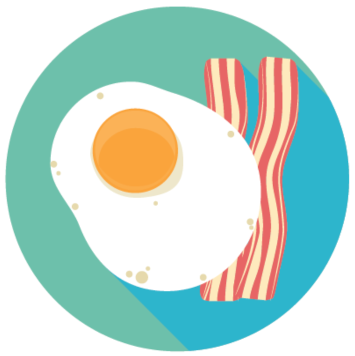
Design Use Cases

Version 1.0.1

Last updated: 2016-05-04



Team

People Order Our Programs

P.O.O.P.

Authors

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

Category
Account (A)

Title	**Account Design Use Case #1 “Create an Account” [A1]**
Description	This use case describes the process by which the user can create an ac
Desired Outcome	The user creates a personal account with their specified username and password.
User Goals	The user can access the application.
Dependent Use Cases	N/A
Requirements	SR01
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user has downloaded the application.
Post-conditions	1. The user has a personal account. 2. The user has access to the application.
Trigger	From login.xml, the user clicks the “Register” button.
Workflow	1. login.xml sends data to the controller as a POST. 2. The controller loads signup.xml which displays the register form. 3. SignUp.java verifies that the form’s strings contain text while not being null. a. Form strings contain text that are null, and returns to signup.xml. 4. The register form sends submitted data to SignUp.java as parameters. 5. SignUp.java creates a user object with given parameters. 6. SignUp.java queries the database to verify that another user obje does not have the same email or username. a. Duplicate exists and the controller sends the user back to signup.xml. 7. SignUp.java makes the user object persistent in the database. 8. The controller loads login.xml.

Title	Account Design Use Case #2 “Log in” [A2]
Description	This use case describes the process by which a user can log in to their account.
Desired Outcome	The user can log in with their personal account.
User Goals	The user wants access to their personal content within the application.
Dependent Use Cases	A1
Requirements	SR02
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user has created an account.
Post-conditions	1. The user is logged in. 2. The user is redirected to the main screen.
Trigger	From login.xml, the user click the “Login” button.
Workflow	1. login.xml sends strings in username and password text fields to Login.java. 2. Login.java verifies that username and password strings are not n a. Username or password strings are null and the controller sends the user back to login.xml. 3. Login.java queries the database to verify that username and password match. a. Username does not exist and the controller sends the use back to login.xml. b. Username and password do not match and the controller sends the user back to login.xml. 4. Login.java retrieves the matching user object in database. 5. The controller loads home.xml.

Title	Account Design Use Case #3 “Reset Password” [A3]
Description	This use case describes the process by which a user can reset their acc password.
Desired Outcome	The user has a new account password.
User Goals	The user has access to their account.
Dependent Use Cases	A1
Requirements	SR03
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user has created an account.
Post-conditions	1. The user has a new password for their account.
Trigger	From login.xml, the user clicks the “Forgot Password?” button.
Workflow	1. login.xml sends data to the controller as a POST. 2. The controller loads forgot_pass.xml which displays the form to e email. 3. ForgotPass.java verifies that the form’s strings contain text while being null. a. Form strings contain text that are null; system stays at ForgotPass.java. 4. The form sends submitted data to ForgotPass.java as parameter 5. ForgotPass.java queries the database to verify the email is alrea linked to an account. a. Email is not linked to an account, ForgotPass.java alerts t user, the controller sends the user back to login.xml. 6. ForgotPass.java calls EmailManager.java, passing the email addr as a parameter. 7. EmailManager.java constructs an email object. 8. EmailManager.java sends an email to the user’s email address.

