11-2004

# Intelligent, Pervasive and Context-Aware Promotion Delivery Framework for Wireless Mobile Devices

Karthik Harihar
*Columbus State University*

## Recommended Citation

Harihar, Karthik, "Intelligent, Pervasive and Context-Aware Promotion Delivery Framework for Wireless Mobile Devices" (2004). *Theses and Dissertations*. 57.
https://csuepress.columbusstate.edu/theses_dissertations/57

# INTELLIGENT, PERVASIVE AND CONTEXT-AWARE PROMOTION DELIVERY FRAMEWORK FOR WIRELESS MOBILE DEVICES

Karthik Harihar

Columbus State University

The College of Science

The Graduate Program in Applied Computer Science

# Intelligent, Pervasive and Context-Aware Promotion Delivery Framework for Wireless Mobile Devices

A Thesis in

Applied Computer Science

by

Karthik Harihar

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

November 2004

I have submitted this thesis in partial fulfillment of the requirements for the degree of Master of Science.

11/04/04
Date

Karthik Harihar

We approve the thesis of Karthik Harihar as presented here.

3/14/05
Date

Stan
Dr. Stan Kurkovsky, Associate Professor
of Computer Science, Thesis Advisor

3/14/05
Date

Dr. Bhagyavati, Assistant Professor
of Computer Science

03/14/05
Date

V. Zanev
Dr. Vladimir Zanev, Professor
of Computer Science

# Abstract

Mobile devices are now powerful systems and are the main vehicles for m-commerce. Market research predicts enormous potential for m-commerce applications and services [9],[10],[11],[12]. Unique features such as portability, location-awareness and personalization make mobile devices effective tools for targeted advertising. Several mobile marketing frameworks have been designed that deliver promotions based on location information and user preferences [3],[4],[5]. These frameworks have several drawbacks that need to be addressed in order to be effective. The aim of this thesis is to research a simple and efficient framework for targeted and personalized advertising.

This work presents SMMART, a context-aware, adaptive and personalized m-commerce application designed to deliver targeted promotions to the users of mobile devices. SMMART delivers personalized promotional information based on context and the preferences of the user by matching the user's shopping interests to current promotions available at a retail store. SMMART analyzes the user's shopping habits and dynamically adapts to the changing interests of its user. SMMART does not reveal any private information about the user to the stores and does not require additional devices such as a GPS Receiver or a Bluetooth Sensor in order to work effectively. SMMART uses XML web services for client/server communication over a Wi-Fi wireless network.

This work includes a brief study of several mobile marketing frameworks and a description of the SMMART framework architecture. We then describe a fully functional prototype of SMMART built for Pocket PCs running Windows CE with .NET Compact

Framework that uses XML Web Services for communication with a SMMART Server. We also present a simulation model to justify the economic feasibility of SMMART [1].

The purpose of this thesis is to research and develop a novel and efficient approach to mobile marketing and to illustrate our work through a fully functional prototype and a justification of the economic feasibility of the framework. This thesis also provides possible directions for future research.

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgements

*"At times our own light goes out and is rekindled by a spark from another person. Each of us has cause to think with deep gratitude of those who have lighted the flame within us" - Albert Schweitzer.*

This thesis would not have been possible if not for the encouragement and guidance from several members and I feel truly honored to acknowledge their efforts.

Working under the supervision of ***Dr. Stanislav Kurkovsky***, my Thesis Advisor and Associate Professor, Department of Computer Science has been a privilege and a real learning experience for me. He has been the guiding light behind this work and I would like to express my heartfelt gratitude for his invaluable insights and encouragement during the course of my research.

I feel proud to have such wonderful parents and fiancé, who believe in me and are my source of inspiration. It gives me real pleasure thanking them for their support and conviction.

I would like to thank ***Dr. Wayne Summers***, Professor and Distinguished Chair Person, Department of Computer Science for his encouragement and guidance.

I am extremely thankful to my thesis committee members ***Dr. Bhagyavati***, Assistant Professor, Department of Computer Science and ***Dr. Vladimir Zanev***, Professor, Department of Computer Science for agreeing to be on my thesis committee and for their valuable suggestions and inputs.

I would like to thank ***John Hudson***, former undergraduate student, Department of Computer Science, for assisting me with the development of a portion of the SMMART client as part of his requirement for the **Research Experience for Undergraduate (REU)** program during summer 2004.

I would also like to thank all the faculty members and staff of the Department of Computer Science for sharing their invaluable knowledge and also for their cooperation and support.

# Dedication

To my wonderful parents and my lovely fiancée

# I. Introduction

Over the last few years, electronic commerce has made significant progress and applications that have been developed for this purpose are geared towards wired infrastructure with the assumption that the users are stationary. Advances in the areas of wireless and mobile networks have been phenomenal over the last few years and have found widespread user acceptance. Mobile devices have become more powerful than ever in terms of processing and storage capabilities. Mobile devices have contributed to a new way of doing business termed m-commerce, which offers users the luxury of mobility and portability. Market Research results indicate an exciting market for wireless applications and services [3], [4], [5]. A new class of m-commerce applications is possible due to their unique features such as mobility, personalization and location-awareness [14]. However, along with these advantages, mobile devices impose certain constraints due to their smaller display size, limited graphics support, restricted computational capability and short battery life. There is also the issue of limited bandwidth of wireless networks. These issues require careful consideration during the design of applications for mobile devices.

Mobile advertising applications offer the convenience of personalization wherein promotional information can be delivered to the consumers based on their preferences and location. Several frameworks have been designed that are aimed in this direction. While these are novel approaches to mobile marketing, most of them lack a personal touch and require special devices such as location sensors or Bluetooth support in order to work effectively.

This thesis describes the SMMART framework – a System for Mobile Marketing: Adaptive, PeRsonalized and Targeted. SMMART offers personalized and narrowly targeted advertisements to the users of mobile devices such as a PDA based on the context and the preferences of the user. SMMART uses the preference information of the user to retrieve offers when they are in close proximity or inside a participating retail store offering promotions and does so without revealing the user's personal information to the stores. SMMART also adapts to the user's browsing patterns on-the-fly by inconspicuously monitoring the shopping habits and learning the personal preferences of the user.

The rest of this thesis is organized as follows: Section II describes the motivation for conducting the research; Section III describes the related work and provides a comparison of SMMART with several related frameworks; Section IV describes SMMART framework in detail with a typical scenario of using SMMART, its architecture and working details, additional tools to extend its basic functionality, SMMART implementation details and a simulation model to justify the economic feasibility of using the SMMART framework . This thesis is concluded with a brief summary along with the possible directions of future work presented in Section V. Also included in this thesis is the list of references and an appendix that contains the discussion of a tool to administer the Inventory database of the store, the database schema, the use cases for designing the SMMART client along with the scripts and stored procedures for creating and accessing the Inventory database.

# II. Motivation

Rapid advances in technology, principally in the area of processing power, storage and wireless computer networks has made present mobile devices such as wireless personal digital assistants (PDA) and internet-enabled wireless phones to perform as effectively as lower-end laptops of yesteryears. Several market research studies predict an enormous increase in the usage of mobile devices and wireless data services [9],[10],[11],[12]. Strategy Analysts [9] estimate a penetration rate of 32% in the worldwide mobile user base, which is an increase from 1.2 billion at the end of 2003 to 2.1 billion by the end of 2009. It also forecasts consumer expenditure of around US$ 112 billion on wireless data services by 2007. According to the estimates by the Yankee Group [10], wireless users are expected to exceed 1.75 billion, thereby growing around 9 percent from 2002 to 2007.

The proliferated usage of mobile devices has led to a budding discipline termed M-commerce. E-commerce involves conducting transactions and business via wired computer networks. M-commerce is not E-commerce, in the classical sense [16], but is a special branch of E-commerce in which mobile applications transact through wireless networks. Though it is possible to alter the existing E-commerce applications into wireless E-commerce or M-commerce applications, wireless medium can also account for new kind of applications that are possible due to its infrastructure [7]. We will look into some of the unique characteristics of wireless devices in the next section.

## II.A. Characteristics of Wireless Mobile Devices

Wireless mobile devices have certain special characteristics that make them suitable for creating ubiquitously interactive applications than a conventional desktop PC or Web-TV and have several implications on marketing [14]. They are characterized by the following features:

- Portable: As mobile devices are generally small in size, they can be carried by the user at all times and are thus available to the user at any time. This characteristic of mobile devices can be used to create ubiquitous computing applications that can be accessed by the user at all times. Also, this feature makes mobile devices convenient over desktop applications and has contributed immensely to their popularity.
- Personal: Mobile devices are usually personal devices associated with a single individual. Based on the usage of a mobile device, it is possible to track an individual using it and provide services based on the individual's personal interests.
- Location – aware: Wireless devices have the know-how to obtain the absolute physical geographic location and can be used for providing routing, mapping and related services to the user. Several technologies can be used to determine the location of a mobile device and its user. Mandated by the U.S. government, by the end of 2005, all mobile phones must be able to support E911 requirements so that they can be pinpointed within a small radius. Most CDMA phones will be equipped with Assisted or Differential Global Positioning System technology (A-

GPS and D-GPS), while GSM networks will be upgraded with Uplink Time Difference of Arrival technology (U-TDOA) [1].

These characteristics allow development of applications that can deliver dynamic content specifically tailored and targeted to a particular user. However, the diminutive nature of mobile devices imposes several constraints that makes wireless advertising (discussed in section II.C) challenging and exciting.

## II.B. Problems with Conventional Advertising

In the wake of stiff competition, businesses come up with various forms of advertising in order to attract customers, increase their revenue and consolidate their customer base. Advertisements can be seen in virtually all types of media, such as TV, radio, magazines, newspaper, e-mail messages and web portals. Though these means of advertising are able to attract customers to a certain extent, the conventional forms often lack a personal touch, which is critical for effective advertising. For example, retail stores usually distribute paper fliers that contain promotional information in order to advertise their current offers to the consumers. Though helpful, they include information that is common to all consumers and might contain extra, unwanted information for a particular consumer who might be in the store looking for offers on some products of his/her choice. Unless the consumer looks into all the details, the possibility of the customer overlooking products of interest cannot be ruled out. Also, upon viewing all the promotions the likelihood that the customer will end up with his/her product(s) of interest is remote, resulting in loss of precious time. Eventually, such retail stores may lose potential sales. Thus, the conventional forms of advertising are unable to offer promotions based on the requirements of particular individuals [4]. This is in sharp contrast to advertising on mobile wireless devices, which offer the potential for personalized promotions.

