DePaul University

PART A (Team)

PART B and C (Individual)

Final Project Documentation

Beautifly Application

Amy Aumpansub

CSC 471: Mobile Application Development for iOS
Professor Xiaoping Jia

March 16, 2020

Table of Contents

PART A
Final Project Description and Documentation

Application Name: Beautifly

Overview: "Beautifly" is developed by Xcode 11 and Swift 5. It offers the users a fun and convenient way to retrieve detailed data of make-up products from various brands. This application is literally like a make-up dictionary which users can easily search for any make-up products by keywords, look at product details, save their favorite items to their personal list, and check the top-reviewed products.

App Features

This app contains 5 main features shown on a tab bar of every screen as follows: MyHome, MySearch, MyAccount, MyList, and MyTop5.

1. MyHome
   - Users can touch the 'Beautifly' letter picture and it will change to the wing photo. That used by multi-touch API feature.
   - Users can tab the items, ' My Home, My Search, User, Top5 rating, My List' under the main page and it will navigate directly to those pages.

2. MySearch
   - Users can search for a make-up product by type and brand. The app will show the matched item found on API in a found-item list
   - From the found-item list, users can select any products on the list to find more details about the selected product. After clicking at the product, the detail view will be shown.
   - The detail view will provide user with the selected product in details including image, description, price, and website.
   - From the detail view, users can also add the item to their favorite list, which will update and add item on the MyList tab

3. MyAccount
   - Users can search their information about account, preference and policy.
   - Users click the account, it shows the tableview that have email, password, phone-number, birthday and address.
   - Users click the email tableview, it shows details and users can change their information by clicking the 'Edit' button. If users leave the textfield with nothing, it will show the alert and users update the new information, it will show the actionsheet.
   - Preference tableview has same function as account tableview.
   - User can click the policy tableview, it shows the detail about privacy policy.

4. MyList
   - Users can check their favorite items in this view. The list on MyList is updated when users select and add the specific items on detail view which can be navigated through MySearch and MyTop5 tabs

5. MyTop5
   - The top 5 rating will provide the most popular of the product from API including image, detail, price, and product type by click on the list and the detail screen will be displayed. User can click the tab bar of top 5, it shows the top 5 items. User can click to see the details.