Title	Account Design Use Case #4 “Change Password” [A4]
Description	This use case describes the process by which a user can change their account password.
Desired Outcome	The user has a new account password.
User Goals	The user has updated their account security.
Dependent Use Cases	A1
Requirements	SR04
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user has created an account.
Post-conditions	1. The user has a new password for their account.
Trigger	1. From settings.xml, the user clicks “Change Password”
Workflow	1. settings.xml sends data to the controller as a POST. 2. The controller loads change_pass.xml which displays the form to change password. 3. ChangePass.java verifies that the form’s strings contain text while being null. a. Form strings contain text that are null; system stays at change_pass.xml. 4. The register form sends submitted data to ChangePass.java as parameters. 5. ChangePass.java queries the database to verify that password d from current password. a. Password is the same and the ChangePass.java alerts us and prompts user for a different password entry. 6. ChangePass.java changes the user password in the database. 7. ChangePass.java tells change_pass.xml to notify user that passw 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 a user to connect their account wi their other accounts on Facebook.
Desired Outcome	The user can be directly connected/interact with their Facebook.
User Goals	The user will have easier access to network with other friends through lin their Facebook.
Dependent Use Cases	A1
Requirements	SR09
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user has access to Facebook. 2. The user has logged in.
Post-conditions	1. The user can message friends on Facebook. 2. The user can also post images from their album on Facebook.
Trigger	From settings.xml, the user clicks the “Connect to Facebook” button.
Workflow	1. nav_bar sends a POST to the controller to open the settings pag 2. Controller loads settings.xml, where the user can change their settings 3. Controller sends the user’s Facebook information to Settings.java 4. Settings.java attempts to connect the user’s profile to the given Facebook information a. If it fails, it goes back to settings.xml 5. Settings.java updates the database to store the user’s Facebook information. 6. On success, settings.xml will return to the home page.

Title	**Networking Design Use Case #2 “Sharing to other social media” [N**
Description	This use case describes the process by which a user will share their acti other social media.
Desired Outcome	The user can publish a link on their desired social media that will direct t application
User Goals	The user can share activity to different sets of friends on different social media.
Dependent Use Cases	N1, P1
Requirements	SR10
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user is logged in. 2. The user has permission to publish a post on their other social m
Post-conditions	1. The user posts a link to the application on other social media tha directs others to the shared photo they have selected.
Trigger	From add_post.xml, user clicks “share to social media” option.
Workflow	1. Add_post.xml sends POST with “share_media” flag to controller 2. Controller loads post_external.xml file which displays “Social Me Post” form 3. PostExternal.java loads form 4. PostExternal.java creates post on appropriate social media site 5. PostExternal.java returns post status to external_site.xml 6. external_site.xml notifies user of post status

Title	Networking Design Use Case #3 “Create Events” [N3]
Description	This use case describes the process by which a user will invite other peo an event or party.
Desired Outcome	The user can create events by clicking on a button which will send notific to users’ friends.
User Goals	The user wants to eat or have parties with friends.
Dependent Use Cases	A1, A2, U1
Requirements	SR11
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user is logged in. 2. The user has friends.
Post-conditions	1. The system send notifications to the user’s friends and set a rem for the event for user and user’s friends.
Trigger	From events.xml, the user clicks “Create New Event”.
Workflow	1. events.xml displays the form to create event. 2. events.xml sends form inputs to Events.java in a POST. 3. Events.java verifies that the form’s strings contain text and the fo numeric fields contain numbers. a. Form strings contain strings that are null or form numbers contain numbers that are null; Events.java notifies events to prompt the user to fill out the form completely. 4. Events.java adds event to the database. 5. Events.java returns event invitations to notification.xml. 6. Notification.xml notifies invited friends.

Category
Photos (P)

Title	**Photos Design Use Case #1 “Upload photos of food” [P1]**
Description	This use case describes the process by which a user may upload photos food.
Desired Outcome	The user’s photo is saved to their ‘recipe book’.
User Goals	The user can share this photo with followers on their feed or recipe book
Dependent Use Cases	A2
Requirements	SR13
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user is logged into their account.
Post-conditions	1. The user’s photo is saved to the server in their “recipe book”.
Trigger	From any screen that contains the nav_bar, the user clicks the “camera”
Workflow	1. nav_bar sends a POST to the controller to check whether the car is activated within the application. a. The POST fails when the camera is disabled or broken ar returns to the previous page. 2. controller loads add_post.xml, where the user can add an existin photo from the device. 3. controller sends chosen photo along with the user’s post informa to AddPost.java. 4. AddPost.java attempts to post the photo to the user’s page. a. On failure, the controller goes back to the camera. 5. On success, add_post.xml will display the post and then return to original page after the user clicks the screen.