## II.C. Wireless Advertising

Wireless services and applications hold a tremendous promise for growth and revenue. Market research by Ovum [11] indicates that the wireless advertising market will generate $16.4 billion in global expenditures by 2005, which represents 20 percent of all the spending on the Internet. Kelsey Group [12] predicts that the wireless-advertising revenues would reach around $17 billion in 2005. It also predicts that in the U.S, the revenue from wireless advertising and promotions will reach $3.9 billion in 2005. These estimates are surely an indicator of the potential market for wireless applications and services. Though the potential for wireless advertising looks promising, there are several challenges that need to be met in order to succeed.

Wireless advertising needs a well-developed wireless network infrastructure, newer marketing ideas and supporting technologies [15]. Several factors, such as the mode of delivery (push/pull), data access support (audio/visual), support in the interface design for limited attention of the user who is often distracted by other events and objects, as well as the physical limitations of the device such as smaller display size, limited power and memory, limited bandwidth of the wireless infrastructure and privacy issues need to be considered during the design and development of a wireless advertising platform. The characteristics of wireless devices allow the development of new applications that are not just an adaptation of an existing e-commerce application.

Varshney [7] states mobile advertising as an important class of m-commerce applications as they provide the potential for targeted advertising. The characteristics of wireless devices such as portability, individual usage, mobility and location-awareness (section 0) make them well-suited for delivering targeted advertisements. Several wireless carriers such as Vodafone and AT&T already send out advertisements in the form of Short Message Service (SMS) messages to the mobile users using the "opt-in" approach wherein the user's permission is solicited before pushing the advertisements [13] .

Mobile marketing applications can be created that offer the following features:

- Personalization;
- Location- or context- awareness;
- Adaptiveness; and
- Customization.

One of the important characteristic of mobile devices is the issue of context-awareness. Dey [2] defines context as any information that can be used to describe the situation of a person or place or object considered pertinent to the interaction between a user and an application. Using situational information or context by an application provides it with the ability to offer relevant information to the user. Context-aware services do not require exact geographic coordinates as mandated by location-based services. Context-aware services are enabled when a service provider and a client (in client/server architectures) or two or more peers (in peer-to-peer architectures) come into a physical proximity of each other. However, context-aware applications are limited in their transmission range. Thus, applications using context can provide services to only those users located within a close physical proximity of the wireless service provider.

Mobile advertising frameworks which make use of context to provide user specific content [3],[4],[5] are discussed in the next section.

The preceding few paragraphs indicate the potential, challenges and scope for wireless services and mobile advertising in particular, which are the motivating factors for our research i.e. to design and develop a unique, simple and efficient approach to mobile advertising that targets and adapts itself to the user's interests. We conducted a comprehensive survey of the existing mobile marketing frameworks in order to learn from their successes as well as their limitations.

# III. Related Products

Several intelligent mobile marketing frameworks have been proposed and prototypes have been developed as proof of concept. We will examine a few and look into their positive aspects as well as their drawbacks. We then propose a framework for mobile advertising that provides the consumers in a retail store with targeted promotions based on the context of their location and precise matching to their preferences. Hence it safeguards the consumers from receiving any unwanted information (spam). It also learns about the user's interests unobtrusively and dynamically and does not mandate location or Bluetooth sensors as mandated by some of the frameworks mentioned below. Our framework ensures user privacy as it does not communicate personally identifiable information with the participating retail stores.

## III.A. eNcentive

eNcentive framework [3] is an agent based, peer-to-peer marketing framework for mobile environments built using Java technology. eNcentive employs an intelligent marketing format wherein users can collect as well as distribute sales information in the form of electronic coupons. The users are rewarded by the businesses depending on their distribution rates whereas the businesses attract more customers in this process, thereby being mutually beneficial. Though eNcentive can be used as an effective marketing tool, it has several drawbacks. In particular, eNcentive

- Employs a 'push' model wherein all available sales promotions are forced onto its users regardless of their preferences;
- Provides no means for learning user preferences over time;
- Increases computational load on the resource constrained mobile devices as they interact with each other and also listen for broadcasted messages quite often; and
- Requires a large number of participating customers to be more effective.

## III.B. Ad-me (Advertising for the Mobile E-Commerce user)

Ad-me [4] is a context-aware and personalized advertising system that is built on top of a mobile tourist guide. Ad-me uses user location as well as their profiles to deliver personalized advertisements that are less obtrusive. It also updates user profiles dynamically depending on user migration and activity. Ad-me is implemented using Java technology and uses a mix of both push and pull technologies in order to fulfill its objective. Ad-me has the ability to deliver multimedia presentations as well.

Ad-me has a host of the above-mentioned features that make it useful for context-sensitive mobile advertising. However, Ad-me has the following drawbacks:

- Needs a GPS receiver in order to obtain the geographical information about the location of each user;
- Due to the nature of its design, huge volumes of data are passed between the Client and the Server frequently and thus tends to slow down the system. For example, though Ad-me updates user profile information dynamically, since it stores user information on the Server, depending on the user activity data is sent often between the Client and the Server; and
- As it uses maps in its display, it makes the client slower due to the limited resources of the PDA.

## III.C. B-MAD (Bluetooth Mobile Advertising)

B-MAD [5] is a permission-based location – aware mobile advertising system that uses Bluetooth technology for positioning and Wireless Application Protocol (WAP) push for delivering advertisements. B-MAD uses permission-based advertising wherein the Server sends in only those advertisements that is associated with the location of the client and which were not sent previously to the client. Advertisements are delivered as WAP messages. Shortcomings of B-MAD include:

- Framework requires Bluetooth sensors. Also, GPRS capable end user devices with XHTML browser are required. These requirements are quite hard to meet, as most devices at present are not equipped to support all the above requirements; and
- Though B-MAD sends in promotions that pertain only to the location of the user, it may still contain offers that are of no interest to the user.

## III.D. Comparision with intended features of SMMART

SMMART framework has several features that make it a useful tool for targeted and personalized mobile advertising. The following table compares the features of the above mentioned frameworks with the intended feature set of the SMMART framework. This is done in order to present SMMART frameworks' effectiveness and suitability for delivering personalized promotions over other frameworks.

| | SMMART | eNcentive | Ad-me | B-MAD |
|---|---|---|---|---|
| **Advertising Mode** | Pull | Push | Push & Pull | Push |
| **Context-aware** | Yes | Yes | Yes | Yes |
| **Personalization** | Yes | No | Yes | No |
| **Customization** | Yes | No | Yes | No |
| **Targeted Delivery** | Yes | No | Yes | No |
| **User profile updating** | Dynamic | No | Dynamic | No |
| **Adaptiveness** | Yes | No | Yes | No |
| **Special Equipment Needs** | No | No | GPS Receiver | Support for Bluetooth |

**Table 1: Comparison of several mobile advertising framework features.**

In this section, we have briefly reviewed several mobile marketing frameworks along with the discussion of their drawbacks and a comparison of the intended features of SMMART with those offered by these frameworks. We discuss our framework in detail in the following section.

# IV. SMMART Framework

The applications mentioned in the previous section were developed for mobile marketing and offer newer ways of promoting products sold by retailers. But, as Table 1 points out, these applications have several drawbacks that limit their usage.

- Some of them 'push' information to the users, which might annoy the users.
- The promotions are not customized to the user's needs. The users have to go through all the products irrespective of their liking in order to find the product(s) of their choice.
- Most of the frameworks do not learn and adapt to the user shopping behavior.
- Some of them require special hardware features such as a Bluetooth Adapter or a Global Positioning System in order to facilitate their usage. This increases the cost of usage.

With these things in mind, we developed a pervasive framework (SMMART) aimed at providing the users with a productive shopping experience. SMMART stands for System for Mobile Marketing: Adaptive, peRsonalized and Targeted. SMMART distributes targeted marketing information to those customers whose preferences match the products that are currently on sale at retail stores. SMMART monitors and analyzes user browsing patterns to display relevant offers on the fly. SMMART has the following features.

- **Adaptive:** SMMART adapts according to the user's browsing patterns and displays relevant information dynamically reflecting their choice.

- **Personalized:** SMMART uses user preference information to display only their choice of offers.

- **Targeted:** SMMART displays promotions that are specifically targeted towards the user. Thus, the user is spared from viewing all advertisements.

- **Context – aware:** SMMART uses context information of the user location along with user preferences to provide the users with a rewarding shopping experience.

- **Customizable:** SMMART allows users to customize their preferences at any time and generates relevant information on-the-fly.

- **Confidential:** SMMART does not divulge the consumer's private information to the stores. As it does not store or communicate personal information of the user such as his name, address, e-mail etc to the stores, the user is spared from being bombarded with unsolicited advertisements (spam).

- **Intelligent**: SMMART learns from the user's behaviors. This helps in displaying more customized information.

- **Efficient:** The SMMART framework has been designed to facilitate efficiency in performance by appropriate distribution of the load on both the client and server. The SMMART Client has been designed to be thin in order to accommodate for the low processing and storage capabilities of wireless mobile devices whereas the SMMART Server handles most of the processing, thus relieving the client of the load. .

## IV.A. Typical Scenario of using SMMART

Simon decides to visit a few stores to look for any good deals on musical albums and movies of his choice. As Simon is interested in musical albums by *U2*, *Sting* and *Oasis* and movies by *Steven Spielberg*, he enters these as his preferences in the SMMART Client installed on his PDA as illustrated in Figure 1. When he enters "Top Buy", the SMMART Client connects to the store's network and informs the SMMART Server inside the store network about Simon's preferences. The Server then returns a list of products with offers that matched his preferences. Thus, Achtung Baby, an album by *U2* and Ten Summoner's Tales by *Sting* are displayed on his PDA. Simon finds the SMMART Client very handy as he did not have to browse through the store's entire flier containing all promotions. He decides to explore Ten Summoner's Tales in detail and clicks on that product on his PDA screen. The SMMART Client makes a note of Simon's interest in this product and adds the keyword *Ten Summoner's Tales* to his list of preferences while also noticing his continued interest in *Sting* by increasing its weight. Along with the product and offer details, products that are related to *Ten Summoner's Tales* are displayed on the screen. Simon decides to find out more about Every Breath you Take: The Classics album by Police that also features *Sting*. Simon's interest in this album is noted by SMMART client by adding keyword *The Police* to his list of preferences. The weight of the keyword *Sting* is also increased by the SMMART Client.

Simon then visits Full-Mart to look for some more offers that would fit his interests. He remembers his son Brian's fascination with the animation movie Shrek and decides to surprise him with a gift. He adds the keyword *Shrek* to his list of preferences.

