Demo - iOSStore

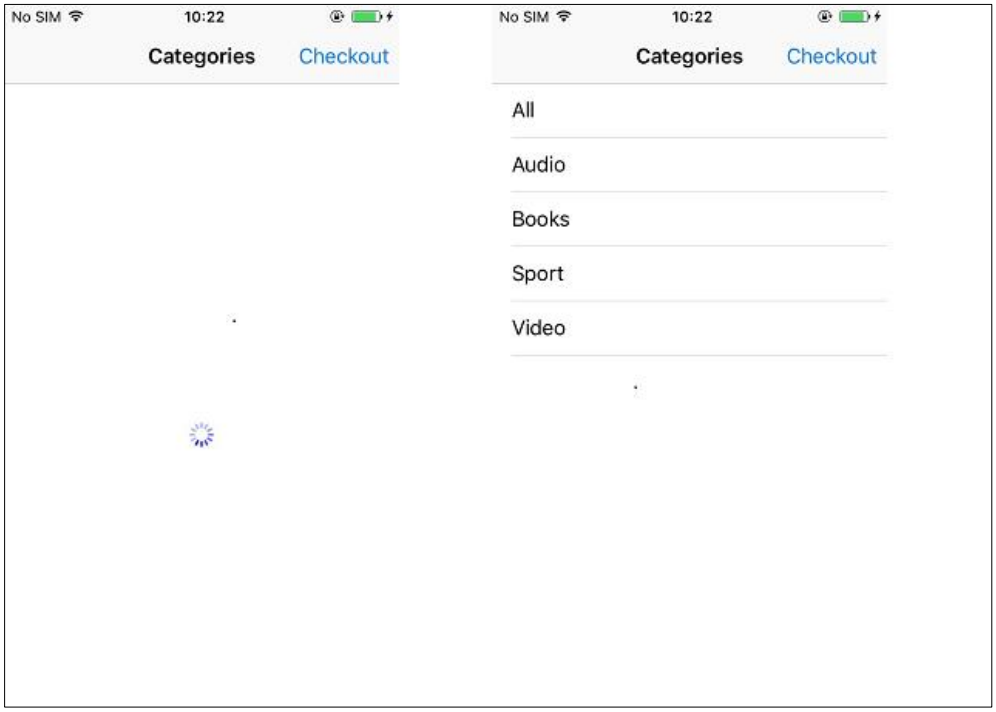
Application that was developed for this demonstration is a shopping application called iOSStore.

Application dataset consists of products that are being retrieved from a web API. Web API is created in .NET, with ASP.NET Web API, Entity Framework, and MS SQL Server for underlying storage. It is hosted in IIS on a local computer (PC). The return format is JSON. The following lines are showing part of JSON data returned for one product:

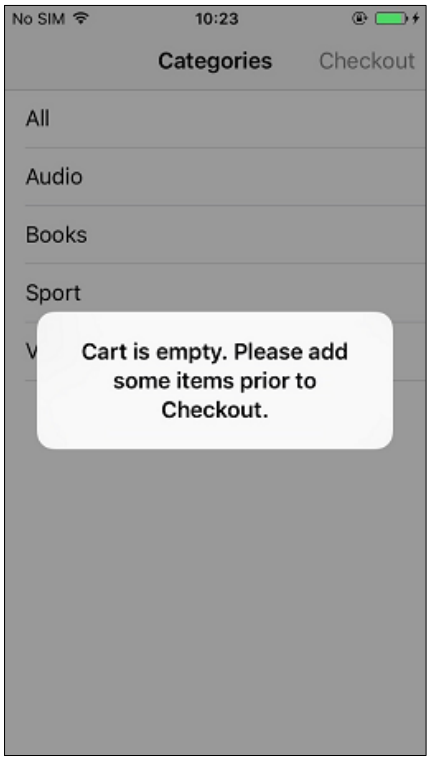

```
{
        "ID": 2,
        "Name": "Crime And Punishment",
        "Description": "Dostoevsky's masterpiece",
        "Category": "Books",
        "Price": 50.00,
        "Image": "iVBORw0KGgoAAAANSUhEUgAAA…"
        "ImageUrl": "https://upload.wikimedia.org/wikipedia/en/4/4b/Crimeandpunishmentcover.png"
}
```


All endpoints that are used for products retrieval are fully configurable in iOS application project configuration. Also, the only things that matter are that endpoints are as defined and the products' format is as defined, otherwise web api could be developed in any technology, like Node.JS, Ruby on Rails, etc.

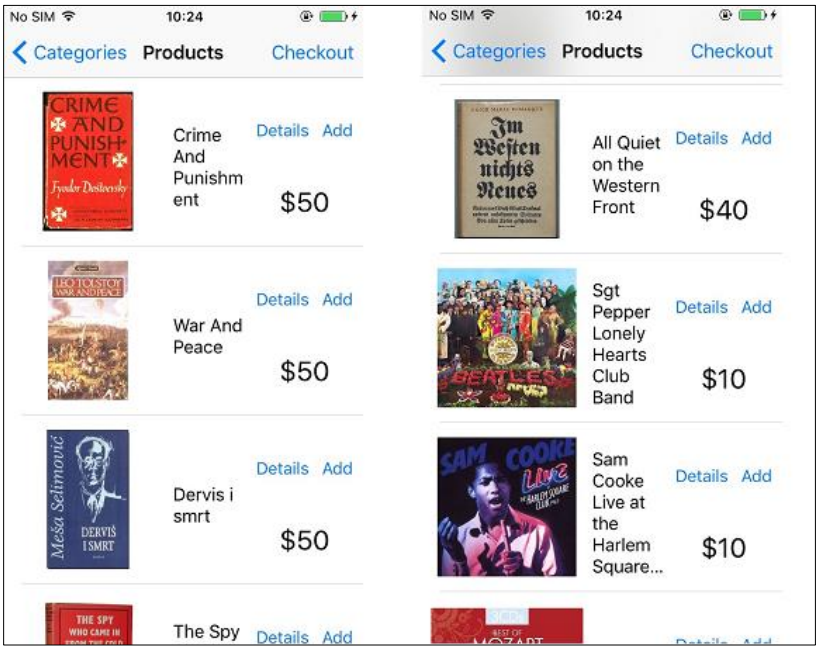
As application is stared, categories of products/articles are shown, like presented on the following screenshot:



