Github username: crazyhitty

Predator for Product Hunt

12 December 2016

DESCRIPTION

A beatifully crafted client for https://www.producthunt.com/ . Get the latest tech products on your fingertips. View latest products and collections uploaded on product hunt. Application contains many features like ability to bookmark any product or collection, view other user's profile, view product details and much more.

INTENDED USER

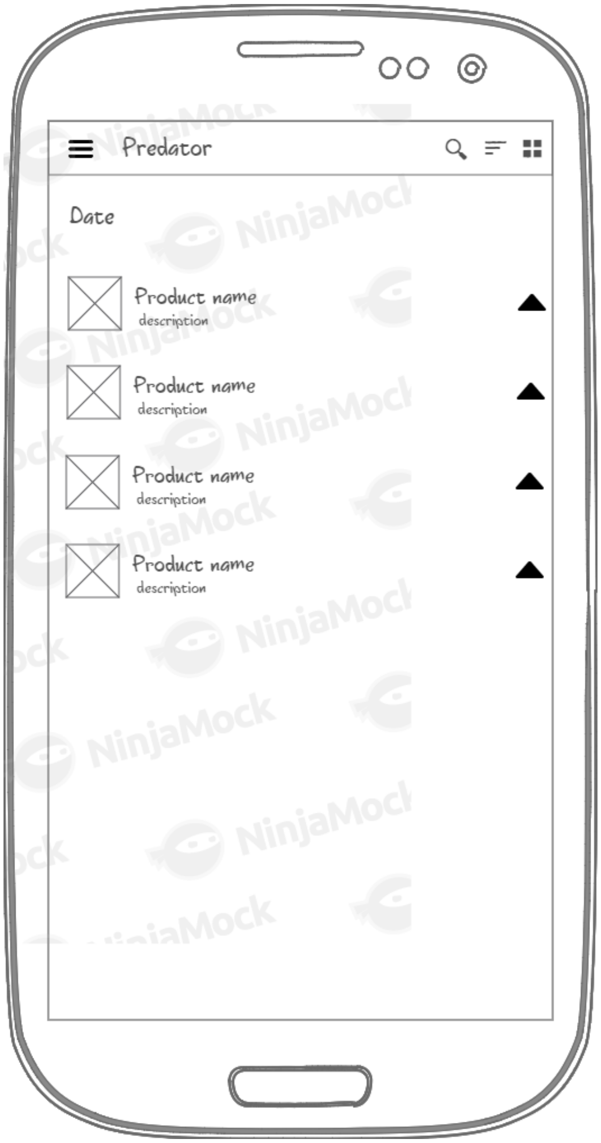
This application is for a tech savvy user who regularly looks out for new and better products. This app can help such kind of users to view latest products and check them out via the application.

FEATURES

1. Show the product listing sorted by categories.
2. View product details.
3. View comments shown on product details.
4. Login and view your profile as well as view profiles of other users.
5. Upvote the products (Only authenticated users).
6. Make comment on a product (Only authenticated users).
7. Share the application content with other apps.