As Full-Mart does not have an offer currently on *Shrek*, it is not displayed on his screen. But, Minority Report, a movie by Steven Spielberg, Lock, Stock and Two Smoking Barrels, a movie by Guy Ritchie featuring Sting and Snatch, a movie with music by Noel Gallagher, founder of *Oasis* match his preferences. Simon views the offer on Snatch and decides to purchase it as he had not seen it before and also due to the fact that Noel Gallagher has scored the music for the movie. The keywords *Snatch* and *Noel Gallagher* are added to his preferences while the weight of *Oasis* is increased.

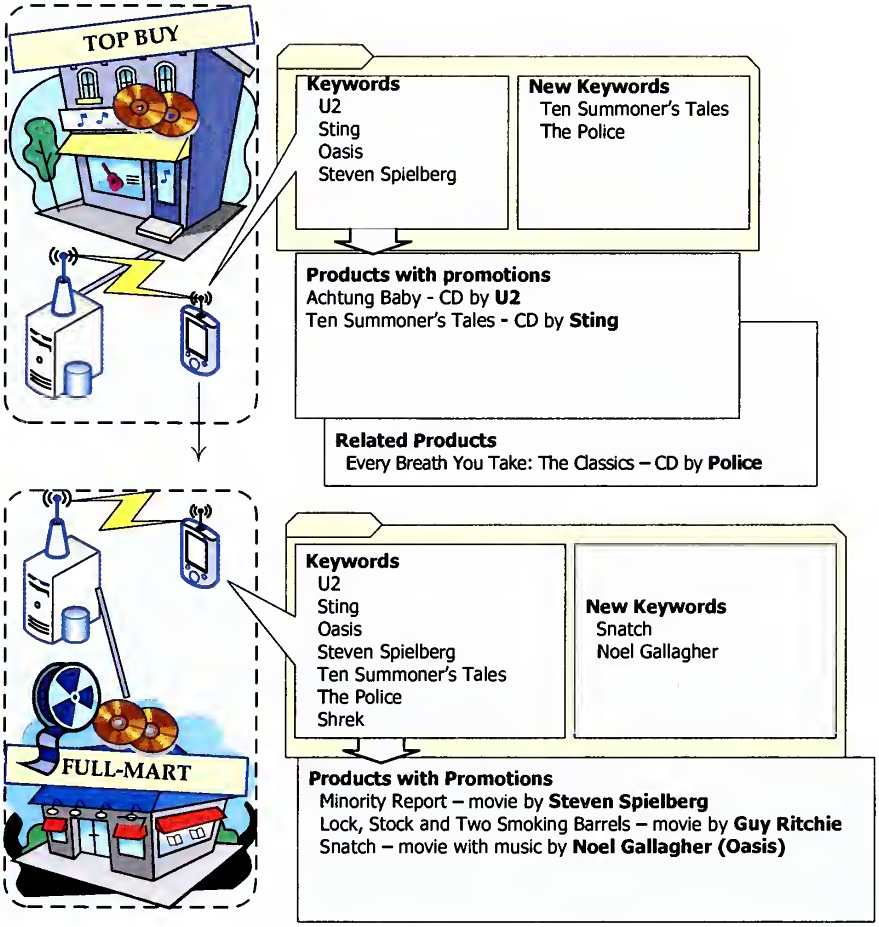


**Figure 1: Scenario depicting typical usage of SMMART**

Simon decides to buy *Shrek* irrespective of the offers and decides to search for it in the store. He enters the keyword in the SMMART Client Search screen. The search result informs him of its presence in the store. As he goes through its details, he also decides to purchase Finding Nemo, which was on the list of related products to *Shrek.* The SMMART Client adds the corresponding keyword, *Finding Nemo*, to his list of preferences and also confirms his interest in *Shrek* by increasing its weight. Simon is pleased as he found excellent promotions on his choice of products. The stores Top Buy and Full Mart increased their sales revenue and also satisfied their customer in the process.

## IV.B. SMMART Framework Architecture

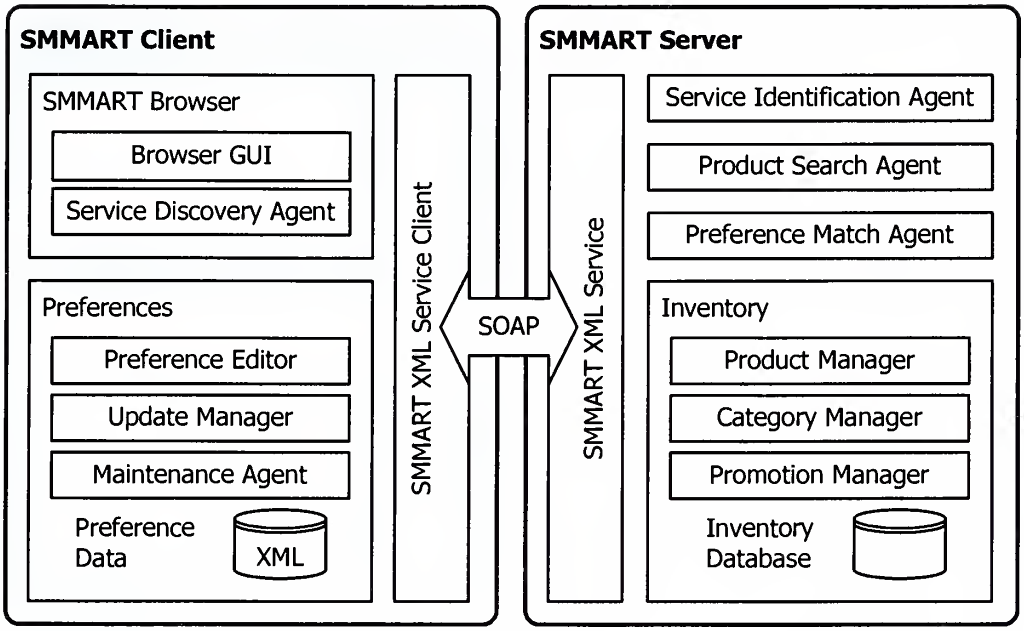


**Figure 2: Architecture of the SMMART Framework**

Figure 1 shows the architecture of a SMMART framework. The framework is composed of two main modules, a SMMART Server that is installed at every participating store location and a SMMART Client that resides on the user's mobile device (typically a PDA) equipped with wireless network connectivity. The SMMART Server hosts an XML web service that handles the data. An Inventory database holds the necessary information such as the details of the products (name, category, price, related products etc) along with the details of the offers associated with a particular product. Before discussing the internal details of the modules, let us consider how the store server is configured for wireless access.

### IV.B.1. Configuration

In order to implement a SMMART framework, each participating store is expected to strictly enforce the following configuration.

- Each store should have its own internal wireless network. The network may be formed by using either a single or multiple wireless access points.
- The access points should be configured in such a way that they route traffic only to the store's server hosting SMMART XML web service. This can be done by setting up the routing parameters within the access points to point only to the SMMART Server.

This approach has many advantages from the store's point of view.

- Stops the users from using the store's network bandwidth for their personal use such as browsing the internet. Thus, users are limited to using the store's network for running only the SMMART Client application.
- Facilitates the automatic service discovery process for the SMMART clients.

The SMMART Server can be typically configured in a single network mode or a multiple network mode as shown in **Error! Reference source not found.**. In a single network mode, the inventory database and the SMMART web service are located on the same network. In a multiple network mode, a chain of participating stores having their own wireless networks connect to a single, centralized inventory database. Note that each store hosts the SMMART web service on its internal network.

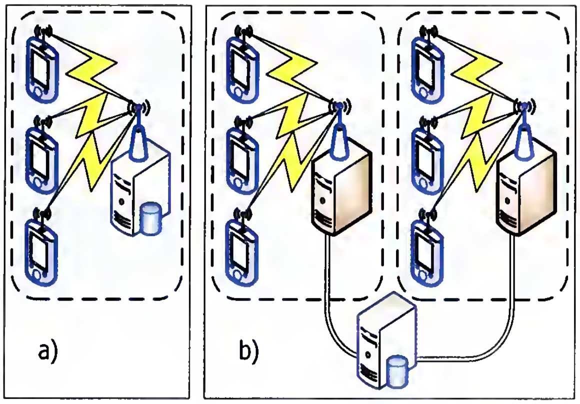


**Figure 3: a) Single Network Mode b) Multiple Network Mode**

Let us now consider the internal details of each main module.

### IV.B.2. SMMART Server

The SMMART Server is responsible for handling the requests from the client and sends the appropriate information back to the client. There are three modules making up the SMMART Service:

a. Service Identification Agent Module;

b. Preference Match Agent Module;

b. Product Search Agent Module; and

c. Inventory Module.

The Service Identification Agent aids the Service Discovery Agent (SDA) of the SMMART Client in the identification of the appropriate store service. When it receives a request for the store network name from the SDA, it responds with the appropriate data that informs the SDA of the current store network. This module is extremely important for the implementation of the automatic service discovery process. The data returned by the SDA to the Client is used as the basis for further communication between the client and the server.

The Preference Match Agent is the heart of the SMMART Server and has built-in intelligence to scan and return customized data to the client. The Agent accepts the keywords sent by the client and checks for constraints, and works in close conjunction with the Inventory module to perform an intelligent scan of the data only for products with promotions that match the client specifications, as illustrated in Figure 4. As more than one keyword can match the same product, there is every possibility that duplicate products can be entered into the results that are sent by the SMMART Server to the client. The Preference Match Agent eliminates this possibility by checking for duplicates before adding the data to the result set. The data is also organized by the Agent in the order of relevance to the client preferences and returned back to the client. This makes it easier for the client to display the information that is related to a preference with higher priority on the top than those with lower ones.

```
function MatchPreferences
  in Preferences : ordered list of preferences
  out Result : ordered list of products
begin
  Kword : keyword
  for each Kword in Preferences do
    NewResult : list of products
    NewResult = SearchInventory(Kword,Ø,Ø,Ø)
    NewProduct : single product from the inventory
    for each NewProduct in NewResult do
      if (NewProduct not in Result) and
         (NewProduct has a current promotion) then
        add NewProduct to end of Result
      end if
    next for
  next for
  return Result

end function
```

**Figure 4: Algorithm for matching user preferences against current offers and products in the store's inventory.**

*Product Search Agent* has two primary functions: it assists the *Preference Match Agent* in finding products in the inventory that match individual keywords and it also serves as a mechanism for querying the store inventory using the Search and Advanced search features of the SMMART Client, whose interface was described in the previous section. Details of the inventory search algorithm used by the *Product Search Agent* are illustrated in Figure 5. The two search features differ in the depth and the flexibility of search. While the basic search allows the user to query the store inventory by just entering the search keywords, it gives the user less control over the search results. The advanced search feature provides complex querying capabilities and offers a better control over the search results.

```
function SearchInventory
  in ANDKeywordList : product description must contain ALL these keywords
  in EXACTPhrase : product description must contain this EXACT phrase
  in ORKeywordList : product description must contain AT LEAST ONE of them
  in NOTKeywordList : product description must NOT contain any of them
  out Result : ordered list of products
begin
  tmpResult : temporary list of products
  Result = Ø
  if ANDKeywordList not empty then
    tmpResult = all products with ALL keywords from ANDKeywordList
    Result = tmpResult
  end if
  if EXACTPhrase not empty then
    tmpResult = all products with EXACTPhrase in the description
    Result = products both in Result and in tmpResult
  end if
  if ORKeywordList not empty then
    tmpResult = all products with AT LEAST ONE keyword from ORKeywordList
    Result = products both in Result and in tmpResult
  end if
  if NOTKeywordList not empty then
    tmpResult = all products with AT LEAST ONE keyword from NOTKeywordList
    Result = Result without products both in Result and in tmpResult
  end if
  return Result
end function
```

**Figure 5: Algorithm for inventory search based on a combination of keywords**

In the Basic search feature, the user provides one or more search keywords, which are then sent to the server. The *Product Search Agent* in the Server performs a logical AND operation on the keywords and searches the entire inventory for products that match all specified keywords. The Advanced Search feature has more options for searching and provides the user with more control over the search process. The user can use any combination of the following options, which are processed by the *Product Search Agent* to produce the desired results as follows.

