

Title:	Creating a Bucket
Description:	This use case describes the steps to create a bucket
System Under Design:	Bucket Manager System
Primary Actor:	User
Participants:	
Goal:	Create a bucket successfully
Following Use Cases:	-Change Privacy Setting on a Bucket -Sorting Items in a Bucket by Priority
Invariant:	
Precondition:	This case assumes the user is already logged into the application.
Success Postcondition:	Bucket is created.

Actor: 1. User selects the menu bar. 3. User selects the My Buckets button. 5. User selects the “+ New Bucket”. 7. User enters the desired name into the entry field and selects the “Save” button.	System Responses: 2. Application displays the “My Buckets” button. 4. Application displays the “+ New Bucket” button. 6. Application displays “Enter Bucket Name:” with an entry field to the right of this dialog and a “Save” button below.
--	---

Title:	Creating an Item
Description:	This use case describes the steps to create an item
System Under Design:	Item Manager System
Primary Actor:	User
Participants:	
Goal:	Create a task successfully
Following Use Cases:	-Adding Images -Adding Deadlines
Invariant:	
Precondition:	This case assumes the user is logged in and is within the “My Buckets” section.
Success Postcondition:	Item is created within a bucket

Actor: 1. User selects the desired bucket that they wish to add a task to. 3. User selects the “+ New Item” button . 5. User enters the desired name into the entry field and selects the “Save” button.	System Responses: 2. Application displays the “+ New Item” button. 4. Application displays “Enter Item Name:” with an entry field to the right of this dialog and a “Save” button below.
--	---

Title:	Deleting an Item
Description:	This use case describes the steps to delete an item.
System Under Design:	Item Manager System
Primary Actor:	User
Participants:	
Goal:	Delete an item within a bucket successfully
Following Use Cases:	
Invariant:	
Precondition:	This case assumes the user is within a bucket already.
Success Postcondition:	Item is deleted from the bucket

Actor: 1. User selects the item that they wish to delete. 3. User selects the “Delete Item” button. 5. User selects the “Confirm” button.	System Responses: 2. Application displays the “Delete Item” button that is red at the bottom of the screen. 4. Application displays the message “Are you sure that you want to delete this item?” with a “Cancel” button below the message and a “Confirm” button below the “Cancel” button
---	--

Title:	Adding deadlines.
Description:	This use case describes the steps to add deadlines to items.
System Under Design:	Item Manager System
Primary Actor:	User
Participants:	
Goal:	To add a deadline to an item successfully
Following Use Cases:	
Invariant:	
Precondition:	This case assumes the user is within a bucket already.
Success Postcondition:	Deadline is added to the item

Actor: 1. User selects the item that they wish to add a deadline to. 3. User selects the “Add Deadline” button. 5. User enters the deadline in any format that they desire (can be M/D/Y, a day of the week, a time, etc) and then selects the “Confirm” button.	System Responses: 2. Application displays the “Add Deadline” button that is shown on the screen. 4. Application displays the message “Enter the Deadline:” with a “Cancel” button below the message and a “Confirm” button below the “Cancel” button
--	---

Title:	Attaching Photos to items
Description:	This use case will describe the steps to add a photo to an item.
System Under Design:	Item Manager System
Primary Actor:	User
Participants:	
Goal:	Attach a photo to an item successfully.
Following Use Cases:	
Invariant:	
Precondition:	This case assumes the user is within a bucket already.
Success Postcondition:	Photo is attached to the item

Actor: 1. User selects the item that they wish to add a photo to. 3. User selects the “Add Image” button. 5. User selects the “File Upload” button. 7. User selects the image that they want to upload. 9. User selects the “Confirm” button.	System Responses: 2. Application displays the “Add Image” button that is shown on the screen. 4. Application displays the message “Upload an Image” with a button that says “File Upload” below it. 6. Application brings the user to their file directory on their computer. 8. Application displays a “Confirm” button and a “Cancel” button.
---	--

Title:	Sharing a Bucket
Description:	This use case describes sharing a bucket with a friend.
System Under Design:	Bucket Manager System
Primary Actor:	User
Participants:	Receiving User(Friend)
Goal:	Share a bucket with a friend so they can contribute to the bucket.
Following Use Cases:	
Invariant:	
Precondition:	Both users must already be friends in the app.
Success Postcondition:	Both users can edit a bucket.