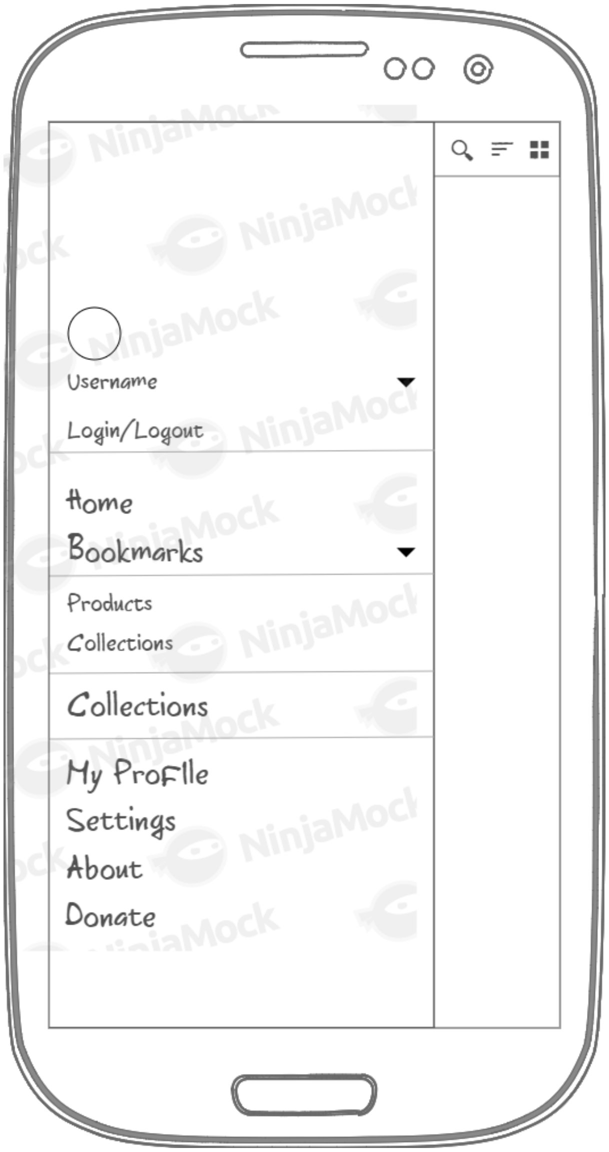
USER INTERFACE MOCKUPS

Home screen



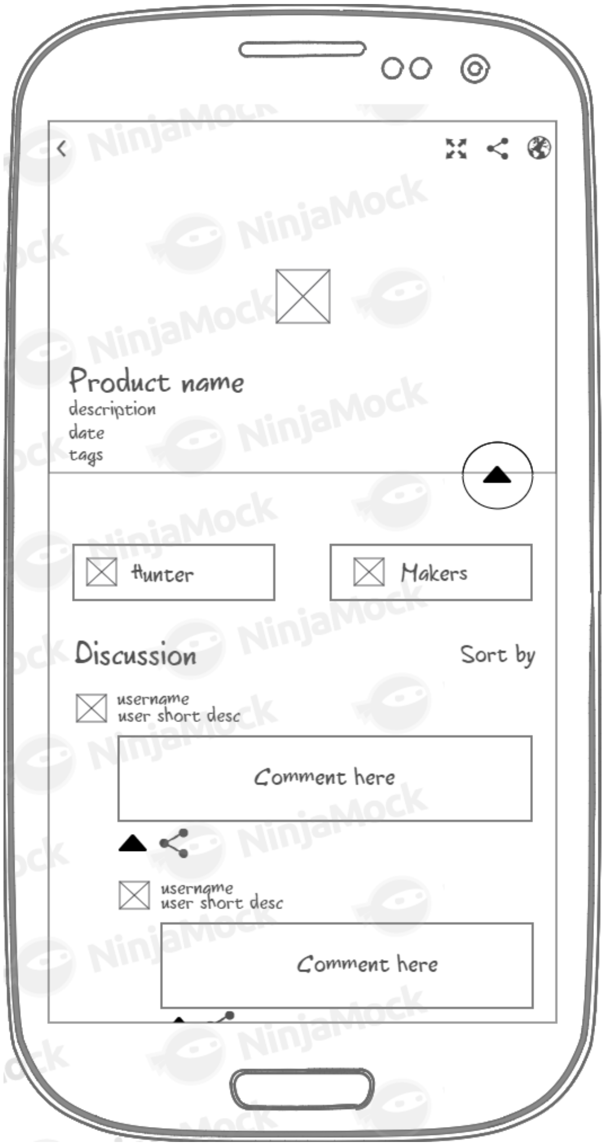
Show product listing on the screen retrieved from the product hunt api.

Navigation drawer



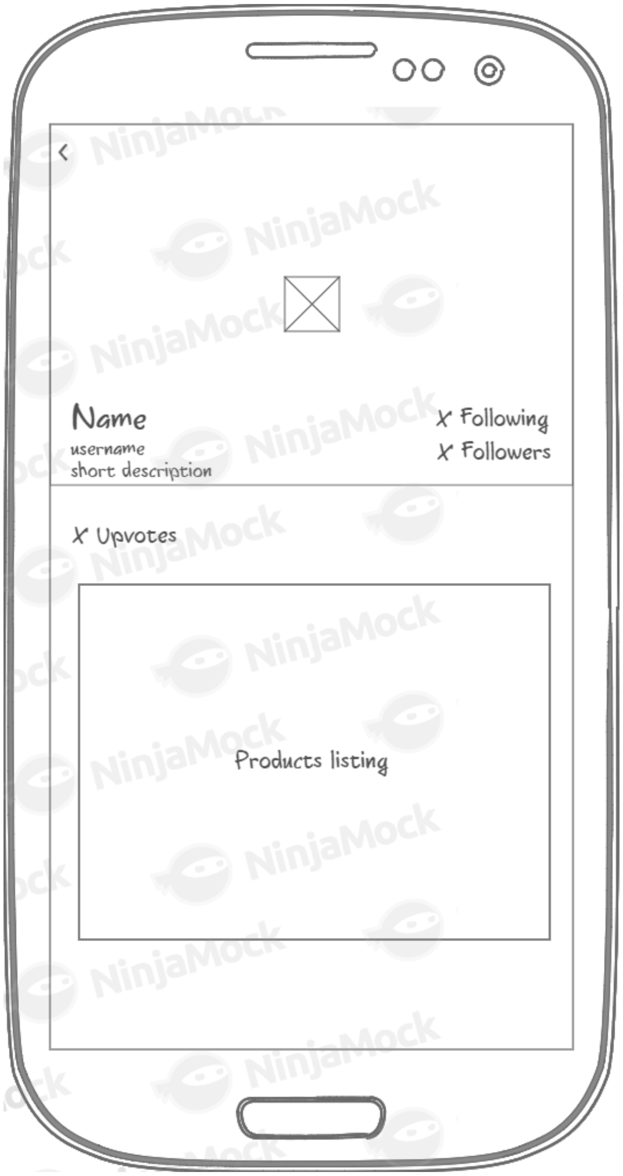
Login/logout buttons will be visible if user taps on navigation header. Products and collections will be sub menus of the primary menu “Bookmarks”. Navigation header will contain a backdrop image.

Product details



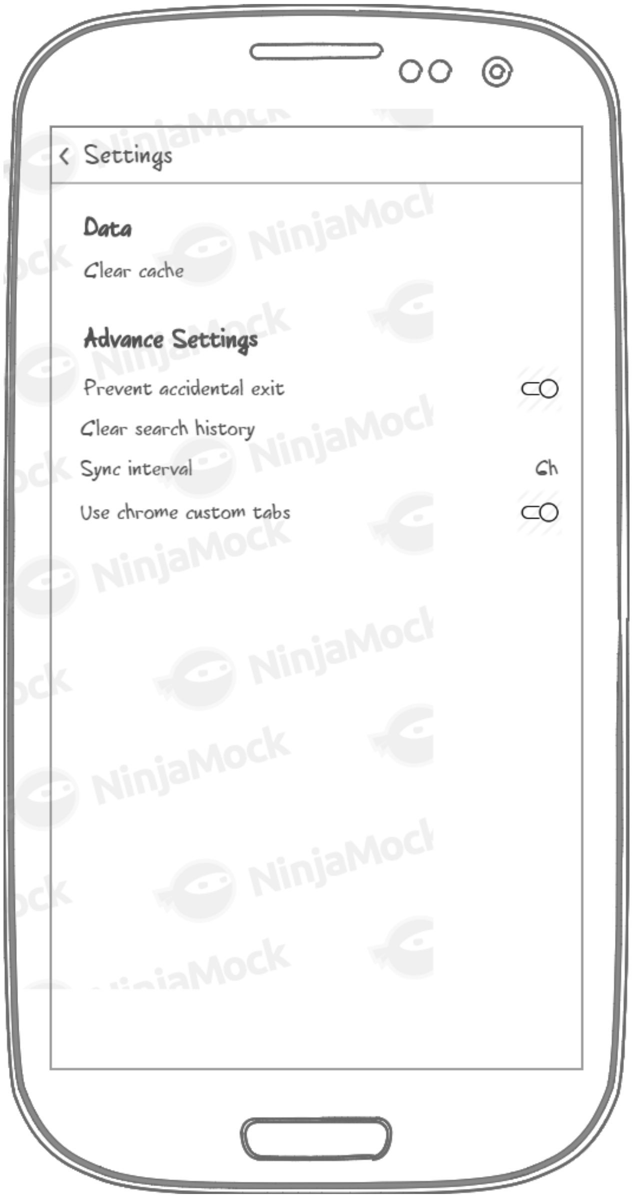
This screen will open whenever user clicks any product on products listing screen. User can view the product images in fullscreen, share the product details and view its webpage too. Also, if the user is authenticated then, he/she can upvote and make new comments on that product.