**All words:** The search is carried out by performing the logical AND operation on all specified keywords. Only those products that have all the specified keywords are returned to the client. When used alone, this option is identical to the basic search feature.

**Exact phrase:** The search results yield only those products whose descriptions contain a match to the entered phrase.

**At least one word:** These keywords undergo a logical OR operation and then used to search the inventory. The results of this search are the products that contain at least one of the specified keywords. This option yields the most products in comparison to the other options when used alone.

**Without:** This is a logical OR operation on all keywords followed by a logical NOT operation on the result of the OR. Only those products that do not have the specified keywords are displayed to the user.

The details returned to the client from the SMMART Server contain information relevant only to the preferences communicated to it by the client. Thus, data returned to several clients are not necessarily the same and is dependent on their choice of preferences. It is thus customized according to the requirements of the client. The keywords sent by the client maybe either the preferences entered by the user or the system or a search keyword entered by the user. SMMART ensures user privacy as it does not exchange any personally identifiable information about its users with the retail stores. As SMMART Client communicates only user-specified keywords to the SMMART Server, the server has no way to know about the identity of the user of the client that sent the request. This prevents the stores from spamming their customers and also selling their information to third parties.

The Inventory Module is built on top of the Inventory Database that holds the available data in every participating store. The Inventory Database contains complete information about products with and without promotions. Inventory module, as the name suggests, handles the inventory aspects of the SMMART Server module. It has three components that implement its functional details namely Product Manager, Category Manager and Promotion Manager.

The Product Manager is responsible for extracting the required data about a particular product. This is used by the SMMART Client when the user chooses to view the details about a particular product. Upon receiving a request for information about a product, the Product Manager supplies the required information. Usually, the basic product details along with the relevant offers and related products are supplied to the client.

The Category Manager retrieves the information about the categories from the Inventory database. Depending on the type of request from the SMMART Client, it either returns the category hierarchy for a product belonging to a category or the sub-categories / products associated with a particular category. The Product Manager works in close conjunction with the Category Manager while retrieving the category information associated with a particular product.

The Promotion Manager retrieves all information about the discount(s) offered on a particular product. This module is closely tied to the Preference Match Agent. Upon receiving the keywords from the SMMART Client, the Preference Match Agent requests information from the Promotion Manager on whether any offers exist on each product in the Inventory. The Promotion Manager extracts the relevant data from the Inventory and returns the promotional information. The Preference Match Agent uses this information to make informed decisions before returning the results to the SMMART Client.

### IV.B.3. SMMART Client

The SMMART Client runs on a PDA and has the following set of features.

- **Light-weight:** The SMMART framework has been designed so that the server handles most of the processing details of the framework. Thus, the client is spared from making database calls directly or complex processing (a *thin* client) and is mainly used for displaying the offer information. This approach makes resource constrained devices such as PDA's or smart phones more efficient with respect to the consumption of processing power, memory and battery.

- **Robust:** SMMART provides mechanisms to handle failure effectively and creates a robust environment for mobile advertising.

- **Ease of use:** The SMMART client has simple screens with useful features that are easy to handle.

- **Compact GUI Screens:** The screens are designed in such a way as to compactly include the necessary details without confusing the user.

- **Automatic Service Discovery Mechanism:** SMMART client discovers any store network when the user is in the range of the Wi-Fi network of the participating store.

- **Automatic Keyword Cleaning Mechanism:** SMMART client has a mechanism for cleaning up any keyword (s) that may be of no interest to the user. This prevents the display of offers based on these keywords to the user.

- **Editable preferences:** The user can add, update or delete preferences at any time during the operation of the SMMART client. This provides more flexibility for the user to express his/her changing interests and view appropriate offers dynamically.

- **Full-inventory search capability:** SMMART client offers the user with the ability to query the store's entire product inventory for products with or without offers.

- **Built-in intelligence to monitor and analyze user browsing habits:** Based on the user browsing behavior, the SMMART client adapts to the user's interests dynamically and displays relevant offers that reflect the user's changing choices.

The SMMART Client is composed of two main components, namely SMMART Browser and Preferences module. The SMMART Client provides the users with a compact, well-defined and easy-to-use display mechanism by means of a SMMART Browser. The SMMART Browser has screens for viewing offers, product details and category information and also to search for products in the store's inventory. Once the Service Discovery Agent discovers and connects to the store's network and the SMMART Client communicates the user's preferences to the store's server, the user is able to view the matching products that are currently on sale. The SMMART Browser allows the user to view the complete product information along with offer details and related products.

The user can edit (add, update, and delete) his Preferences through the Preference Editor. The Preference Data is the storehouse for these preferences. The Update Manager is responsible for intelligent monitoring of user browsing habits. This adds new preferences (keyword, weight and timestamp) to the Preference Data if the keywords associated with a product are not present in Preference Data. Otherwise, it increments the weights and updates the timestamp of the preferences. Details of the algorithm used by the Update Manager are as shown in Figure 6.

```
function UpdateKeywords
  in Keywords : keywords to be added/updated in the user preferences
  in/out Preferences : current preference data
begin
  KeywordEntry : keyword record in the preference data format
                 (includes keyword itself, its weight and date stamp)
  for each Kword in Keywords do
    if Kword in Preferences then
      KeywordEntry.Weight = KeywordEntry.Weight + 1
      KeywordEntry.Date = today
    else
      KeywordEntry.Keyword = Kword
      KeywordEntry.Weight = default weight for a new keyword
      KeywordEntry.Date = today
      Insert KeywordEntry into Preferences
    end if
  next for

end function
```

**Figure 6: Algorithm for updating the Preference Data with new or existing keywords**

The Maintenance Agent contains an automatic keyword cleaning mechanism that runs at application startup and searches for keywords which have not been used for a given number of days and decrements their weights and updates their timestamp. These keywords reflect the user's apathy towards them and they are sent to the Recycle Bin after their weights reach a value of zero. The user can review the contents of the Recycle Bin and restore or purge all or part of its contents. The restored keywords are added to the Preference Data with a default weight and the timestamp reflecting the current date.

## IV.C. Working Details of SMMART Framework

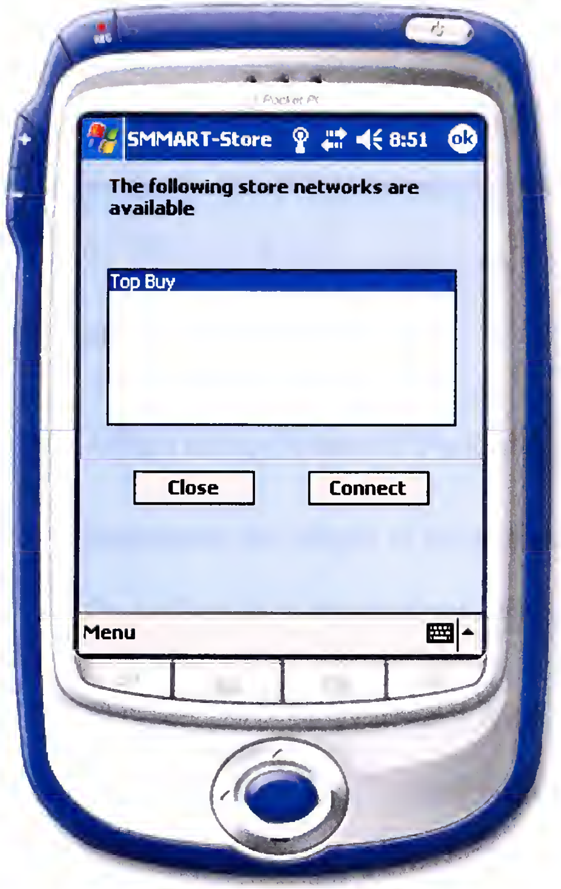
The working details of SMMART have been divided into the following sections:

- Service Discovery;
- Maintenance;
- Browsing; and
- Editing (Preferences).

### IV.C.1. Service Discovery

When the SMMART Client is started, the Service Discovery Agent scans for available store network(s). Upon sensing a network, the SDA looks for an instance of the SMMART web service on a server located in the network. If a SMMART web service is found, the SDA makes contact with the Service Identification Agent on the server and fetches the name of the store network.

This holds true for N available networks. The SDA displays the names of the N discovered networks to the user as shown in **Error! Reference source not found.**. The user opts for their choice of network and the SMMART Client then connects to that network. If there are no available networks, the SDA informs the user about this and gives the user an option either to retry the scanning process or exit the application. If the user decides to retry, the SDA repeats the service discovery process all over again.



The following store networks are available

Top Buy

Figure 7: Screen with available Store Networks

### IV.C.2. Maintenance

Upon connecting to a store's SMMART web service, the Maintenance Agent is invoked and runs in the background. The maintenance Agent is responsible for performing the following operations:

- Removing all the unwanted preferences from the Preference Data; and
- Decreasing the weights of the unused preferences.

### a) Need for maintenance

When a user browses through the products, the Update Manager extracts the keywords associated with the product and checks the Preference Data for the presence of keywords. For each keyword associated with the product, the Update Manager does the following:

- Adds a new preference if the keyword is not present already; and
- Increments the weight of the preference if the keyword is present. The weight of the preference is incremented until a maximum value is reached.

