Final Project - Amazon Price Tracker

Alexis Charpentier
McKensie Sherlock

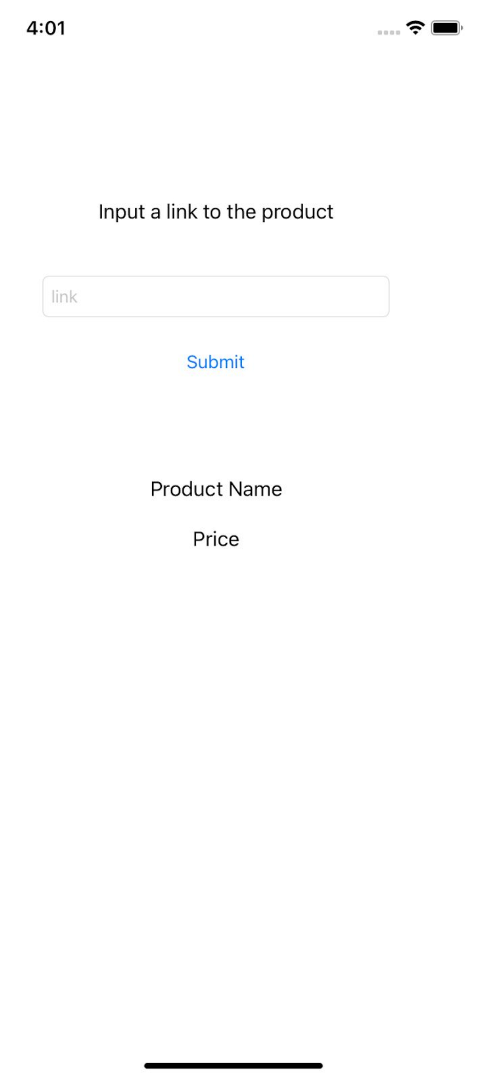
Summary

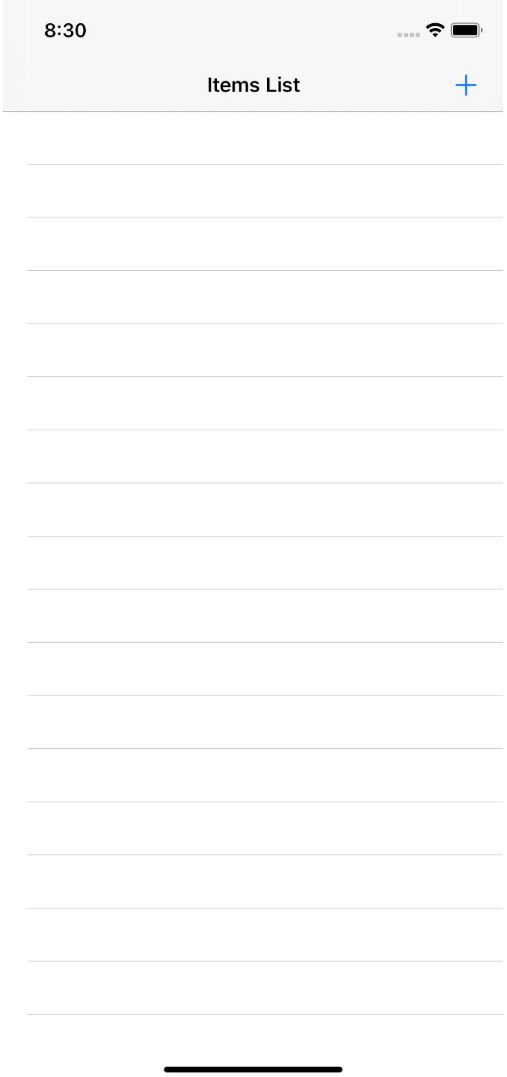
This app's original goal was to be a price tracker that monitors the prices of an item from Amazon. The user is notified when the price of the item drops below a user-specified threshold. The app will store the price changes so the user may identify any sale trends.

Methods

The cocoapod, SwiftSoup was downloaded and installed to the project. The program homepage displays a list of products inputted by the user. This default view controller contains a "+" button at the top right corner of the screen which leads the user to a second view controller that will request an amazon link to an item. The first view controller seen on startup, ListViewController, has been binded with a Navigation Controller. This facilitates easy and straightforward navigation between the as of now two view controllers. These two view controllers are attached below in the "Results" section. With the user-inputted link, the program parses the html data to find the defined item name and price. To more easily parse the html data, the Swift library SwiftSoup was used. The product name is defined by the title of the html while the price is defined by this class: "a-size-medium a-color-price priceBlockBuyingPriceString." Once these values are defined in the code as variables, they can be displayed for the user on that same page as well as on the list controller using outlets and actions connected to those variables.

Results





Discussion

While restrictions caused by COVID-19's sudden increase in cases in Rhode Island did negatively impact the development schedule of the project, progress was still made. Of the goals outlined in the previously submitted Software Requirements Specification, the initial stages of an amazon item's price and name collector, as well as the structure of an item list, were created. However, collecting the price of an item proved more complicated than initially expected. Amazon displays prices in different ways for different circumstances, for example, textbooks often list multiple price choices for hardback, paperback, as well as renting options. Another example is when an item is on sale--the product page displays the item's original price with a strikethrough, as well as the new marked-down price. It was discovered that the classes that correspond with these different price display methods differ from the class used above. Steps will need to be taken to either find a more overarching class that corresponds to a value that doesn't change depending on how the varying ways the price can be displayed on the product page, or, write code that goes through and accounts for all the different ways amazon product prices can be displayed, if the product page is not displaying the vanilla single-price method.

To meet the original goals of this app’s functionality, a pop up or additional view controller for the user-inputted notification settings will need to be implemented as well as memory storage for the product list. This app requires memory storage to function as desired so the user can view the price at any time with no additional inputs and be notified when a product’s price drops below a user-specified threshold. Additionally, this app could be improved by accommodating other suppliers than Amazon. By using conditional statements, the program would be able to identify the website and the classes needed to define the name and price of the desired product.

Documentation

Please see the .zip file attached in the email.