User profile



Will contain total upvotes as well those products that were upvoted by this user. My profile as well as other user's profile screen will be similar.

Settings



< Settings

Data

Clear cache

Advance Settings

Prevent accidental exit ○

Clear search history

Sync interval 6h

Use chrome custom tabs ○

Through settings, user can perform certain actions:

1. **Clear cache:** Clear image cache.
2. **Prevent accident exit:** Override back button functionality in home screen.
3. **Clear search history:** Delete all search items which will be shown in autocomplete.
4. **Sync interval:** Interval of sync adapter, user can also turn it off.
5. **Use chrome custom tabs:** Use custom chrome tabs or use system's browser.

KEY CONSIDERATIONS

How will your app handle data persistence?

Application will use content provider and shared preferences to maintain its data locally.

Describe any corner cases in the UX.

1. **Orientation change while API request is being made:** App might crash in such circumstances.
2. **User rapidly moves in b/w app screens:** App might lag or slow down.
3. **User closes the application while a API request was taking place:** App will crash in such circumstances.
4. **Loading a large image in the app:** App might crash in low end devices due to out of memory issue.

Describe any libraries you'll be using and sharing your reasoning for including them.

1. **Design support library:** For material goodness.
2. **Retrofit:** For managing api requests.
3. **Gson:** Json to Pojo and vice versa.
4. **Stetho:** Debugging android applications over chrome.
5. **RxJava/RxAndroid:** For managing asynchronous tasks.
6. **Butterknife:** Removing the boilerplate code like "findViewById".
7. **Fresco:** Image caching library.
8. **Firebase:** Crash analytics and push notifications.

Describe how you will implement Google Play Services.

Application will be using firebase crash analytics and push notifications which are dependent on google play services.

REQUIRED TASKS

Task 1 : Create wrapper for product hunt api.

Create a wrapper as a separate module around product hunt api so that it can be easily used in the application. These web services would be included in the wrapper:

1. Authentication.

2. Products listing.
3. Search products.
4. Product details.
5. Get all comments.
6. Make a comment.
7. Upvote product.
8. Get collections.
9. My/other user profile.

Task 2 : Write tests for the wrapper.

Write tests for wrapper so that we can check if our wrapper is working correctly or not.

Task 3 : Create UI for the application.

Create proper UI for the application, these screens will be created:

1. Dashboard screen - Activity
   a. Product listing screen - Fragment
   b. Collections screen - Fragment
   c. Bookmarks
      i. Products screen - Fragment
      ii. Collections screen - Fragment
   d. Login - Dialog
   e. Logout - Dialog
2. Product details screen - Activity
3. My/other user profile screen - Activity
4. Settings screen - PreferenceActivity
5. About screen - Activity

Task 4 : Integrating wrapper functionality with UI.

Showing data retrieved from wrapper properly on screens. Also, handling various cases like handling internet connection and showing proper error messages.

Task 5 : Adding data persistence.

Introducing content provider into the application to store the data locally in order to show data at any available time regardless of the device's network state. Also, integrate sync adapter in the application to save the data in frequent intervals.

Task 6 : Write UI tests.

Create UI tests to automatically test the application UI.

Task 7 : Prepare app for release.

Fix out some of the minor bugs remaining and prepare the play store listing. Steps to follow before publishing to play store:

1. Create app icon.
2. Create splash screen for app.
3. Prepare store listing.
4. Create video demonstrating application.