Title	Photos Design Use Case #2 “Take Photos With Camera From App” [P2]
Description	This use case describes the process by which a user may take pictures t upload them directly through their camera phone
Desired Outcome	The user’s photo is saved to their ‘recipe book’ without having to exit the to use the camera.
User Goals	The user can conveniently share this photo with their followers when the view the photo in their feed or when looking at the user’s recipe book wit having to take the picture.
Dependent Use Cases	A2
Requirements	SR14
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user is logged into their account. 2. The user’s phone has a camera
Post-conditions	1. The user takes a photo that is saved to the server in their “recipe book”.
Trigger	From any screen that contains the nav_bar, the user clicks the “camera”
Workflow	1. nav_bar sends a POST to the controller to check whether the ca is activated within the application. a. The POST fails when the camera is disabled or broken a returns to the previous page. 2. controller loads add_post.xml, where the user can take a new ph 3. controller sends chosen photo along with the user’s post informa to AddPost.java. 4. AddPost.java attempts to post the photo to the user’s page. a. On failure, the controller goes back to the camera. 5. On success, add_post.xml will display the post and then return to original page after the user clicks the screen.

Title	Photos Design Use Case #3 “Apply filter to photo” [P3]
Description	This use case describes the implementation for being able to apply a filte photo
Desired Outcome	Photo has filter applied to it
User Goals	Make the photo more aesthetically appealing without using a 3rd party p editing application and leaving the current app
Dependent Use Cases	A2, P1, P2
Requirements	SR15
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user is logged into their accounts. 2. The user has uploaded a photo or take a picture with the in-app camera feature.
Post-conditions	1. The user can now save their edited photo to the server in their ‘r book’.
Trigger	From the edit_post.xml, the user clicks “Filter”.
Workflow	1. edit_post.xml sends data (original photo) to the controller as a P 2. controller displays the original photo with filter options then passe data from edit_post, and the user’s input of filters applied to Filter.java. 3. Filter.java applies filter to the photo and returns the modified pho edit_post.xml. 4. edit_post.xml displays the filtered photo to the user.

Title	Photos Design Use Case #4 “Crop Photos” [P4]
Description	This use case describes the implementation for cropping photos.
Desired Outcome	Photo is cropped so only a partial piece can be uploaded.
User Goals	Remove unnecessary background which distracts from food
Dependent Use Cases	A2, P2
Requirements	SR16
Priority Level	2
Status	Incomplete
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	From edit_post.xml, the user clicks “Crop”.
Workflow	1. edit_post.xml sends data (original photo) to the controller as a P 2. controller displays the original photo with a crop box, then passes data from edit_post, and the user’s input of the crop boundary to Crop.java. 3. Crop.java crops the photo with given input and returns the modifi photo to edit_post.xml. 4. edit_post.xml displays the cropped photo to the user.

Title	**Photos Design Use Case #5 “Add Text” [P5]**
Description	This use case describes the ability for the user to add text to photos.
Desired Outcome	Photo has text on it.
User Goals	The user wants to personalize the user’s photos with text.
Dependent Use Cases	P1
Requirements	SR17
Priority Level	1
Status	Incomplete
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 on it.
Trigger	From add_post.xml, the user clicks on any location on the photo.
Workflow	1. add_post.xml sends data to the controller. 2. The controller prompts a keyboard and textbox. 3. The user types any amount of text. 4. The user clicks the “Done” button. 5. The controller sends the text to add_post.xml as a parameter. 6. add_post.xml displays the text onto the original location that the clicked. 7. The controller loads add_post.xml.

Title	**Photos Design Use Case #6 “Tag photos” [P6]**
Description	This use case describes the ability for the user to add tags to their photo
Desired Outcome	The user can tag their own photos.
User Goals	The photo is tagged.
Dependent Use Cases	P1, P2
Requirements	SR18
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user takes a photo they wants to tag.
Post-conditions	1. The user’s photo has the given tags.
Trigger	From add_post.xml, the user clicks “add tags”.
Workflow	1. add_post.xml sends data to the controller. 2. The controller prompts the area on the photo to tag. 3. The controller prompts the keyboard and textbox to enter a tag. 4. AddPost.java verifies that the textbox string contains text while n being null. a. Steps 2-4 are repeated for as many tags as the user wan 5. The controller sends chosen photo along with the user’s post information to AddPost.java. 6. AddPost.java posts the photo to the user’s page. a. Post fails, the controller goes back to the camera. 7. The controller loads home.xml.