User should be able to select a category or click on "Checkout" button. This button should take user to the checkout screen, but if there are no products added to the cart, the info dialog should be shown warning user that some items should be added prior to checkout:

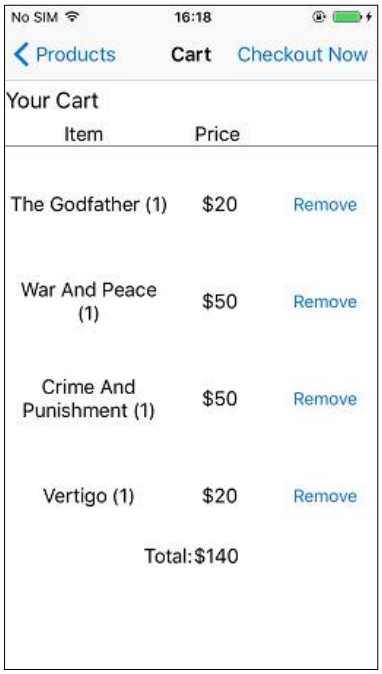


After a category is selected, the products screen is shown, as like on the following screenshot. This time “All” is selected meaning that items should be retrieved for all categories. It should be noted here that every data retrieval from the web api is being done asynchronously, using iOS SDK methods for HTTP requests. If some data is retrieved once it is cached, so it will not need to be retrieved again in the same user session.

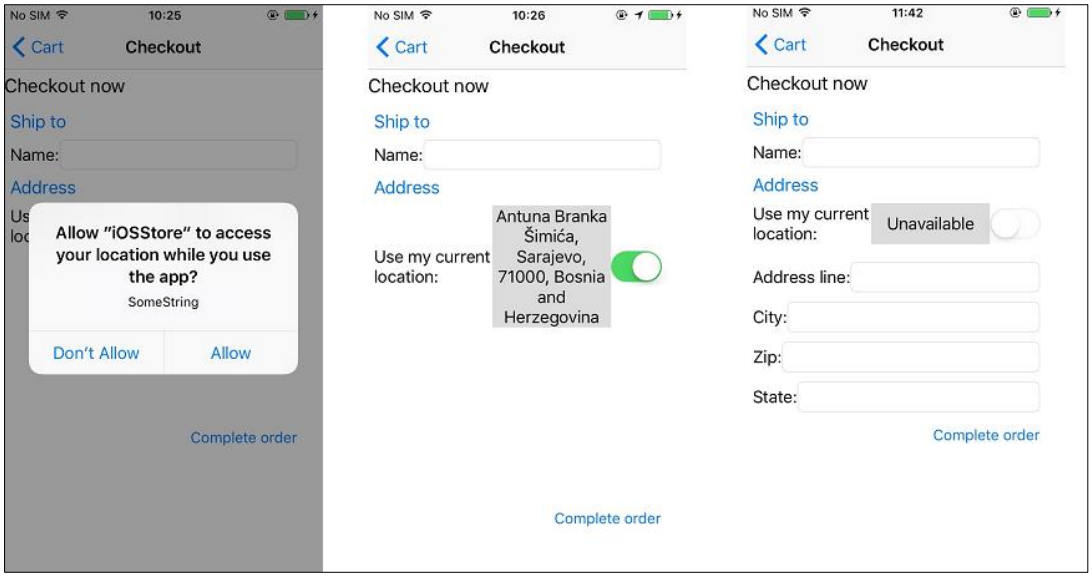


Clicking on “Add” button adds corresponding product to the cart. Clicking “Details” initiate display of detailed view for corresponding product:

When some items are added to the cart, checkout could be performed, which takes user to cart screen, with the list of products, quantities, and total value:



At this point, user could return to products, or further back to categories, and add more items to the cart. Or click “Checkout Now”, and complete the purchase.



The preceding picture shows three scheenshots from checkout screen. The first shows a dialog asking user if he would allow this application to use location services. After clicking “Allow”, the current location address is calculated and preselected. If user does not want this, but he would like to enter the address manually, or location services are unavailable, or location cannot be retrieved for whatever reason, the switch for enabling use of current location is turned off and new form fields for address are displayed (as shown on the right of the picture). The form is submitted on “Complete order” click. Both Name and one for of addresses should be present or form submission will fail validation.

When form is submitted, an email is being sent to the user configured in application project configuration, containing contents of the cart, name and address. External plugin MailCore is used for sending emails. This is a bit clumsy, but I had no time to think about the other way, and anyway this application will be used only for demonstration.

On success, confirmation screen is shown: