Mobile Application Store Collection Service Requirements

Author: Eric Gieseke

Date: 10/10/2013

Introduction

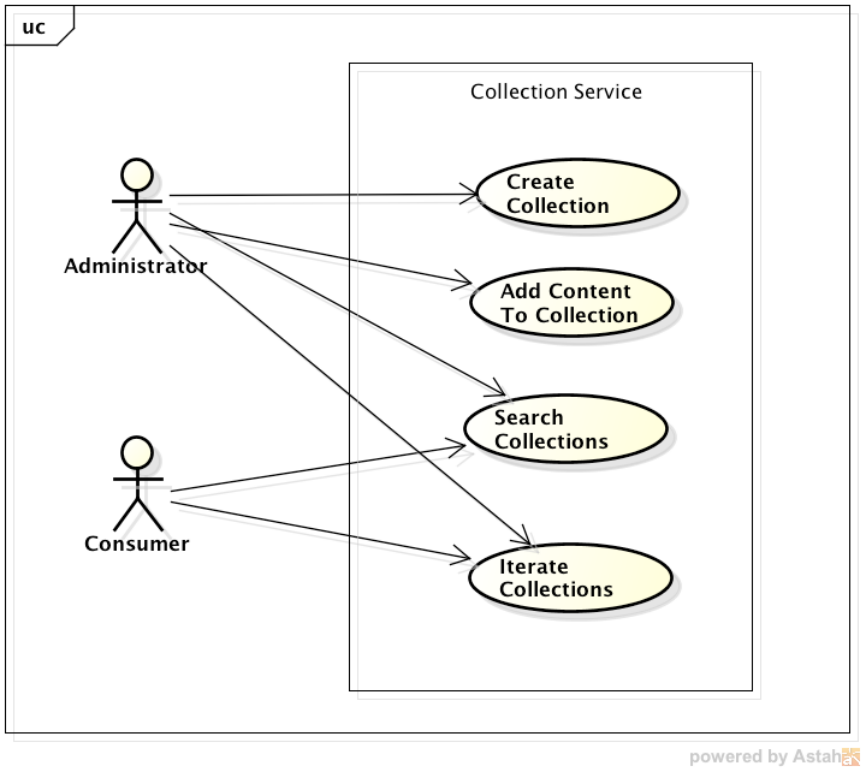
This document provides the requirements for the Mobile Application Store Collection Service.

Overview

Mobile Application stores provide a source of applications, ringtones, and wallpapers for mobile consumers to download to and access from their mobile devices. A wide variety of applications are available, including games, office applications, and social applications. Ringtones and wallpapers allow users to customize their mobile experience. Both free and premium (e.g. paid) content is available for download.

Collection Service

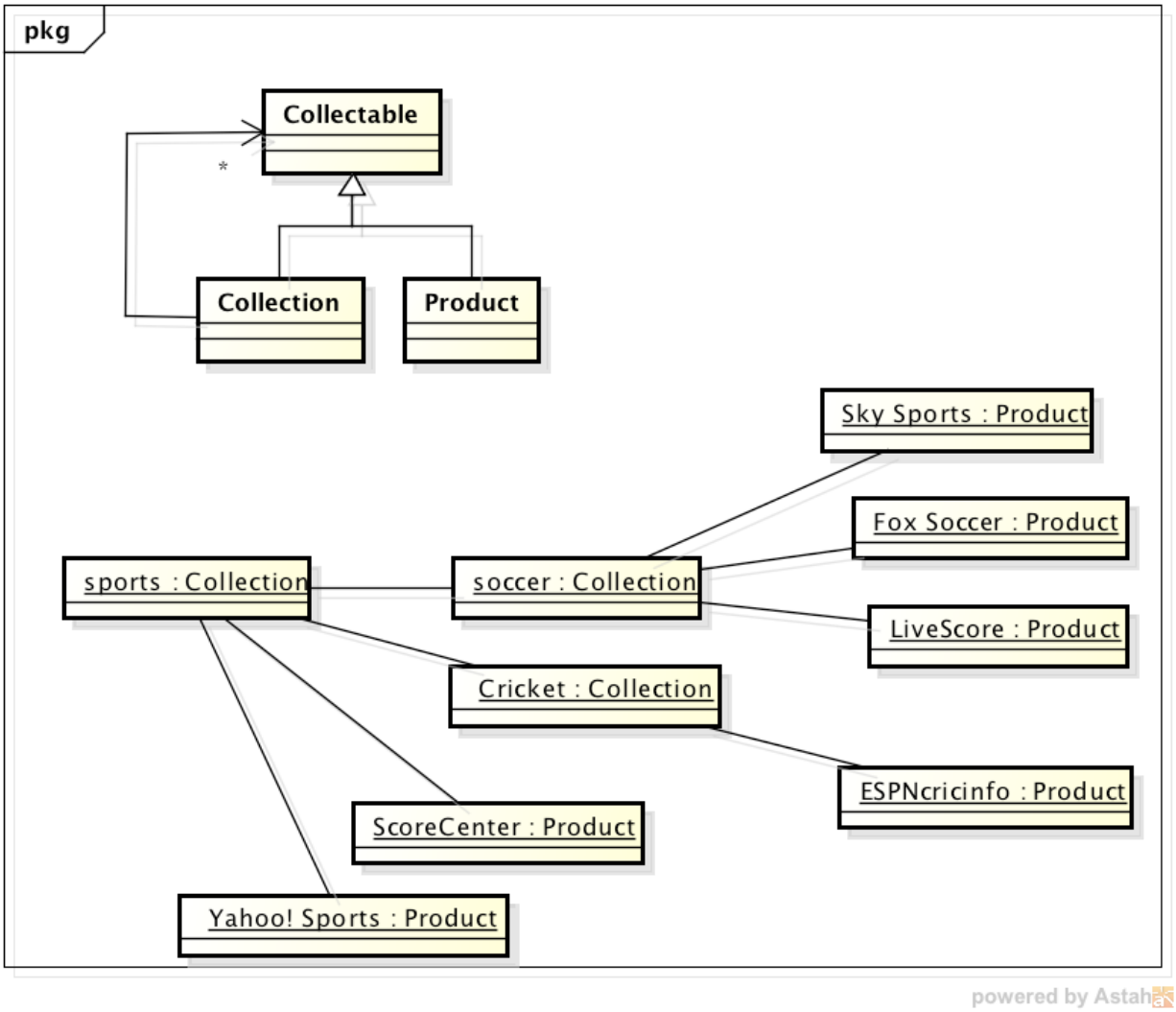
To help consumers find the content that is of interest to them, there is a requirement to provide a way to organize content into Collections.