Title	Photos Design Use Case #8 “Remove uploaded pictures” [P8]
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	SR20
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user has a photo that’s already been published. 2. The user wants to remove said photo.
Post-conditions	1. The photo is removed and no longer accessible.
Trigger	1. User clicks “Delete Post” on one of the user’s own posts from home.xml
Workflow	1. home.xml sends data to the controller as a POST. 2. The controller loads delete_post.xml which asks the user to confi the delete action. 3. The user chooses whether to confirm delete or cancel. a. User chooses delete; delete_post.xml notifies DeletePost.java with a POST. i. DeletePost.java deletes the object in the database b. User chooses cancel; DeletePost.java takes user back to home.xml

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 a friend
Desired Outcome	The user becomes ‘friends’ with a person.
User Goals	The user wants to share photos, communicate, or organize groups and events with someone they know.
Dependent Use Cases	A2
Requirements	SR22
Priority Level	1
Status	Incomplete
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 are listed in each other’s ‘friend’ list.
Trigger	From profile.xml, user selects “Add Friend”
Workflow	1. profile.xml sends current profile (friend to be added) as a POST t controller 2. Controller loads my_profile.xml 3. my_profile.xml loads AddFriend.java which adds the current profi my_profile’s list of friends 4. AddFriend.java updates follow/friends list in the database 5. profile.xml notifies 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 a friend.
User Goals	The user wishes to maintain privacy of photos from a friend.
Dependent Use Cases	A2, U1
Requirements	SR23
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user is logged into their account. 2. The user is ‘friends’ with the user they wish to remove.
Post-conditions	1. The user is no longer ‘friends’ with the user they wish to remove.
Trigger	From profile.xml, user selects “remove friend”
Workflow	1. profile.xml sends current profile (friend to be removed) as a POS controller 2. Controller loads my_profile.xml 3. my_profile.xml loads RemoveFriend.java which removes current profile from my_profile’s list of friends 4. RemoveFriend.java updates follow/friends list in the database 5. profile.xml notifies user of friend removal

Title	User Interaction Design Use Case #5 “View friends’ food” [U5]
Description	This use case describes the process of how a user views uploads made other approved users (friends)
Desired Outcome	User can view pictures of foods uploaded by a pre-approved set of other users (friends)
User Goals	User be updated on friends’ dietary activities
Dependent Use Cases	A2, U1
Requirements	SR26
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user is logged into their account.
Post-conditions	1. The user sees uploads made by friends
Trigger	The user accesses home.xml from nav_bar.
Workflow	1. Home.java accesses the database to find updates to friend’s’ pos 2. Home.java returns all changes posts to home.xml 3. home.xml displays the newly updated friend’s newsfeed.

Title	**User Interaction Design Use Case #7 “User page visibility” [U7]**
Description	This use case describes a system that allows the user to choose who ca view their homepage.
Desired Outcome	The user can choose who their homepage is visible to.
User Goals	The user wishes to have control over their homepage privacy.
Dependent Use Cases	U3, U4
Requirements	SR28
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user is on their profile page.
Post-conditions	1. Certain people are blocked from viewing the user's homepage.
Trigger	From profile.xml, the user clicks on “Change homepage visibility”
Workflow	1. profile.xml sends POST to controller with a flag to change visibilit 2. controller displays a page with a list of friends for the user to sele specific visibilities for 3. controller sends the selected friend(s) to ChangeVisibility.java 4. ChangeVisibility.java disables selected friend(s) from accessing t user’s page 5. profile.xml displays a confirmation message saying the specified friend(s) are now unable to view the user’s page

Title	User Interaction Design Use Case #8 “Save Favorite Pictures” [U8]
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 view anytime.
Dependent Use Cases	A2, P1
Requirements	SR29
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user is logged in.
Post-conditions	1. The photo is saved to the user’s favorite photos.
Trigger	From any page displaying photos, the user clicks the “favorite” button on specific photo.
Workflow	1. profile.xml sends POST to controller with a flag to save the curre photo 2. controller sends the photo to SavePhoto.java 3. SavePhoto.java receives the photo and creates a save photo act 4. The action updates the database to add the photo to the user’s favorite photos list. 5. The current view (whatever xml the photo was displayed in) displ a confirmation message saying that the photo has successfully b saved.