This process is automatic and helps in capturing and analyzing the behavior of the user. The SMMART Client makes an assumption that viewing information about a product indicates the user's interest in the product. This assumption works well in most cases. However, there may be instances where keywords that do not represent the user's real interests are unintentionally added to the Preference Data. For example, the user may click on a product by chance and though the user is not interested in the product, the preference ends up being added in the Preference Data. Subsequently, the promotions (if any) for that preference get displayed along with the list of other promotions. This might turn out to be particularly annoying for the user. Another factor that highlights the need for maintenance is to account for the possibility that the user might lose interest in a product over a period of time. It is important to monitor the behavioral changes of the user in order to ensure that products having offers that are of less interest to the user are displayed later in the order and products that the user is not interested anymore are not displayed.

### b) Maintenance Agent Working Details

```
<?xml version="1.0" encoding="utf-8" ?>
<preferences>
   <keyword name="Sting"       weight="9" date="9/30/2004" />
   <keyword name="Tom Hanks"   weight="6" date="10/02/2004" />
   <keyword name="Guy Ritchie" weight="3" date="7/15/2004" />
</preferences>
```

**Figure 8: Format of the Preference Data**

Figure 8 shows the structure of Preference Data. The data is stored in XML format. Each preference is stored in a <keyword> tag and has the following attributes associated with it.

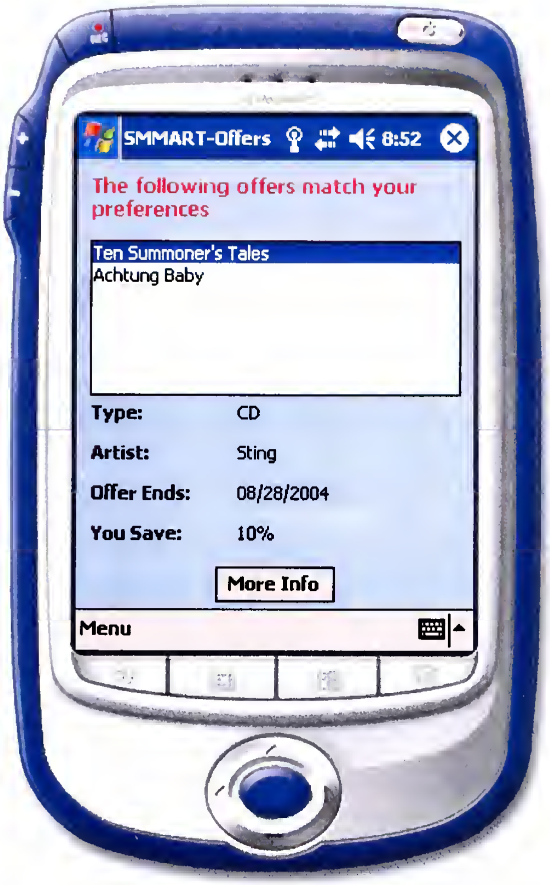
- name: The name of the preference.
- weight: A number that indicates the user's interest in the preference.
- date: A Timestamp that indicates the last time product(s) associated with this preference was viewed.

The Maintenance Agent scans the Preferences Data for preferences whose date attribute has not been updated since d days. (Value of d can be adjusted by the user). The larger the value of d, the lesser the user is interested in the preference and vice versa. The weights of these preferences are decremented by 1. Upon decrementing, if the weight of the preference reaches zero, it is phased out from Preference Data and moved into the Recycle Bin with a default weight.

If the Preference Data is empty, the user is prompted to populate Preference Data with new keywords by using the Preference Editor.  The Preference Editor is discussed later.

### IV.C.3. Browsing

Once the user enters his/her preferences or if they already exist in the Preference Data, they are sent to the server when the SMMART client establishes a connection with a SMMART server. The Preference Match Agent in collaboration with the Product Manager and Category Manager searches the Inventory database for products with offers that match the preferences and sends them back to the Client. The Client then displays the offers in the user's order of interest, which is indicated by the weight of the preference(s). Figure 9 shows the screen with offer information. Along with the name of each product with offers, additional information is provided in order to aid the user in knowing more about the offer(s). Details such as the product type, offer end date and the savings potential (in percentage) are useful indicators for users to make informed decisions.



SMMART-Offers 8:52

The following offers match your preferences

Ten Summoner's Tales
Achtung Baby

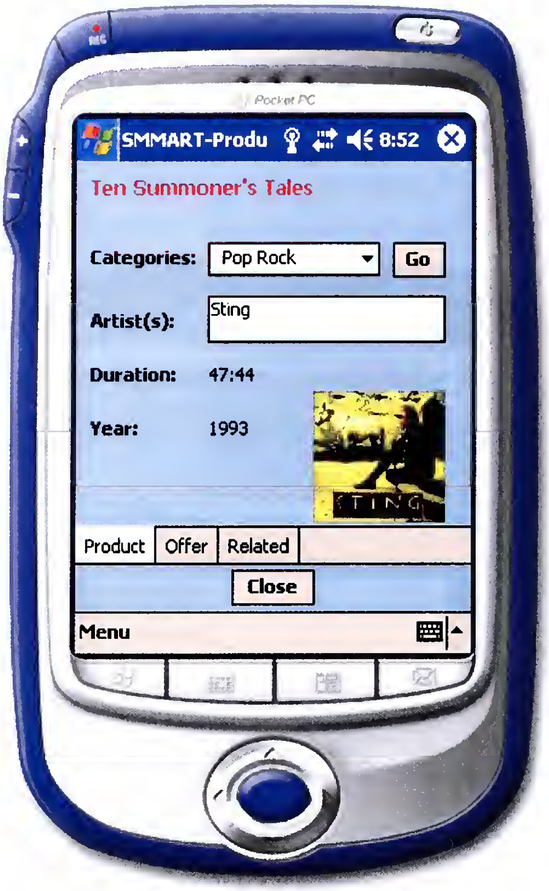
Type: CD

Artist: Sting

Offer Ends: 08/28/2004

You Save: 10%

Figure 9: Offer Screen with available Offers



SMMART-Produ 8:52

Ten Summoner's Tales

Categories: Pop Rock

Artist(s): Sting

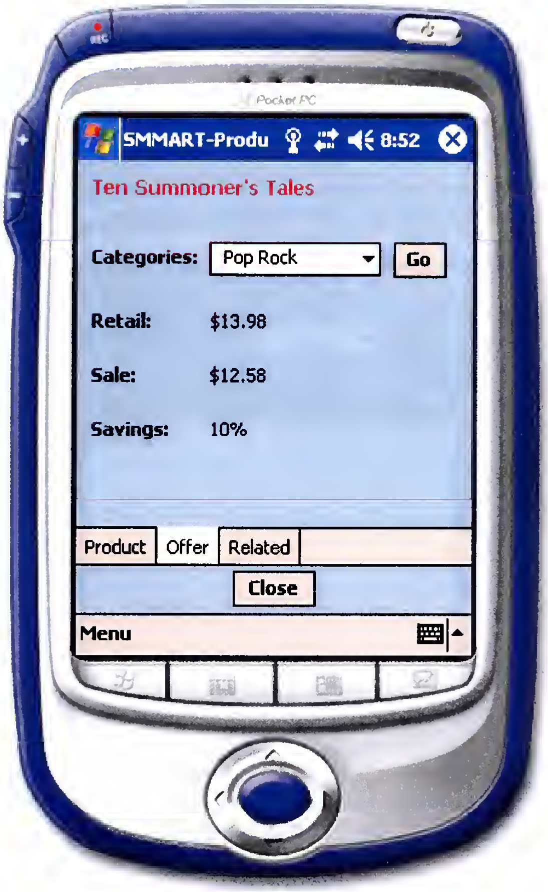
Duration: 47:44

Year: 1993

Figure 10: Product Screen with Basic Product Details

If a user is interested in exploring a product with an offer, the SMMART Client displays a screen packed with lots of useful information about the product. The product screen is as shown in Figure 10. This screen contains details that inform the user about the product such as its name, artists etc. It also contains information about the offer and related products for the product. The product screen has been designed taking into consideration the limited space available for including all the above information in a compact manner.

As can be seen from Figure 7, the screen includes the category information of the product along with the category hierarchy. Other details, such as the artist(s), director (for movies only), year of release and rating (for movies only), are also included.


SMMART-Produ 8:52

Ten Summoner's Tales

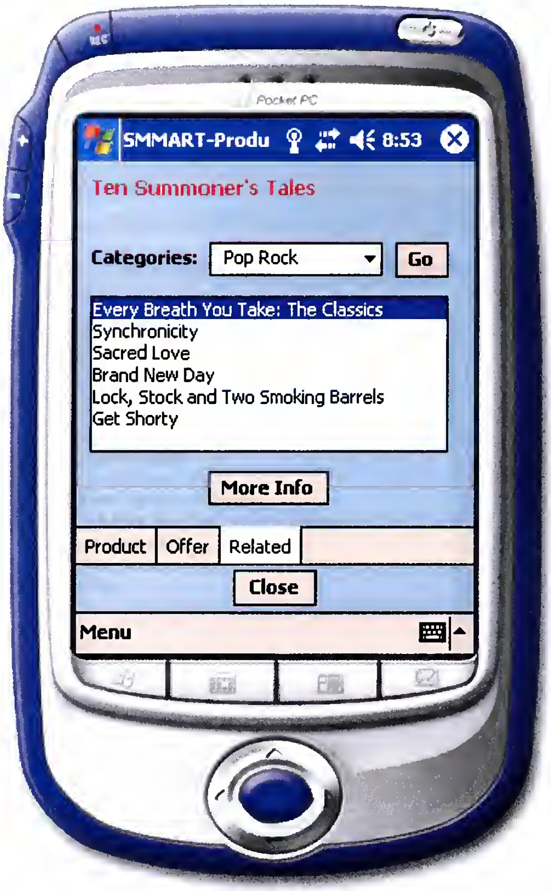
Categories: Pop Rock

Retail: $13.98

Sale: $12.58

Savings: 10%

Figure 11: Product screen with Offer Details


SMMART-Produ 8:53

Ten Summoner's Tales

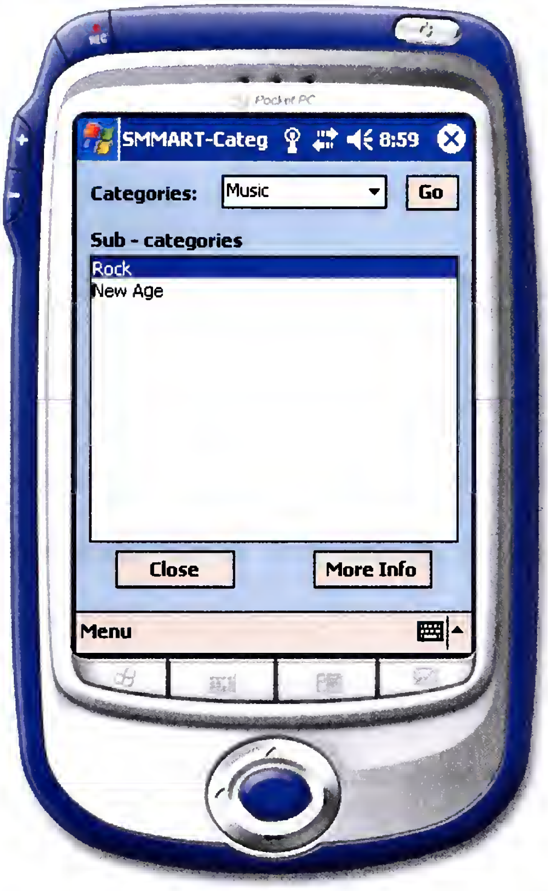
Categories: Pop Rock

Every Breath You Take: The Classics
Synchronicity
Sacred Love
Brand New Day
Lock, Stock and Two Smoking Barrels
Get Shorty

Figure 12: Product screen with Related Products

Figure 11 shows the product screen with offer information while Figure 12 shows the product screen with related product(s). Also note that if the product does not have any offer, the 'Offer' tab is changed to 'Price' and the screen displays the price of the product only. The users can browse their related products of interest.