Screens Explanation

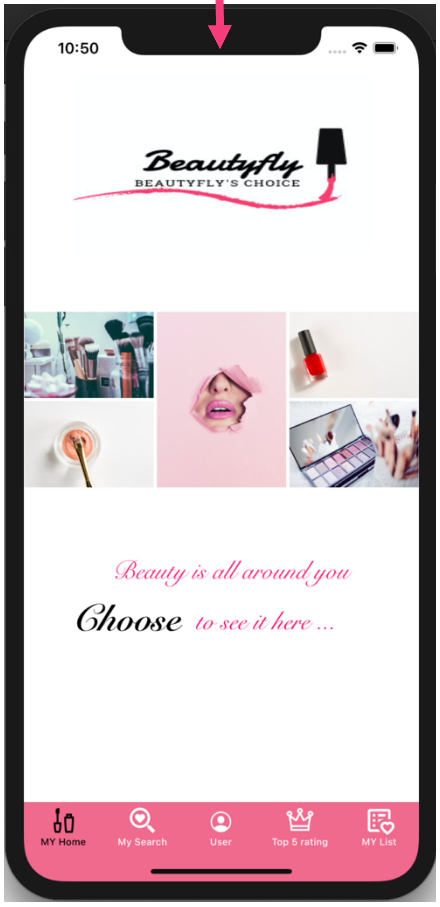
I. My Home Tab

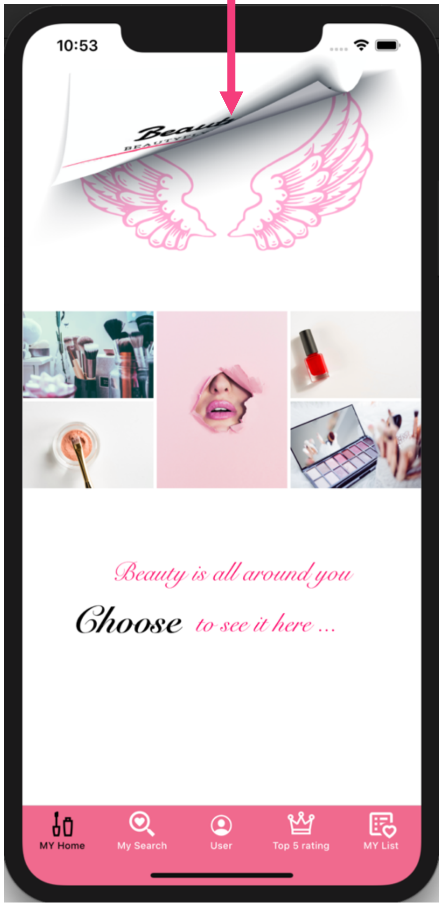
1.1 Home Screen

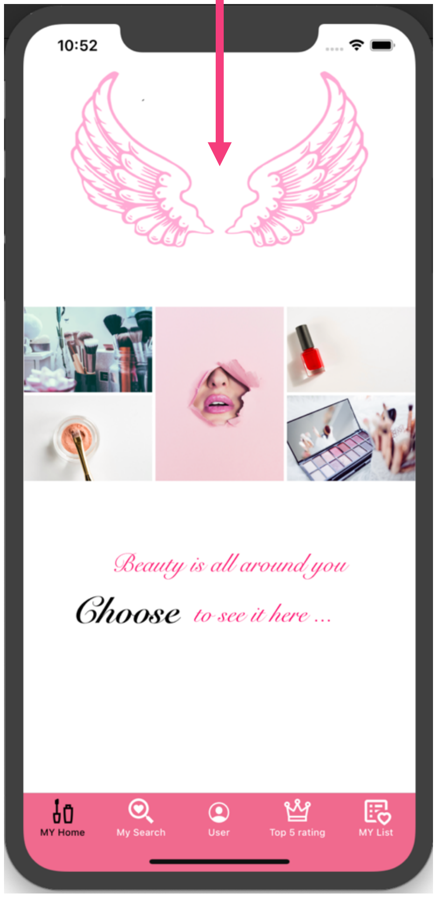
This is a main page of this app. Users tab the 'MY Home', it always shows this page.

If users touch this photo, it will change the wing photo.

Users can check the changed photo, now it is a wing photo.







II. My Search Tab

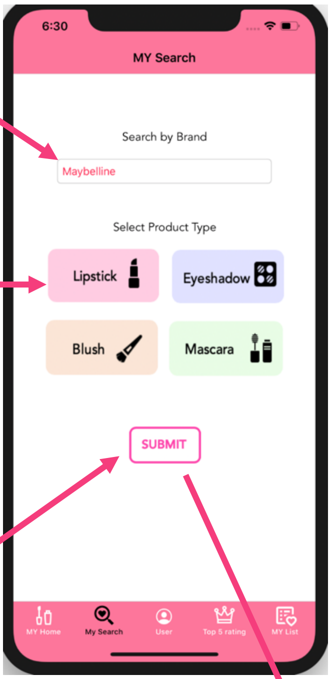
2.1 My Search Screen

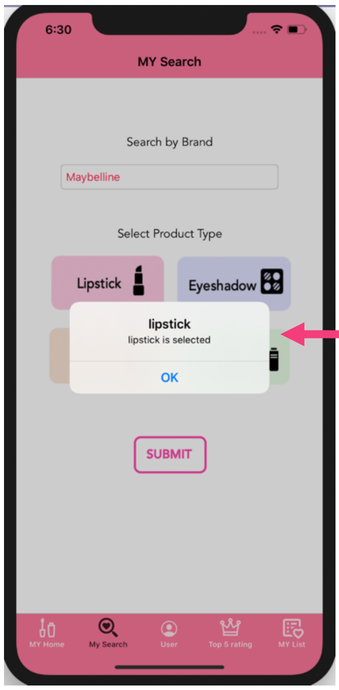
Note: The make-up API provides products' data from only 50 make-up brands, so users may not be able to search some brands on app. We will further discuss about this issue in Part B: Limitation.

Search Products by typing brand in the input field

Select Product Type by clicking at any of 4 buttons

Click at submit button will trigger the app to search item on make-up API and show the all matched items on the Found List screen in 2.2 below

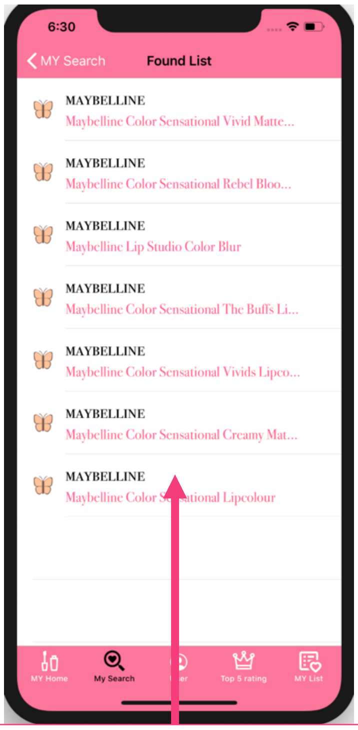




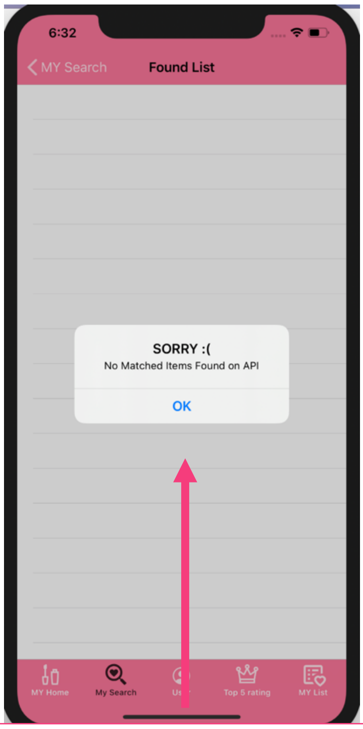
lipstick
lipstick is selected
OK

After clicking at any Product Type buttons, an alert box will appear to confirm your selected type