Category
Post Details (D)

Title	**Post Details Design Use Case #1 “Rate and Comment” [D1]pet**
Description	This use case describes the process by which a user may add comment rating under his or her friends’ photo.
Desired Outcome	The user’s comment or rating is displayed.
User Goals	The user wants to give feedback to his or her friends’ photo.
Dependent Use Cases	A1, A2, U1
Requirements	SR30
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user shall be logged into their account.
Post-conditions	1. The user’s comment or rating will be displayed under his or her friends’ photo.
Trigger	From profile.xml, the user clicks “Comment”.
Workflow	1. profile.xml sends data (the post ID) to the controller as a POST. 2. The controller calls AddComment.java to retrieve requested threa within the database. 3. AddComment.java adds user comment to list of posts under spec post. 4. profile.xml updates the display with the new comment on the pos

Title	**Post Details Design Use Case #2 “View More Details” [D2]**
Description	This use case describes the process by which the user can access comn ratings or other details on one of their friend’s photos.
Desired Outcome	The friend’s comments and ratings are displayed, as well as a descriptio tags and an option for a larger photo. There is also an option to view the recipe.
User Goals	The user wants to view more details about their friend’s photo, give comr or feedback, or learn from a recipe.
Dependent Use Cases	A2, P9, D1, D3, D4, D6, D7
Requirements	SR31
Priority Level	1
Status	Incomplete
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. 2. The user sees how the food was made.
Trigger	From profile.xml, the user clicks “View More”.
Workflow	1. profile.xml sends data (view more clicked, and the original post I the controller as a POST. 2. The controller calls ViewMore.java to retrieve requested threads within the database. 3. ViewMore.java returns the list of threads and constructs an order list of posts with the threads in a Hashmap object, then sends the object to profile.xml. 4. profile.xml displays the detailed post to the user.

Title	Post Details Design Use Case #3 “Add Tags to Photo” [D3]
Description	This use case describes the process by which a user can add tags to the photos.
Desired Outcome	The user’s photo is now associated with tags they chose to tag the photo with.
User Goals	The user wishes to categorize their photos.
Dependent Use Cases	A2, P1
Requirements	SR32
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user is logged into their account. 2. The user has uploaded a photo they want to add tags to.
Post-conditions	1. The user’s photo has tags associated with their photo. 2. The user’s photo shows up when others search the tag.
Trigger	From edit_post.xml, the user clicks the “Add Tags” button.
Workflow	1. edit_post.xml sends data to the controller. 2. The controller prompts a keyboard and text box. 3. The user types in any text. 4. The user clicks the “Done” button. 5. The controller sends this text as a string parameter to EditPost.ja a. The string is null and the controller sends the user back t edit_post.xml. 6. EditPost.java adds the new tag to the post object. 7. The controller loads edit_post.xml.

Title	**Post Details Design Use Case #4 “Add Description to Photo” [D4]**
Description	This use case describes the process by which a user may add a descript their photo
Desired Outcome	The user’s photo has a description added to it
User Goals	The user wishes to include a description of food
Dependent Use Cases	A2
Requirements	SR33
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user is logged into their account. 2. The user has uploaded a photo to add a description to.
Post-conditions	1. User’s photo has a description that can be viewed alongside the photo.
Trigger	From edit_post.xml, the user clicks “Description”.
Workflow	1. edit_post.xml sends data to the controller. 2. The controller prompts a keyboard and textbox. 3. The controller sends the text as a string parameter to EditPost.ja 4. EditPost.java adds the description to the post object. 5. The controller loads home.xml.

