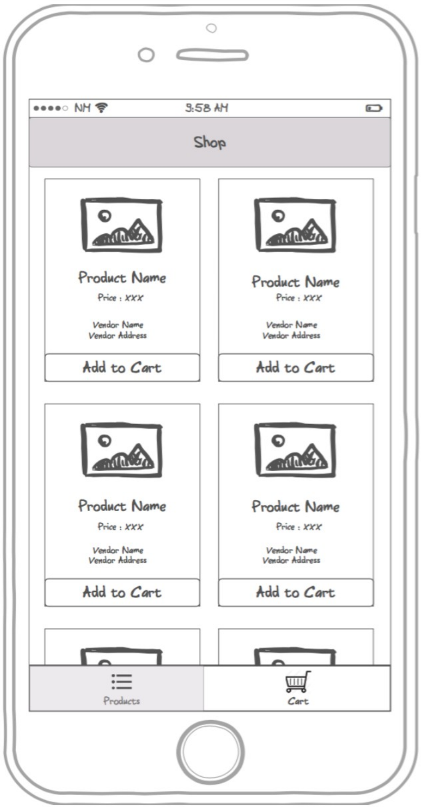
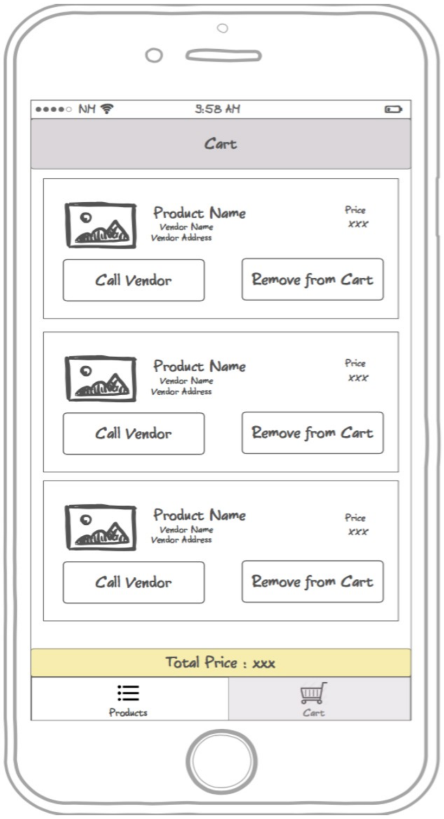
Assignment:

Create a simple application having screens and functionality as shown in the Template1 and Template2.

Template1



Template2.



Details:

1. Fetch all the product details from API (‘’https://mobiletest-hackathon.herokuapp.com/getdata/”).
2. The app has 2 tabs having the Products and Cart screens.
3. Products tab will display all the available products as shown in Template1.png. Every single product has information like Product Image, Name, Price, Vendor Name and Vendor Address. All this information should be displayed exactly as shown in the template.
4. For every product there is “Add to Cart” button available. On adding to cart, the product should move into a cart. (It needs to be handled locally. No need to maintain it on server side).

5. On selecting the Cart tab, all the products available in cart should be displayed in a list as shown in Template2.png. The required product details are Name, Price, Vendor Name, Vendor Address.
6. Every product in Cart has a functionality to “Remove from Cart” and “Call Vendor”. On removing from cart, the product should be removed from the list. On “Call Vendor”, the phone call should get initiated for the vendor’s number.
7. In Cart’s tab, you need to display the Total Price of all the products added in cart. The price should get updated when you add/remove any product from cart.

Expectations:

1. Proper coding standards should be followed. Try to write optimised code.
2. Class designing and structuring should be according to design patterns (e.g. MVC)
3. For UI designing, try to make use of latest available concepts.
4. App should provide support on different resolution and dimension based devices.
5. App should support both the orientations (Would be an added advantage)

Note:

1. *Don’t miss out any product details while consuming the API.*
2. *Code should be written considering future enhancements/changes in the assignment.*
3. *Design the assignment to handle dynamic data.*

For iOS:

Its expected to make use of (not all but if needed) UITableView, UICollectionView, Storyboards, Autolayout, SizeClass, Delegates, Blocks, lazy loading.

For Android:

Its expected to make use of (not all but if needed) CardView, ScrollView, ImageViewer, Custom Layouts, RecyclerView, lazy loading.

Tips

1. Feel free to use third party libraries for network operations and anywhere else.
2. Feel free to use Google, Stackoverflow, etc.
3. See if you can be creative with your architecture even if you think it’s overkill.
4. Be sure to have a deliverable at the deadline.
5. Less is more.