A Collection Service API is needed to support the following functions:

- creating collections (restricted interface)
- adding content to collections (restricted interface)
- searching collections
- iterating over collections

Collections are recursive structures, they can contain Products as well as other Collections. The following diagram describes how collections and products are related.



Dynamic and Static Collections

To help make collections more flexible and easier for store administrators to manage, the Collection Service will support 2 types of collections, static and dynamic.

Static Collections

Static collections refer to specific instances of products. Products are added to collections using the product id as a reference. There should be a loose coupling between the Collection Service and the Product API, meaning that static collections should use product ids to reference Products.

Dynamic Collections

Dynamic collections are defined using product search criteria. In this way, dynamic collections automatically adjust their content, based on the current Product inventory. As new products are added to the store, dynamic collections will include all content that matches their product search criteria.

Implementation Hint: Leverage the Product API Search interface to dynamically fetch the contents of the collection, when the collection contents are requested by a client. Do not attempt to keep dynamic collections in synch with product catalog, let the product search api do the work when the collection contents are requested.

For example, a dynamic “Soccer” collection may be defined with the following search criteria: category=“Soccer”, and textSearch=”Soccer”. The Soccer collection will call the product api search interface passing the collection criteria. Content returned from product search will be presented to user as contents of the dynamic “Soccer” collection (content that contains the category “Soccer” or has a name or description that contains the word “soccer”).

Collection Properties

All collections have the following properties:

- identifier, unique identifier for the collection of type String
- name, name of the collection
- description, description of the collection
- list of child collections, list of child collection identifiers

Static collections have:

- list of product ids

Dynamic Collections have:

- product search Criteria, similar to Product API search parameters

Collection Service API

Creating Collections

Support the ability to create new collections.
This is a restricted Interface, requires access token.

Adding Content to Collections

Support the ability to add content to collections.
This is a restricted interface, requires access token.

Searching Collections

The collection service should support the ability to perform a text search on collections. The provided search string should be compared with the collection name and description. Any matching collections should be returned.

Iterating over Collections

The collection service should return an iterator for a given collection. If no collection is specified, then the iterator should iterate over all collections.

The Iterator should perform a depth first iteration. The getCurrent() method used to access the current element may return either a collection element or one of the contained products.

Sample Collection Data:

 # define collections

 # define_collection, <collection_type>, <collection_id>, <collection_name>, <collection_description>

define_collection, static, sports_collection, sports, cool sports apps

define_collection, dynamic, cricket_collection, cricket, cricket apps

 # add content to collections

 # add_collection_content, <collection_id>, <content_type>, <content_id>

add_collection_content, sports_collection, product, Yahoo!_Sports

add_collection_content, sports_collection, product, Score_Center

add_collection_content, sports_collection, collection, cricket_collection

 # set search criteria for dynamic cricket_collection, dynamic collection criteria are specified as with product api search criteria

 # set_dynamic_criteria, <collection_id>, <category list>, <text search>, <minimum rating>, <max price>, <language list>, <country code>, <device id>, <content type list>

set_dynamic_criteria, cricket_collection, , Cricket, , , , , ,