Title	Post Details Design Use Case #5 “Add a Recipe” [D5]
Description	This use case describes the process of how the user uses the applicatior recipe book and links a recipe with a food tag and photo.
Desired Outcome	The user has a recipe uploaded to his recipe book that links to a food tag their photo.
User Goals	The user wishes to share a recipe of a food to others.
Dependent Use Cases	A2, P1, P6
Requirements	SR34
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user has logged into their account. 2. The user has uploaded a photo to link a recipe with. 3. The user has added tags to the photo.
Post-conditions	1. The user’s photo has a recipe linked to it and the tags associated the photo link to the recipe as well.
Trigger	1. User clicks “Add Recipe” from edit_post.xml or add_post.xml 2. edit_post.xml or add_post.xml sends data to the controller as a POST, loading add_recipe.xml
Workflow	1. add_recipe.xml displays the form to add a recipe. 2. add_recipe.xml sends form inputs to AddRecipe.java in a POST. 3. AddRecipe.java verifies that the form’s strings contain text a. Form strings contain strings that are null; AddRecipe.java notifies add_recipe.xml to prompt the user to fill out the fo completely. 4. AddRecipe.java adds recipe to the post’s “recipe” field in the database.

Title	Post Details Design Use Case #6 “View recipe” [D6]
Description	This use case describes the process of how a user views recipes of the posted food
Desired Outcome	The user can view other users’ recipes.
User Goals	The user has instructions to make food without having to ask for the reci
Dependent Use Cases	U2, D4
Requirements	SR35
Priority Level	2
Status	Incomplete
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	From profile.xml, the user clicks “View Recipe” on a post.
Workflow	1. profile.xml sends data (view recipe clicked, and the original post to the controller as a POST. 2. The controller calls ViewRecipe.java to retrieve requested recipe within the database. 3. ViewRecipe.java returns the recipe and sends it to profile.xml 4. profile.xml displays the detailed recipe to the user.

Title	Post Details Design Use Case #7 “Tag other users in post”[D7]
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 ca easily see the picture and other users can see who ate the food.
Dependent Use Cases	U1, P1
Requirements	SR36
Priority Level	2
Status	Incomplete
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	From add_post.xml, the user selects “share a photo”.
Workflow	1. add_post.xml sends data to controller as a POST 2. Controller displays “Tag Friends” form 3. Valid form is sent as parameter to tagFriends.java 4. tagFriends.java updates photo’s “tagged-friends” list in database 5. add_post.xml notifies 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 ph 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 dietary restriction.
Dependent Use Cases	P6
Requirements	SR37
Priority Level	1
Status	Incomplete
Pre-conditions	1. The photos have tags associated with them.
Post-conditions	1. Photos matching the given tags are displayed.
Trigger	From search.xml, the user checks “search by tag”.
Workflow	1. search.xml sends submitted data (searchByTags flag and user’s specified tags) to the controller. 2. controller implements SearchTags.java with data from search.xm 3. SearchTags.java retrieves requested Posts within the database. 4. SearchTags.java returns the list of posts in order of highest relava to user’s specified tags 5. home.xml displays 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 fo 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	A2, D1
Requirements	SR38
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user is logged in. 2. The food posts within the area have ratings or otherwise conside low priority.
Post-conditions	1. Photos are sorted by ratings in decreasing order.
Trigger	From search.xml, the user checks “search by rating”.
Workflow	1. search.xml sends submitted data (searchByRating flag and user’ specified rating [1-5]) to the controller. 2. controller implements Rating.java with data from search.xml. 3. Rating.java retrieves requested Posts within the database. 4. Rating.java returns the list of posts in order of highest rating 5. home.xml displays the returned posts to the user.