As mentioned earlier, while the user is browsing the products, the Update Manager is activated. For each product viewed by the user, it compares the keywords associated with the product with the keywords (preferences) in Preference Data and either adds the preferences if they are not found or updates the weights of the preferences already present in the Preference Data. This is done unobtrusively in the background.

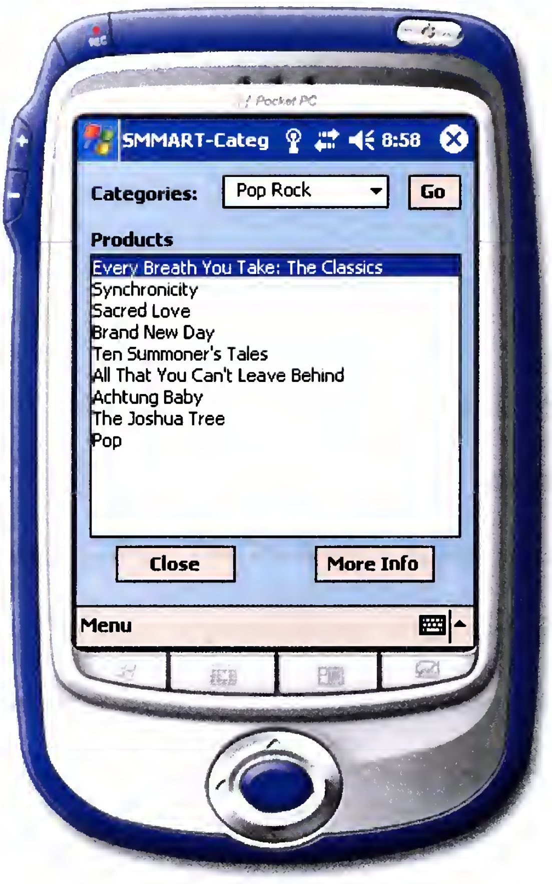


Categories: Music

Sub - categories

- Rock
- New Age

Figure 13: Category Screen with sub-categories



Categories: Pop Rock

Products

- Every Breath You Take: The Classics
- Synchronicity
- Sacred Love
- Brand New Day
- Ten Summoner's Tales
- All That You Can't Leave Behind
- Achtung Baby
- The Joshua Tree
- Pop

Figure 14: Category Screen without sub-categories

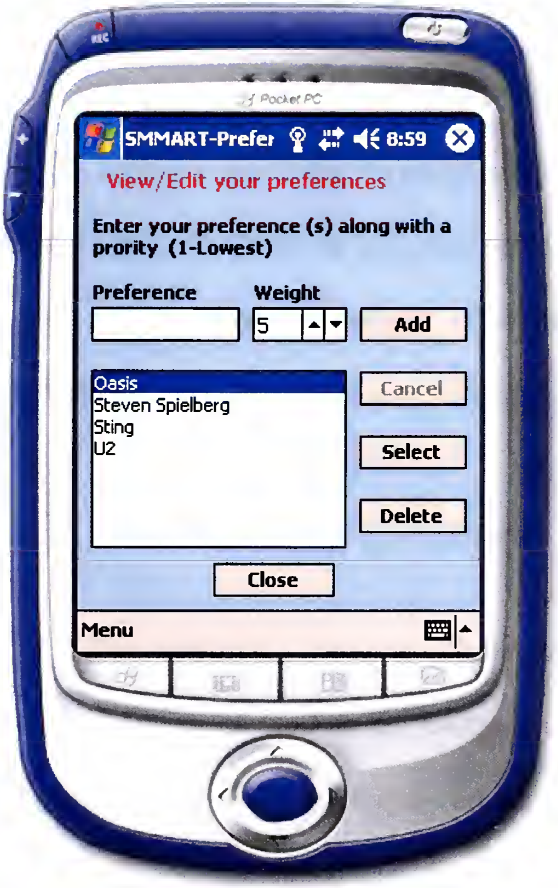
In the product screen, if the user is interested in the category information, the SMMART client displays a screen containing the sub-categories of the selected category. This is as shown in Figure 13. If the category has no sub-categories, products that belong to the selected category are displayed (Figure 14). The user can further traverse the subcategories or view the products associated with the category.

How does a user specify his/her interest in a product? For example, if a user is interested in CD's or Movies by Sting, how to convey this to the application? Also, how does the user notify the application on how interested he/she is in a particular choice? For example, though the user is interested in movies featuring Sting, he/she might be more interested in movies starring Tom Hanks. The SMMART Client has an editing tool named Preference Editor designed and developed for this purpose.

### IV.C.4. Preference Editing

When the user starts the application, if there are no preferences stored in the Preference Data, the SMMART Client gives the user an option of manually adding preferences. This is done through the Preference Editor. The Preference Editor is an interface for manipulating the Preference Data. The main responsibilities of this module include adding new keywords, editing and deleting existing keywords and manually changing the weights of existing keywords if needed. The SMMART Client gives the user the flexibility to alter the preferences at any time during the usage of the application. The offers displayed by the SMMART Client are reflective of the current state of preferences in the Preference Data. Figure 15 shows the Preference Editor screen.

The editor interface allows the users to add, modify or delete the preference(s). When the preferences are added, they are added with a default weight (unless specified otherwise by the user) and the current time. The user can delete unwanted preferences or update either the name or the weight of an existing preference or even both. In addition, when a preference is updated, the timestamp of the preference is replaced by the current time.



View/Edit your preferences

Enter your preference (s) along with a prority (1-Lowest)

Preference

Weight 5

Oasis
Steven Spielberg
Sting
U2

**Figure 15: Preference Editor**

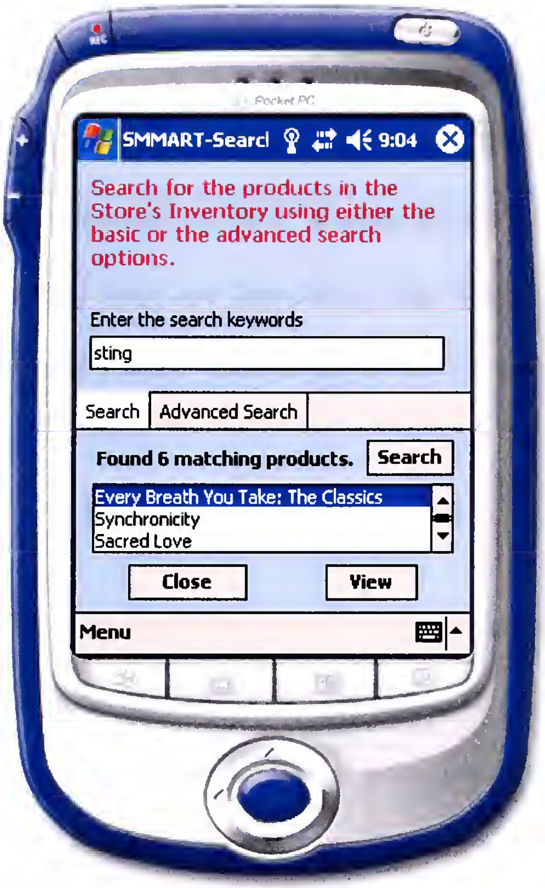
By default, the preferences are sorted alphabetically before display. The sorting of data takes place each time the data in the Preference Data is manipulated. The user can also manually sort the data by weights or keywords (alphabetically). While sorting by weights, the preferences with the same weight are also sorted alphabetically.

## IV.D. SMMART Client Tools

The last few sections described in detail the architecture and the basic functionality of the SMMART Framework. As an extension to the basic functionality of the SMMART Client, few tools have been developed that are meant to supplement the SMMART Client in order to offer more assistance to the users when using the SMMART Client. Tools for searching the entire product inventory (Search), restoring and removing unused keywords (Recycle Bin), and specifying the default settings for the Maintenance Agent and the Update Manager (Default Settings) have been incorporated into the SMMART client. The following sections explain these tools in detail.

### IV.D.1. Search Feature

The SMMART Browser allows the users to explore the different offers, their related products, as well as the category information. They can also view the different products with or without offers by browsing through the Category Browser. However, if the user is looking for a specific product irrespective of the offers, the SMMART Browser offers little help. Though the entire product inventory can be searched through the Category Browser, the user either has to have the knowledge of the category that the product belongs to or make an educated guess. It is thus an arduous task involving trial and errors. SMMART Client has in it a built-in search capability that simplifies the above-mentioned task. It allows the user to search for products in the store's entire product inventory with a simple and easy to use interface



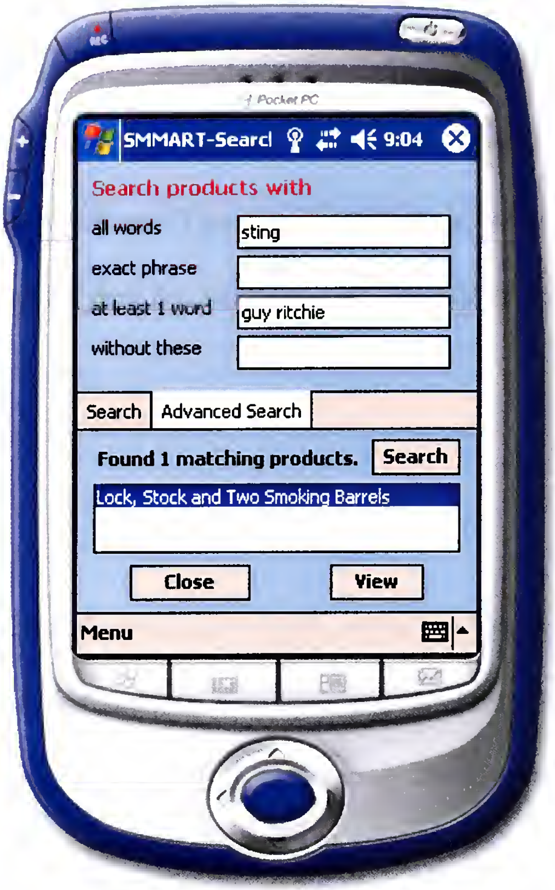
Search for the products in the Store's Inventory using either the basic or the advanced search options.