Actor: 1.User selects the bucket they wish to shair. 3. The user taps the share button. 5. The user taps a plus sign next to the icon of the friend they want to share the bucket with. 8. The user can add more friends.	System Responses: 2.The app shows all the contents and options of the selected bucket. 4. The app shows a list of all the user's friends. 6. The plus that was tapped is replaced with an x. 7. The friend that was add gets a notification that they were added
--	---

Title:	Creating an Account
Description:	This use case describes the steps of creating a Dream Catcher account
System Under Design:	User Authentication System
Primary Actor:	User
Participants:	
Goal:	Create a Dream Catcher Account
Following Use Cases:	
Invariant:	
Precondition:	App is in login page
Success Postcondition:	User successfully creates an account

Actor: 1. User taps “Create Account” button 3. User enters username and password	System Responses: 2. System changes view to account creation page 4. System checks if username is registered 5. System registers user to database 6. System displays confirmation message to user
---	--

Title:	Logging In
Description:	This use case describes the steps to logging into a Dream Catcher account
System Under Design:	User Authentication System
Primary Actor:	User
Participants:	
Goal:	Log into Dream Catcher account
Following Use Cases:	Creating an Account
Invariant:	
Precondition:	App is on login page and user has previously created an account
Success Postcondition:	User successfully logs in

Actor: 1. User enters username 3. User enters password 4. User taps login button	System Responses: 2. System checks if username exists 5. System checks if username and password match
--	--

Title:	Logging Out
Description:	This use case describes the steps to log out of your Dream Catcher account
System Under Design:	Account management system
Primary Actor:	User
Participants:	
Goal:	Log out of Dream Catcher account
Following Use Cases:	
Invariant:	
Precondition:	User must already have an account
Success Postcondition:	User successfully logs out

Actor: 1. User taps on the hamburger menu icon 3. User taps on logout button	System Responses: 2. System displays menu bar 4. System logs user out and returns to login page
---	--

Title:	Setting Profile Picture
Description:	This use case describes the steps to set or change your profile picture
System Under Design:	Account Management System
Primary Actor:	User
Participants:	
Goal:	To set or update user profile picture
Following Use Cases:	
Invariant:	
Precondition:	User is logged in
Success Postcondition:	User successfully changes profile picture

Actor: 1. User taps on hamburger menu icon 3. User taps “My Account” button 5. User taps “Change Profile Picture” button 7. User grants access to camera roll 9. User selects a photo from camera roll and taps “Update Profile Picture” button	System Responses: 2. System displays menu bar 4. System changes view to account management page 6. System requests access to user’s camera roll 8. System displays photos from user’s camera roll 10. System updates user’s profile picture with their selected photo
---	---

Title:	Checking the time to complete item
Description:	This use case describes the steps to check how long it took to complete a task.
System Under Design:	Statistic System
Primary Actor:	User
Participants:	
Goal:	See the time from item creation to completion.
Following Use Cases:	
Invariant:	
Precondition:	The item must exist and be completed to see the time it took. This case assumes the user is within a bucket already.
Success Postcondition:	Time to complete an item is shown

Actor: 1. The user clicks an item in the bucket. 3. The user can see the time it took to complete the item.	System Responses: 2. The system shows the information for that item.
--	--

Title:	Change Privacy Setting on a Bucket
Description:	This use case describes the steps required to change the privacy settings on a bucket.
System Under Design:	Bucket Manager System
Primary Actor:	User
Participants:	
Goal:	Change the privacy setting on a bucket
Following Use Cases:	
Invariant:	
Precondition:	This case assumes the user is within a bucket already.
Success Postcondition:	The privacy setting is applied to the bucket.

Actor: 1. The user clicks on the bucket settings. 3. The user clicks the privacy button. 5. The user clicks the option they want to apply.	System Responses: 2. The system shows all the settings for the bucket. 4. The system shows a radio menu with the different privacy options.
--	--

Title:	Display Completed Items in a Bucket
Description:	This use case describes the steps a user will take to see the items in a bucket that are completed.
System Under Design:	Bucket Manager System
Primary Actor:	User
Participants:	
Goal:	To display the items in a bucket that have been completed
Following Use Cases:	
Invariant:	
Precondition:	This case assumes the user is within a bucket already.
Success Postcondition:	Completed items are shown in a bucket

Actor: 1. The user taps the “Completed” tab near the top of the bucket page.	System Responses: 2. The system shows a list of completed items.
--	--

Title:	Sorting Items in Bucket by Priority
Description:	This use case describes the steps a user will take to sort the items in a bucket by priority (High-Low or Low-High)
System Under Design:	Bucket Manager System
Primary Actor:	User
Participants:	
Goal:	To display the items in a bucket by their priority level.
Following Use Cases:	
Invariant:	
Precondition:	This case assumes the user is within a bucket already.
Success Postcondition:	

Actor: 1. The user tabs the priority button on the bucket page to sort from High-Low. 3. The user tabs the priority button on the bucket page to sort from Low-High.	System Responses: 2. The system sorts the items by priority level High-Low. 4. The system sorts the items by priority level Low-High.
---	--

USE CASE MODELS