Title	**Browsing Design Use Case #3 “Search by popularity within a timeframe” [B3]**
Description	This use case describes the process by which the user can search trend 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	A1, A2, B7
Requirements	SR39
Priority Level	1
Status	Incomplete
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 display
Trigger	From search.xml, the user checks “search by popularity”.
Workflow	1. search.xml sends submitted data (searchByPopularity flag and us specified timeframe (day, week, month, etc.)) to the controller. 2. controller implements Popularity.java with data from search.xml. 3. Popularity.java retrieves requested Posts within the database. 4. Popularity.java returns the list of posts in order of highest popular 5. home.xml displays 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 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	A1, A2
Requirements	SR40
Priority Level	1
Status	Incomplete
Pre-conditions	1. The user is logged in.
Post-conditions	1. Posts from the specified location will be selected from the server are displayed on the user’s wall.
Trigger	From search.xml, the user checks “search by location”.
Workflow	1. search.xml sends submitted data (searchByLocation flag and use input of location) to the controller. 2. The controller implements Location.java with data from search.xr 3. Location.java retrieves requested Posts within the database. 4. Location.java returns the list of posts and constructs an ordered posts with the threads in a Hashmap object, then sends the objec home.xml. 5. home.xml displays the returned posts to the user.

Title	Browsing Design Use Case #5 "Search by Budget" [B5]
Description	This use case describes the process by which the user can search for fo a budget range.
Desired Outcome	Posts containing food within the user's specified budget are displayed.
User Goals	The user wants to find food that fits their budget.
Dependent Use Cases	A2, D4
Requirements	SR41
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user shall be logged in 2. Other posts shall be made with their price displayed
Post-conditions	1. The user can see all photos that contain food within their budget
Trigger	From search.xml, the user checks "search by budget".
Workflow	1. search.xml sends submitted data (searchByBudget flag and user input of price bounds) to the controller. 2. The controller implements Budget.java with data from search.xm 3. Budget.java retrieves requested Posts within the database. 4. Budget.java returns the list of posts and constructs an ordered lis posts with the threads in a Hashmap object, then sends the objec home.xml. 5. home.xml displays the returned posts to the user.

Title	**Browsing Design Use Case #6 “Search by Upload Time” [B6]**
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	A1, A2
Requirements	SR42
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user has logged in.
Post-conditions	1. The system displays posts most recently uploaded to the user’s feed.
Trigger	From home.xml, the user clicks the “Search” button.
Workflow	1. home.xml sends data to the controller as a POST. 2. The controller loads search.xml which displays a textbox and keyboard. 3. The user clicks the “Most Recent” button. 4. search.xml sends data to the controller. 5. The user types in any text. 6. The user clicks the “Search” button. 7. The controller passes data including the text as a string paramet Search.java. a. The string is null and the controller sends the user back t search.xml. 8. Search.java queries the database to find all posts that match the search criteria. 9. The controller loads search.xml with all the objects found by Search.java.

Title	Browsing Design Use Case #7 "Advanced Browsing" [B7]
Description	This use case describes the way the advanced search works.
Desired Outcome	Combined search options to enhance the precision search function
User Goals	The user can search with precision for photos they want to view
Dependent Use Cases	A2, P1, B1, B2, B3, B4, B5, B6
Requirements	SR43
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user is logged in. 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-B6 into one search and get ph that match the query.
Trigger	From search.xml, the user clicks the "Advanced Search" button.
Workflow	1. search.xml sends data to the controller as a POST. 2. The controller loads adv_search.xml. 3. adv_search.xml sends submitted data to the controller. 4. The controller implements AdvSearch.java with data from adv_search.xml. 5. AdvSearch.java retrieves requested post objects from the databa 6. AdvSearch.java returns returns the list of posts and constructs ar ordered list of posts with the threads in a Hashmap object. 6. home.xml displays the returned posts to the user.

Title	Browsing Design Use Case #9 “Refresh News Feed” [B9]
Description	The user shall be able to fetch the new posts that have been made while are browsing their news feed.
Desired Outcome	The user shall see new posts made when refreshing.
User Goals	The user shall view new uploads.
Dependent Use Cases	A2, U1
Requirements	SR45
Priority Level	2
Status	Incomplete
Pre-conditions	1. The user has logged in.
Post-conditions	1. The system displays the latest news feed results.
Trigger	1. The user swipes down from home.xml
Workflow	1. home.xml tells Home.java that a refresh action has been request a POST. 2. Home.java accesses the database to find newer posts. 3. Home.java returns new posts to home.xml 4. home.xml displays the newest posts.