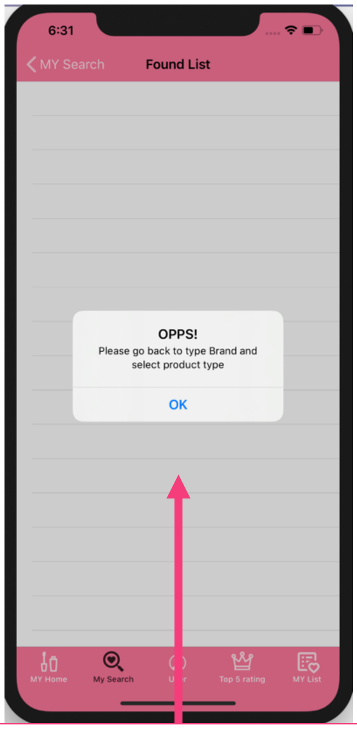
2.2 Found List Screen



MAYBELLINE
Maybelline Color Sensational Vivid Matte...
MAYBELLINE
Maybelline Color Sensational Rebel Bloo...
MAYBELLINE
Maybelline Lip Studio Color Blur
MAYBELLINE
Maybelline Color Sensational The Buffs Li...
MAYBELLINE
Maybelline Color Sensational Vivids Lipco...
MAYBELLINE
Maybelline Color Sensational Creamy Mat...
MAYBELLINE
Maybelline Color Sensational Lipcolour



SORRY :(
No Matched Items Found on API
OK



OPPS!
Please go back to type Brand and select product type
OK

Show all matched items found on API on Found List screen

Alert Box will appear if there is no matched items on API

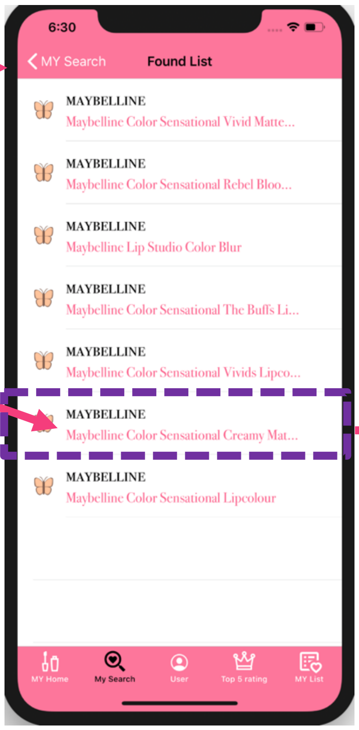
Alert Box will appear if user doesn't type brand & select product type

2.2 Found List Screen

2.3 Product Details Screen

Show all matched items found on API on Found List

Select any items by clicking at the cell on the list. It will trigger the program to load product details from API and show them on product details screen in 2.3



Found List

MAYBELLINE
Maybelline Color Sensational Vivid Matte...

MAYBELLINE
Maybelline Color Sensational Rebel Bloo...

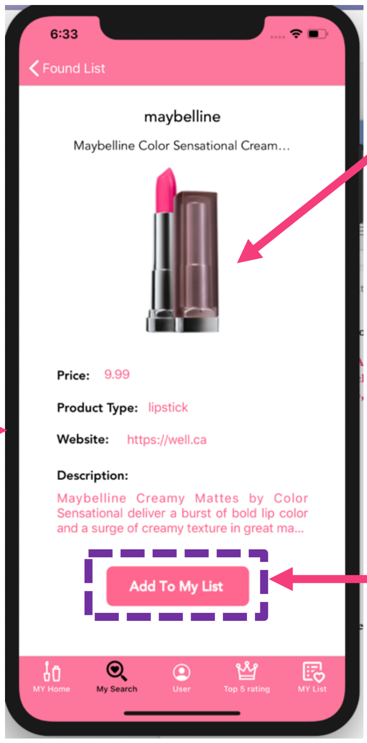
MAYBELLINE
Maybelline Lip Studio Color Blur

MAYBELLINE
Maybelline Color Sensational The Buffs Li...

MAYBELLINE
Maybelline Color Sensational Vivids Lipco...

MAYBELLINE
Maybelline Color Sensational Creamy Mat...

MAYBELLINE
Maybelline Color Sensational Lipcolour



maybelline

Maybelline Color Sensational Cream...

Price: 9.99

Product Type: lipstick

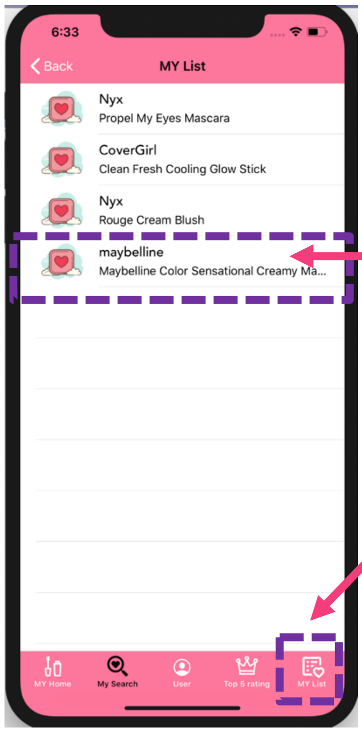
Website: https://well.ca

Description:
Maybelline Creamy Mattes by Color Sensational deliver a burst of bold lip color and a surge of creamy texture in great ma...

The details of selected item show on this screen including name, image, brand, price, type, website, description.