Enter the search keywords: sting

Found 6 matching products.

Every Breath You Take: The Classics
Synchronicity
Sacred Love

Figure 16: Search Interface for Basic Search



Search products with

all words: sting
exact phrase:
at least 1 word: guy ritchie
without these:

Found 1 matching products.

Lock, Stock and Two Smoking Barrels

Figure 17: Search Interface for Advanced Search

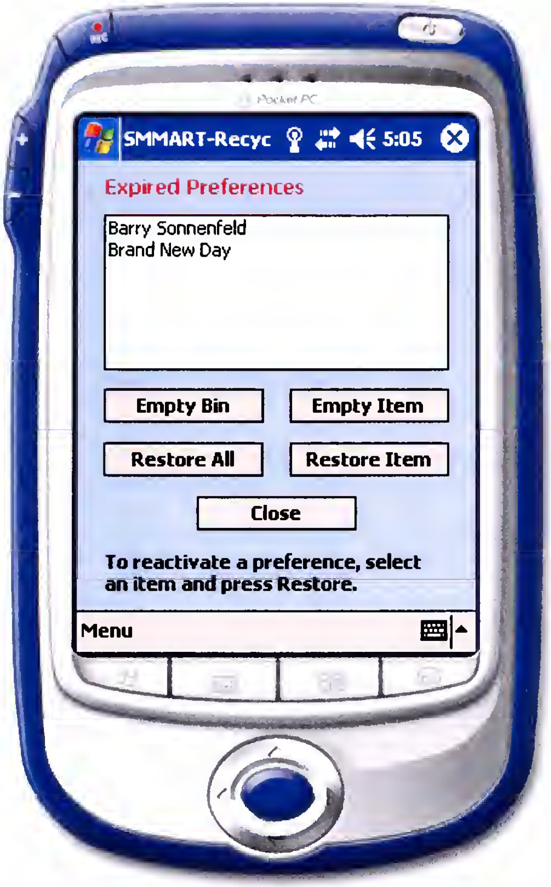
SMMART provides the user with the option of a Basic Search as shown in Figure 16 or an advanced Search as shown in Figure 17. While the basic search allows the users to search for products by just entering one or more keywords, the advanced search gives users more control over the search results and assists in narrowing the search focus.

The basic search screen is a simple interface that allows the users to enter their choice of keyword(s) and displays the relevant results. The advanced search screen is modeled after Google search and provides more options to narrow down the search. The users can search for products whose descriptions contain all keywords, the exact phrase, at least one search keyword, as well as those products that do not contain the given list of keywords. These four options can be used in any combination. As shown in **Error! Reference source not found.**, the result of basic search indicates six matching products featuring *Sting*. However, in **Error! Reference source not found.**, the search result has been narrowed down to a single product, *Lock, Stock and Two Smoking Barrels*, which features *Sting* and directed by *Guy Ritchie* by specifying the keyword *Guy Ritchie* in the "at least 1 word" field in the advanced search feature.

### IV.D.2. Recycle Bin

As explained above in the discussion of the Maintenance Agent, while the Update Manager unobtrusively performs its function of adding or updating preferences depending on the previous presence of the preference in the Preference Data, some garbage value might crop in the data stored and may end up in the display of erroneous products. The Maintenance Agent performs a check for such keywords and transfers them from the Preference Data into the Recycle Bin. The above-mentioned process is automatic and runs in the background. The Recycle Bin is shown in Figure 18.

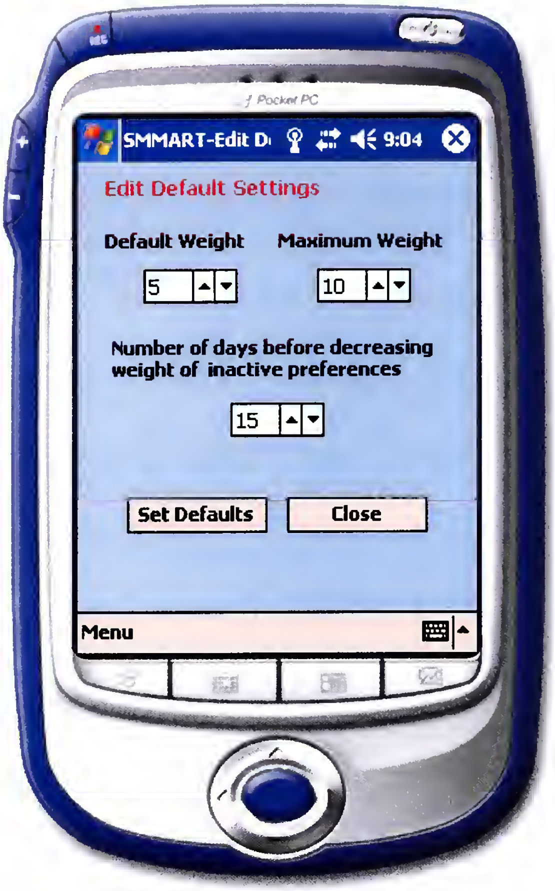
The Recycle Bin gives the user options to either delete the preference(s) permanently or restore them back to the Preference Data. The user has the choice to delete or restore preference(s) one at a time or all at once. If the user decides to restore the preference (s), they are added to the Preference Data with the default weight and the current time. Once preferences are deleted from the Recycle Bin, they cannot be recovered again.



Expired Preferences

Barry Sonnenfeld
Brand New Day

To reactivate a preference, select an item and press Restore.



Edit Default Settings

Default Weight: 5

Maximum Weight: 10

Number of days before decreasing weight of inactive preferences: 15

**Figure 18: Recycle Bin**

**Figure 19: Default Settings Editor**

### IV.D.3. Default Settings Editor

The Maintenance Agent and the Update Manager use default values for the initial weight and maximum weight of a preference and also for the number of days before the weight of a preference can be decreased. The default weights in the above mentioned order are 5, 10 and 15 respectively. This are the standard set of values that come with the application. These values can be customized by the user depending on the user's interest upon invoking the Default Settings tool which is as shown in Figure 19. When the defaults are altered, the system remembers the user's settings and uses these values for future maintenance and automatic editing of preferences.

## IV.E. SMMART Framework Implementation

In order to implement a fully functional prototype, we needed to identify the technologies for development. While choosing the technology for the implementation of SMMART framework, we looked for the following features.

- Ease of development: The underlying technology should support wide variety of languages and operating systems. This gives us more options on our choice of languages and operating systems for implementation.
- Efficient Performance: The technology should facilitate development of robust and efficient applications by providing the necessary infrastructure.
- Support for XML Web Services: As we intended to use XML Web Services for client\server communication, the technology should offer support for XML Web Services.
- Easy Database connectivity: The technology should provide the framework for communicating with various databases without minimal effort.
- Facilitate SMMART client development on smart devices: As the SMMART Client is intended for deployment on smart devices, the technology should provide sufficient features for developing an efficient client application.

We considered some of the most widely used technologies such as Microsoft's .NET and Sun's Jini Networking Technology before coming up with our choice for development. Sun's Jini is an open source implementation in java that allows for self-healing, self-configurable networks and adaptive networks. It enables spontaneous networking of services and eliminates the distinction between hardware devices and software by considering everything as a service. Some of the distinctive features of Jini include its support for discovery, join, lease and events that allows dynamic plug and play. Though the above features make it an attractive choice, Jini is a relatively new technology and has very limited support for developing mobile applications at present thus making it not so effective for implementing our framework.

Microsoft .NET Framework is a software suite that facilitates the development of software applications as well as Web services. The .NET Framework has a rich set of features that make it a viable choice for implementing our framework. The .NET framework supports multiple programming languages, provides a consistent development interface across all supported languages through its base class libraries, has native support for XML through XML classes that enable manipulation, searching and translations of XML documents, has extensive support for XML web services, provides a platform for building and deploying secure and high-performing applications, facilitates easy database connectivity through ADO.NET classes, enables software developers to focus primarily on the business logic by managing the required groundwork for software development through its Common Language Runtime (CLR) and has a rich set of features for mobile and smart client development through its subset termed as .NET Compact Framework. Also, .NET supports web applications and web services through ASP.NET and desktop based smart client applications through Windows forms.

The heart of the .NET framework is its common language runtime (CLR). CLR accounts for services such as memory, process and thread management, language integration, life-cycle management, exception handling etc that aid the developer in developing robust and efficient code resulting in reduced development time.

The .NET Compact Framework is a subset of the .NET Framework optimized for facilitating application development for small devices. Some of the features of .NET Compact Framework are its seamless connectivity with web services, a compact CLR that performs memory management, provides language neutrality etc for optimal performance, classes for managing XML data and documents and complete access to the features of the native platform.

We developed the SMMART client on PDA's running Windows CE using C# as the language of development on Microsoft .NET Compact Framework. The Server was developed using C# and ASP.NET on Microsoft .NET. Our inventory database runs on SQL Server 2000.

We chose XML Web Services for communication between SMMART clients and servers [20] due to the following reasons.

- The architecture of a typical web service fits well in the general philosophy of SMMART. A number of functionally and semantically related methods are united under the umbrella of a single service. All methods work with the same data, i.e. the store's inventory.
- Universal Description, Discovery and Integration (UDDI) entries facilitate the discovery of each individual store's service by their clients and provide an easy way to identify existing web services. As each SMMART site routes all network

traffic from the access point only to the web server hosting SMMART web services, this enables an unambiguous identification of the store that the SMMART Client is connected to and the availability of the corresponding SMMART services.