User can click "Add to My List" Button to add this item to their favorite list called "My List" shown on 2.4 and tab bar

2.4 My List Screen



MY List

Nyx
Propel My Eyes Mascara

CoverGirl
Clean Fresh Cooling Glow Stick

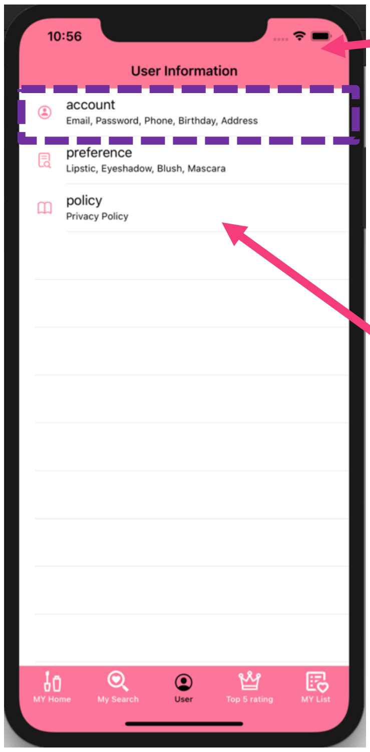
Nyx
Rouge Cream Blush

maybelline
Maybelline Color Sensational Creamy Ma...

The selected item is added to their favorite list called "My List" and can be later navigated on the tab bar

III. My User Tab

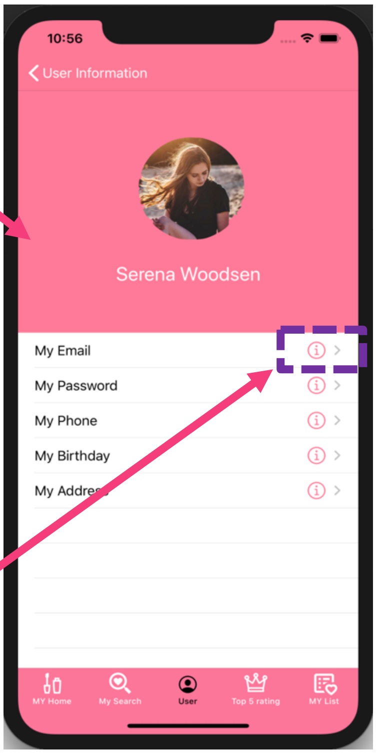
3.1 Account Screen.



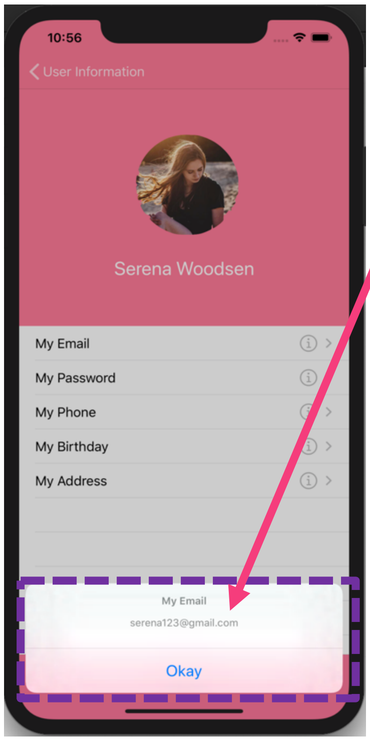
If users choose 'account' table view, it goes detail user table views. User can check the detail information about email, password, phone number, birthday, address.

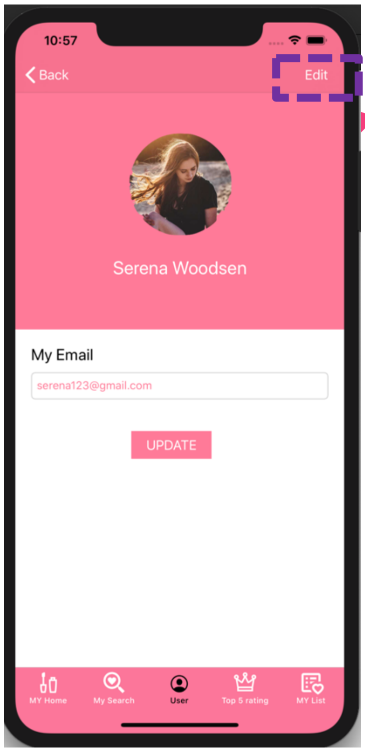
Main User page, three table views that users can click.

3.2 Account detail Screen 1-1



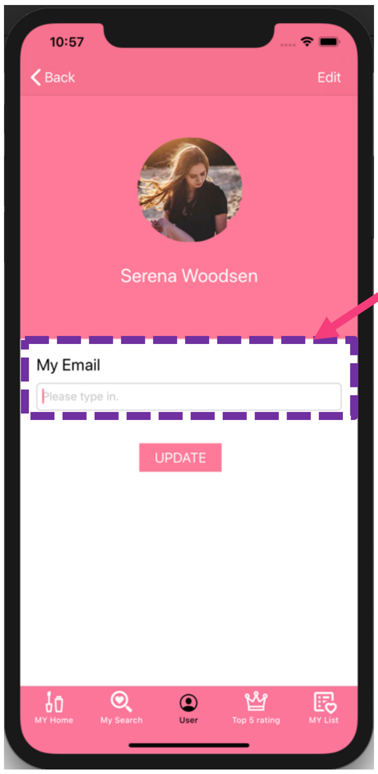
Click the accessary button,
it shows the detail of email information.



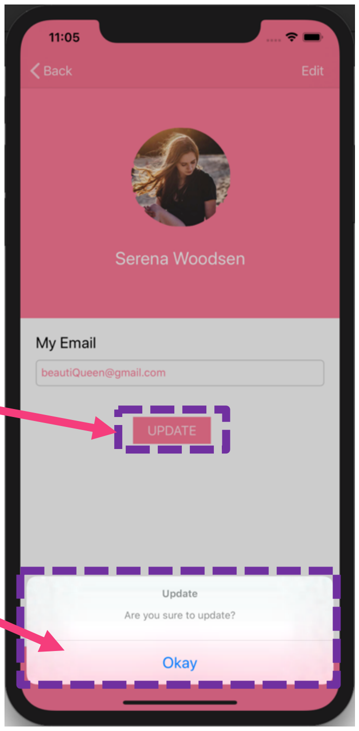


Also, click the table view, it goes the detail view and user can edit the detail information by using 'Edit' button.

3.2 Account detail Screen 1-2

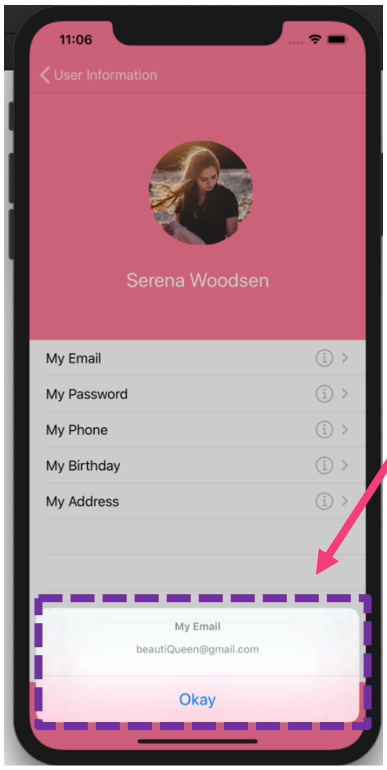


If users touch the 'Edit' button, My Email's textfield activated and users can write a new email address.



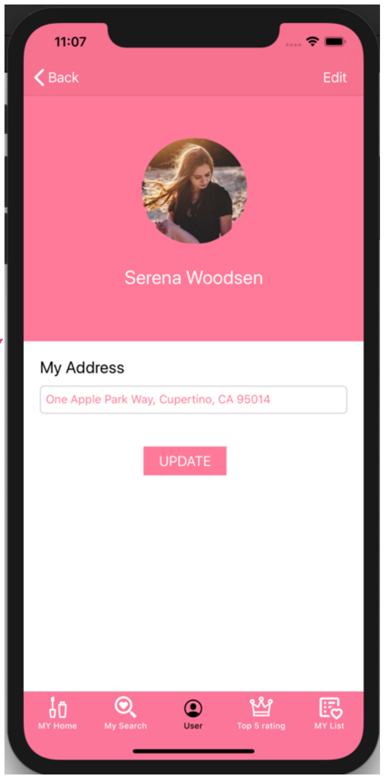
If users click the 'UPDATE' button, it will show an action sheet and let them know, your textfield is going to update.

3.2 Account detail Screen 1-1



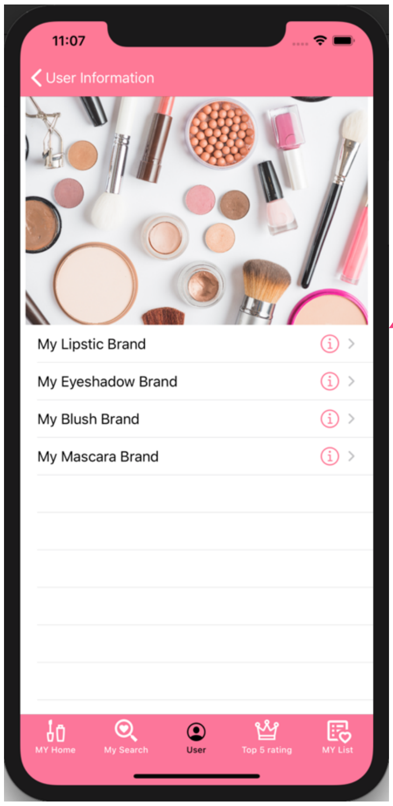
After users click the back arrow (<Back), users click the accessary button again and check the updated information.

3.2 Account detail Screen 1-2

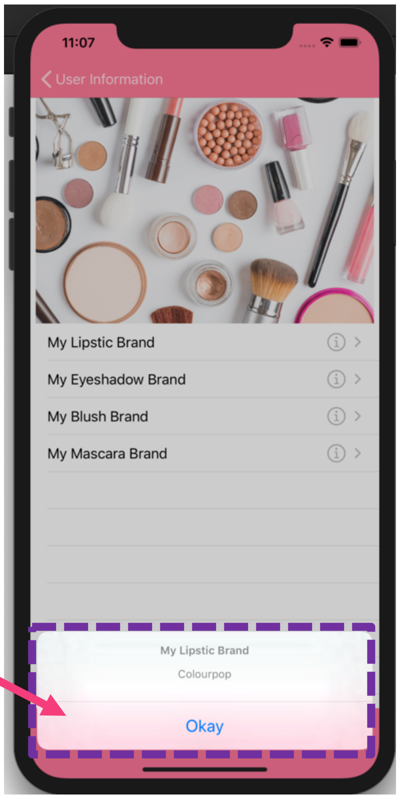


Other tableview-title functions work as same way as above.

3.2 Account detail Screen 2-1

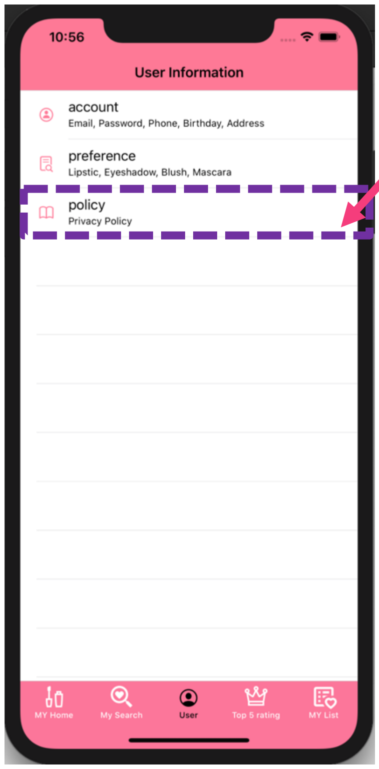


If users click the 'preference' tableview, it goes this page.



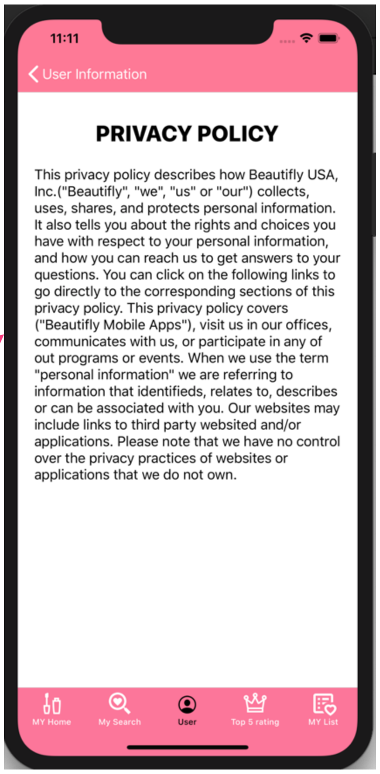
Same way, users click the accessary button, it pops up the actionsheet.

3.2 Account Screen



Back to main user page, if users click the policy tableview, it will go to the detail policy page.

3.2 Account detail Screen 3



PRIVACY POLICY

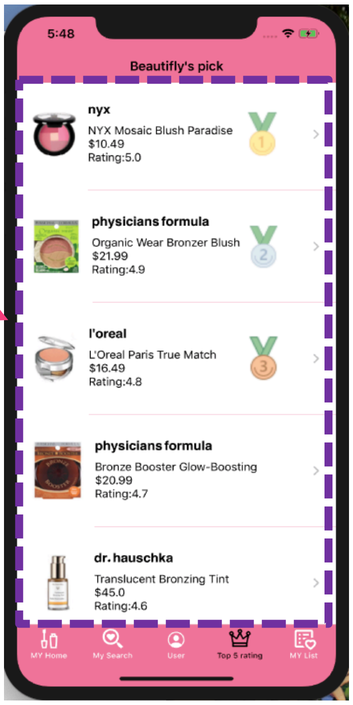
This privacy policy describes how Beautifly USA, Inc.("Beautifly", "we", "us" or "our") collects, uses, shares, and protects personal information. It also tells you about the rights and choices you have with respect to your personal information, and how you can reach us to get answers to your questions. You can click on the following links to go directly to the corresponding sections of this privacy policy. This privacy policy covers ("Beautifly Mobile Apps"), visit us in our offices, communicates with us, or participate in any of out programs or events. When we use the term "personal information" we are referring to information that identifieds, relates to, describes or can be associated with you. Our websites may include links to third party websited and/or applications. Please note that we have no control over the privacy practices of websites or applications that we do not own.

It shows the detail about privacy policy.

IV. My Top 5 Rating Tab

4.1 My Top 5 Screen

Show all list of top 5 items that is rated by user from API


5:48

Beautifly's pick

nyx
NYX Mosaic Blush Paradise
$10.49
Rating:5.0

physicians formula
Organic Wear Bronzer Blush
$21.99
Rating:4.9

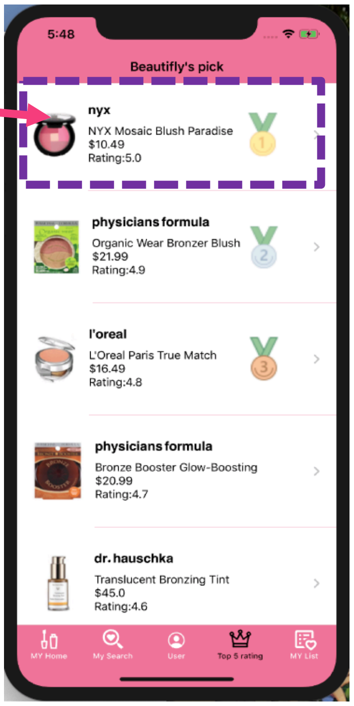
l'oreal
L'Oreal Paris True Match
$16.49
Rating:4.8

physicians formula
Bronze Booster Glow-Boosting
$20.99
Rating:4.7

dr. hauschka
Translucent Bronzing Tint
$45.0
Rating:4.6

4.2 Details Screen

Select any from top 5 items by clicking at the cell on the list. It will trigger the program to load product details from API and show them on product details after click.


5:48

Beautifly's pick

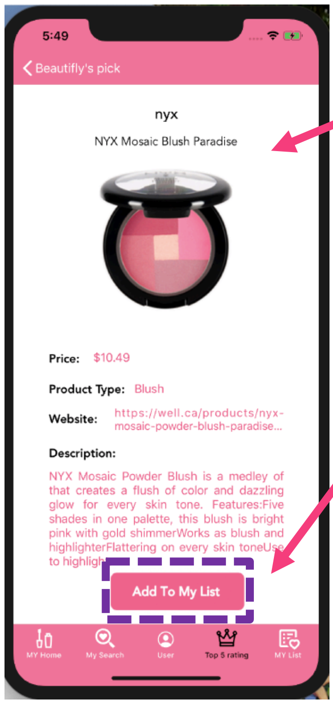
nyx
NYX Mosaic Blush Paradise
$10.49
Rating:5.0

physicians formula
Organic Wear Bronzer Blush
$21.99
Rating:4.9

l'oreal
L'Oreal Paris True Match
$16.49
Rating:4.8

physicians formula
Bronze Booster Glow-Boosting
$20.99
Rating:4.7

dr. hauschka
Translucent Bronzing Tint
$45.0
Rating:4.6


5:49

Beautifly's pick

nyx

NYX Mosaic Blush Paradise

Price: $10.49

Product Type: Blush

Website: https://well.ca/products/nyx-mosaic-powder-blush-paradise...

Description:

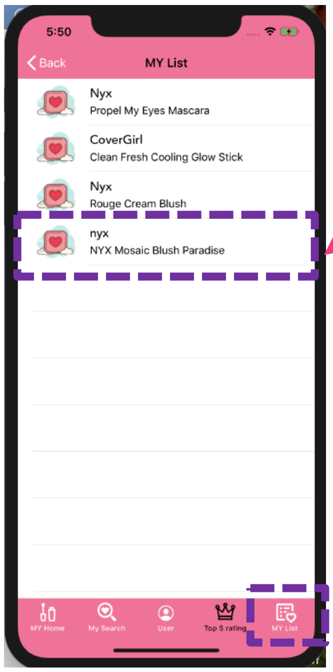
NYX Mosaic Powder Blush is a medley of that creates a flush of color and dazzling glow for every skin tone. Features:Five shades in one palette, this blush is bright pink with gold shimmerWorks as blush and highlighterFlattering on every skin toneUse to highligh

The details of top 5 selected item show on this screen including name, image, brand, price, type, website, description.

User can click "Add to My List" Button to add selected item to favorite list called "My List" at the bottom of tab bar in screen 5.1 on next page

V. My List Tab

5.1 My List Screen (Add to List)



MY List

Nyx	Propel My Eyes Mascara
CoverGirl	Clean Fresh Cooling Glow Stick
Nyx	Rouge Cream Blush
nyx	NYX Mosaic Blush Paradise

The top 5 selected item is added to their favorite list called "My List" and can be later navigated on the tab bar

PART B: Final Project Discussion

i. API features

Make-Up API (Not covered in lecture):

All data related to make-up products presented in this application are retrieved from Make- Up API by herokuapp using asynchronous request. In MySearch view, it received data of the user's input from the MySearch view and call the function in ProductRequest class to generate new URL customized by user's input and request response from a make-up API. The destination URL is http://makeup-api.herokuapp.com/api/v1/products.json?brand=\(searchBrand)&product_type=\ (searchType). After getting response from API as JSON arrays, I preprocessed JSON arrays to create a Product Object which contains all data of one product such as its imageURL, name, prices, and descriptions. Those data will be reused in other functions to perform tasks in MyListViewController, MyResultTableViewController, MyResultTableViewController, and ProductDetailViewController. The specific use of this make-up API in each view are explained in the following parts.

Tabbed Application:

Beautifly is a Tabbed Application which contains 5 different tabs which user can select 5 features from them. I customize images of tab items, create the functionality for MySearch and MyList tabs. The MySearch and MyList view controllers are connected with Tab Bar using relationship segues. The MySearch tab takes user's keyword input, performs searching products on a make-up API, getting response from API, and presenting search results. It also shows product details retrieved from API and allow user to add selected item to his favorite list on MyList tab.

Navigation Controller:

For presenting all views in MySearch tab for searching items and showing the search results and product details, I have relied on the use of navigation controller in UINavigationController class to present a hierarchical structure of those 4 views as follows: search view, found-item list view, product details view, and favorite list view. A navigation bar with a back button shows on all screens in which MySearchViewController is a root view controller and connected with search view. The user can fully navigate all 4 views through navigation bar.

Dynamic Table View:

MyResultTableViewController is connected to the Found list view. The dynamic table view is used because we do not know the specific number of found items (number of rows) ahead. This view can be scrollable and selectable. The MyResultTableViewController received data of the user's input from the MySearch view and call the function in ProductRequest class to generate new URL customized by user's input and request response from a make-up API. After getting response from API, the search results (All Matched Items) will be listed on the Found list view. The row is populated using dequeueReusableCell with the unique identifier named "foundItem." User can then select specific row on the list to see more details of selected product. Each prototype cell displays its brand and product name for 1 matched item.

The Alert Popup:

UIAlertController is utilized in MyResultTableViewController to ensure that user types the make-up brand in a search box and to confirm product type that user selects before the app begins to send a request to a make-up API. Additionally, another customized alert popup will display to inform the user if there are no matched items from on API.

Text Input and Buttons:

The MySearchViewController takes a user's input of product brand and product's type as keyword to pass it to function to create URL and request a response from Make-Up API. For product brand, the user can type the brand they want to find in the search box connected an outlet of the UITextField and connect a didEndOnExit action in the MySearch view controller. The didEndOnExit function takes sender as an input to call the resignFirstResponder()to resign first responder for the searching text field when user presses "Return" Key. The 4 Buttons in UIButton class are utilized to allow users to click buttons to select the product's type including: lipstick, mascara, eyeshadow, and blush. These buttons are connected to the actions in the MySearch view controller When buttons are clicked, they trigger the actions to set the product types for searching. The "submit" button will pass the info of product brand (User's text input) and product type selected by user to the MyResultTableViewController" by overriding prepare function.

Loading Details from Make-Up API and Image from URL:

The ProductDetailViewController is connected with the Product Details View which will display all details of selected product that user selected from the Found list view. Like the found list view, the details view also call the function to request the response for downloading image of products by passing the imageURL using asynchronous request. After getting the response, the image will be set to and display in the UIImageView in the Product Detail View. The view also contains "add to list" button that will update the dynamic table view in the MY List view.

ii. Challenges

The first challenge is that I need to limit the number of product types that user can search from make-up API because this API grouped data into 4 types. Thus, I solved this problem by using 4 buttons for each product type: lipstick, mascara, blush, and eyeshadow that allows user to select at button, so we can send the correct request of product type to API. Another challenge is how to efficiently and correctly process the product's data from JSON array retrieved from a make-up API. The data will be reused in many view controllers such as MySearch, MyTop5, MyList, Found item List, and Product Detail view, so I need to ensure that I received the correct data from API and correctly stored in object. Also, one product's data contains almost 20 keys in JSON data. Thus, my team and I need to research more about passing and processing JSON data in Swift. Then, we decided to create a Struct "ProductDetail" that is decodable to save product's info including: id, brand, name, imageURL, prices, etc. This struct object will be used in the "Product Details" View. Additionally, Struct "Products" is used to save the ProductDetail of all products found on API by user's keywords and iterate over it in the Found-List view" to show the search results from API. By doing this, we have overcome challenge.

iii. Limitation of Beautifly Application

The limitation of the search feature in the app is that it mainly relies on only one make-up API from by herokuapp for retrieving product's data. However, this make-up api contains a small set of make-up brands (50 brands in totals), so they user may not able to find all brands they like through the search feature in our app. Moreover, the data from API don't include luxurious brand and provide are only affordable make-up brands such as ColourPop and Maybelline. Thus, it limits our Beautifly app to offer more varieties of products and we target only in users who looks for affordable not luxurious products. For the future development, we need to add more API to retrieve a wider range of data for all make-up products and use the picker to allow user to select the brand from the list to ensure that they can search for the brand that available on make-up API.

iv. Limitation of SDK/SWIFT

The first limitation that I found there is some bugs on iOS 13 that haven't been fixed such as action sheet. When I use the alert sheet, it gives constraint error which appear on IOS12.2 and beyond, so I can only file a bug report to Apple and have to limit our use of action sheet to avoid crashing our app. Another limitation is the AutoLayout part which we need to work on it separately for each device version. I found that layouts ofscreens do not look nicely on a device with a smaller screen like iPhone 5. So, this limits our design goal to mainly focus only on new versions of iPhone.

v. Overall experience

As an iOS developer, I have applied many concepts that I learned from class to this project. My team and I developed an application that is useful for the potential users. It's a great starter to build on this app in the future to provide a wider range of data to a larger group of target users. Before taking this class, I didn't know Swift and functionality for developing iOS app. From doing this project, I gained more knowledge on iOS application and get more proficiency in coding in Swift and using XCode. The most valuable experience for me is that this project gives me an opportunity to work with other iOS developer as a team, and it gives me a great opportunity to bring my creativity and imagination into the real-world application.

PART C: Final Project Contribution

List of Specific Contributions:

Our team divided all works by the tabs in Application. My main responsibility is tied to MySearch tab and MyList tab. I created screens, view controller, navigation bar, tab bar, and functionality for MySearch tab and MyList tab including Search View, Found-List View, and Product Detail View and My List View. Specific tasks and contributions are listed below.

a) Customized images of tab items, buttons, and alert box for views related to MySearch tab
b) Designed the app icon and Home view except the multi-touch portion
c) Designed and built the whole screens of Search View, Found-List View, Product Detail View, and My List View in Main Story Board
d) Created the MySearchViewController, MyResultTableViewController, ProductDetailViewController, and MyListTabelController.
e) Connected all views in part c with all ViewControllers in part d
f) CreatednavigationcontrollerandbartoconnectallviewsforMySearchtabandMyListtab
g) Wrote programs to receive user's input from MySearch tab and send the request and receive the response from a make-up API for a "search" and "add to list" features.
h) Wrote programs in each view controller in MySearch tab and MyList tab to perform all features needed such as searching for products, sending request to API, processing JSON data, downloading and presenting product details.
i) Wrote programs to send request for downloading image from URL and displaying image on the product-detail view and connected to the Search and Product Details Views.
j) Wrote programs to pass the user's input and selected product types to another view for sending request to make-up API and retrieved data back to display them in Table View to ensure the completed functionality of Search and Add to List features.