- Using XML Web Services help SMMART framework overcome the burden of possible network disconnections. This is because XML Web Services follow the stateless connection paradigm, i.e. the server (Web Service provider) does not maintain the record of, nor does it require any particular sequence of the service invocations.
- As XML Web Services is supported by big names such as Sun Microsystems, Microsoft Corporation, IBM, HP etc, it enables us with more choices with respect to the implementation technologies. We could chose to develop the client using .NET framework while the Server could be implemented using java technologies.
- Using XML Web Service in the SMMART framework enabled the framework to support complex data type exchange. This is due to the ability of the XML Web Services to transport both simple and complex data types in XML format [21].

## IV.F.Economic Feasibility of SMMART

SMMART is a mobile marketing framework that is aimed at increasing the revenue of the participating store or a chain of stores by providing targeted advertisements to its consumers. But, in order for the stores to adopt our framework they need substantive evidence on ROI (Return on Investment). The store needs to invest on the infrastructure for setting up the SMMART environment as well as its maintenance. This necessitates economic justification, i.e. the income generated from additional sales by the customers using SMMART framework must be greater than the total costs for setup and maintenance of SMMART framework. We have developed a simulation model [1] that justifies our claim.

We consider a situation wherein n customers frequent the participating store with SMMART framework over a certain period of time. Several independent parameters impact the upshot of this experiment directly and are described below. Our objective is to measure the impact of SMMART framework on sales at the given retail site, which is denoted by F (P, S, I). It is measured as the ratio of the total sales to the sales generated only by the customers who do not use SMMART Clients and are thus uninfluenced by promotions received through them.

$$F(P,S,I)=\frac{\sum_{j=1}^{n}\left\{A_1\left(P(c_j)\right)+A_2\left(S(c_j)\cdot I(c_j)\right)\right\}}{\sum_{j=1}^{n}A_1\left(P(c_j)\right)}$$

Where

P(cj) = The probability that a customer cj will make a purchase.

A1(P(cj))= The amount when a customer cj decides to make a purchase, which is distributed normally.

S(cj) = The probability that the customer cj is using a mobile device equipped with SMMART Client.

I(cj) = The probability that this customer cj will receive a promotion from SMMART and make a purchase influenced by it.

A2(S(cj)·I(cj)) = The amount of the respective purchase which is distributed normally.

We conducted a number of experiments with different values of the independent variables with the simulation model and the experimental results are indicated in the graphs shown in Figure 20.

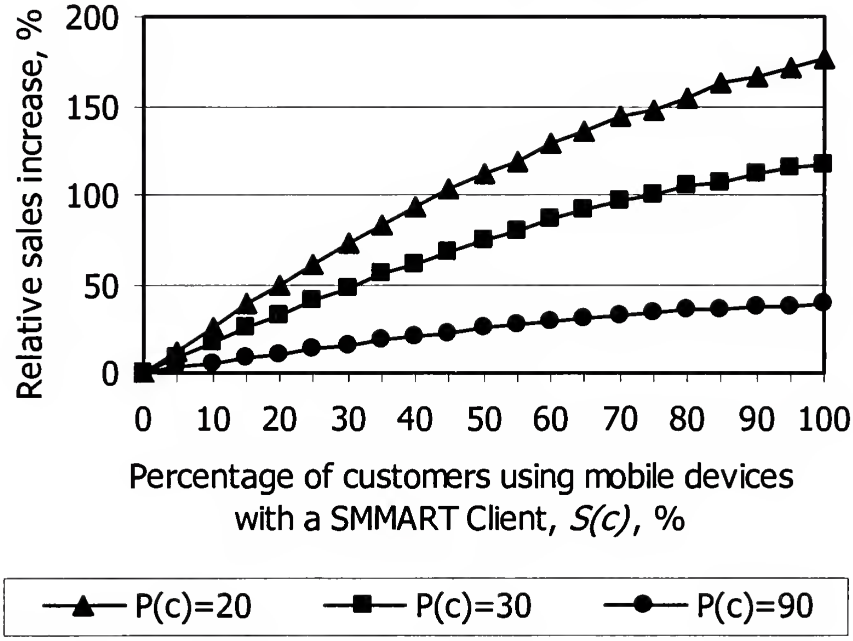


**Figure 20: Graph depicting relative increase in sales for varying percentage of mobile customers**

The graphs shown above indicate the results obtained for the following values of the independent variables.

- I(c) = 50%.
- P(c) = Values varying from 20% to 90%.
- n = 100,000.

As can be observed from Figure 20, SMMART framework has more impact on sales for low values of P(c) and shows a decreasing trend for higher values of P(c). As an example to illustrate the above claim, let us consider the case where only 5% of all customers are carrying mobile devices with SMMART Clients (S(c) = 5%). For P(c) = 20%, SMMART framework increases sales by around 13% whereas for (P(c) = 90%), SMMART causes a not so significant 3% increase in sales. Thus, SMMART can be effective in stores where people not only for shopping but also to socialize as well.

# V. Conclusion and Future Work

In this thesis we presented the SMMART framework, which is an innovative approach for targeted, personalized and adaptive mobile marketing in context-aware environments. SMMART aids the shoppers in a store with targeted promotions that are currently on sale in a retail store based on the context of their location and precise matching to their preferences. SMMART also adapts dynamically according to the user's interests and displays only relevant offers on the fly. From the shopper's perspective, SMMART helps the buyer to identify and find interesting products without having to browse through the many pages of promotional brochures often available at retail stores. From the seller's perspective, SMMART creates more buying opportunities, thus potentially increasing the profits.

We examined several proposed mobile marketing frameworks with respect to their uniqueness in their approach and implementation and found that most of them were location based frameworks and hence entailed requirement for special devices such as location sensors or Bluetooth support making them less attractive choices. Comparison of intended features of SMMART with the features of other mobile marketing frameworks highlighted the suitability of the SMMART framework for delivering targeted promotions.

The thesis described the overall architecture of the SMMART framework explaining each module in detail along with the algorithms used. The working details of the SMMART framework included extensive description along with the screenshots of the prototype to enable better understanding of our approach.

A working prototype of the SMMART framework was built to demonstrate the feasibility of using our technique. We conducted several experiments involving many users and were able to successfully demonstrate that the system can effectively adapt to the user's changing interests, find promotions that match the user's interests and display them on the client PDA ordered by the relevancy to the user's preferences. We also studied the economic feasibility of SMMART that indicate its effectiveness in stores where customers need additional incentives to make purchases. Possible examples include stores in shopping malls, bookstores, consumer electronics warehouses, and any other retailers where consumers come not only to shop, but also to socialize.

This work could be extended by enhancing the framework to handle more resource-constrained devices, such as cellular phones, wherein the location information could be used to enhance the framework's capability. Location information could be used to provide location specific services to the user. This would necessitate research in exploring various possibilities of handling data and location information in mobile networks in order to select the best fit or coming up with a newer approach.

The prototype could also be extended with additional features such as an enhancement in the product information page of the SMMART Browser with a map displaying the exact location of the product in the current store when needed. The prototype could also be provided with the capability to purchase an in stock product for instant pick-up or on a designated date and time, reserve a product that is on back order at the sale price.

# References

[1] S. Kurkovsky, K. Harihar, J. Hudson. SMMART: Personalized Adaptive System for Mobile Marketing. Submitted to The 2005 IEEE Consumer Communications and Networking Conference (CCNC-05), Las Vegas, NV, January 2005.

[2] Dey. Understanding and Using Context, Personal and Ubiquitous Computing, Volume 5, Issue1, February 2001, Pages 4-7, Springer-Verlag

[3] O. Ratsimor, T. Finin, A. Joshi, Y. Yesha, eNcentive: a framework for intelligent marketing in mobile peer-to-peer environments,In Proceedings of the 5th international conference on Electronic commerce, Pittsburgh, Pennsylvania, September 2003, Pages: 87 – 94, ACM Press

[4] N. Hristova, G.M.P. O'Hare. Ad-me: Wireless Advertising Adapted to the User Location, Device and Emotions In Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), Hawaii, January 2004, IEEE Computer Society.

[5] L. Aalto, N. Göthlin, J. Korhonen, T. Ojala, Bluetooth and WAP push based location-aware mobile advertising system, In Proceedings of the 2nd international conference on Mobile systems, applications, and services, Boston, MA, Pages: 49 – 58, ACM Press

[6] S. Kurkovsky, K. Harihar. "Intelligent Pervasive Framework for Consumer-Supplier Interaction," In Proceedings of The 2004 International Conference on Pervasive

Computing and Communications (PCC-04), Las Vegas, NV, June 2004, CSREA Press.

[7] U. Varshney, Addressing Location Issues in Mobile Commerce, In Proceedings of the 26th Annual IEEE Conference on Local Computer Networks, November 2001, Page 184, IEEE Computer Society Press

[8] U. Varshney, Location management for mobile commerce applications in wireless Internet environment, ACM Transactions on Internet Technology, Volume 3, Issue 3, August 2003, Pages: 236 – 255, ACM Press

[9] Strategy Analytics Consulting, http:// www.strategyanalytics.com

[10] Yankee Group Research, http://www.yankeegroup.com

[11] Ovum Research, http://www.ovum.com

[12] Kelsey Group Consulting, http://www.kelseygroup.com

[13] U. Varshney, Location management support for mobile commerce applications, In Proceedings of the 1st international workshop on Mobile commerce, Rome, Italy, July 2001, Pages: 1 – 6, ACM Press

[14] P. Kannan, A. Chang, A. Whinston, Wireless Commerce: Marketing Issues and Possibilities, In Proceedings of the 34th Annual Hawaii International Conference on System Sciences, Volume 9, January 2001, Page 9015, IEEE Computer Society Press

[15] H. Yunos, J. Gao, S. Shim, Wireless Advertising's Challenges and Opportunities, Computer, Volume 36, Issue 5, May 2003, Pages: 30 – 37, IEEE Computer Society Press

[16] T. Stafford, M. Gillenson, Mobile commerce: what it is and what it could be, Communications of the ACM, Volume 46, Issue 12, December 2003, Pages: 33 – 34, ACM Press

[17] Jini Specification. http://wwws.sun.com/software/jini/specs

[18] A. Wigley, S. Wheelwright. Microsoft .NET Compact Framework. Microsoft Press, 2002.

[19] .NET framework Developer Center. http://msdn.microsoft.com/netframework/

[20] T. Pilioura, T. Tsalgatidou, S. Hadjiefthymiades. "Scenarios of Using Web Services in M-Commerce." ACM SIGecom Exchanges. Vol. 3, No. 4, Jan. 2003, pp. 28-36.

[21] A. Ferrara, M. MacDonald, .NET Web Services, O'Reilly, 2002

# Appendix

## Database Schema

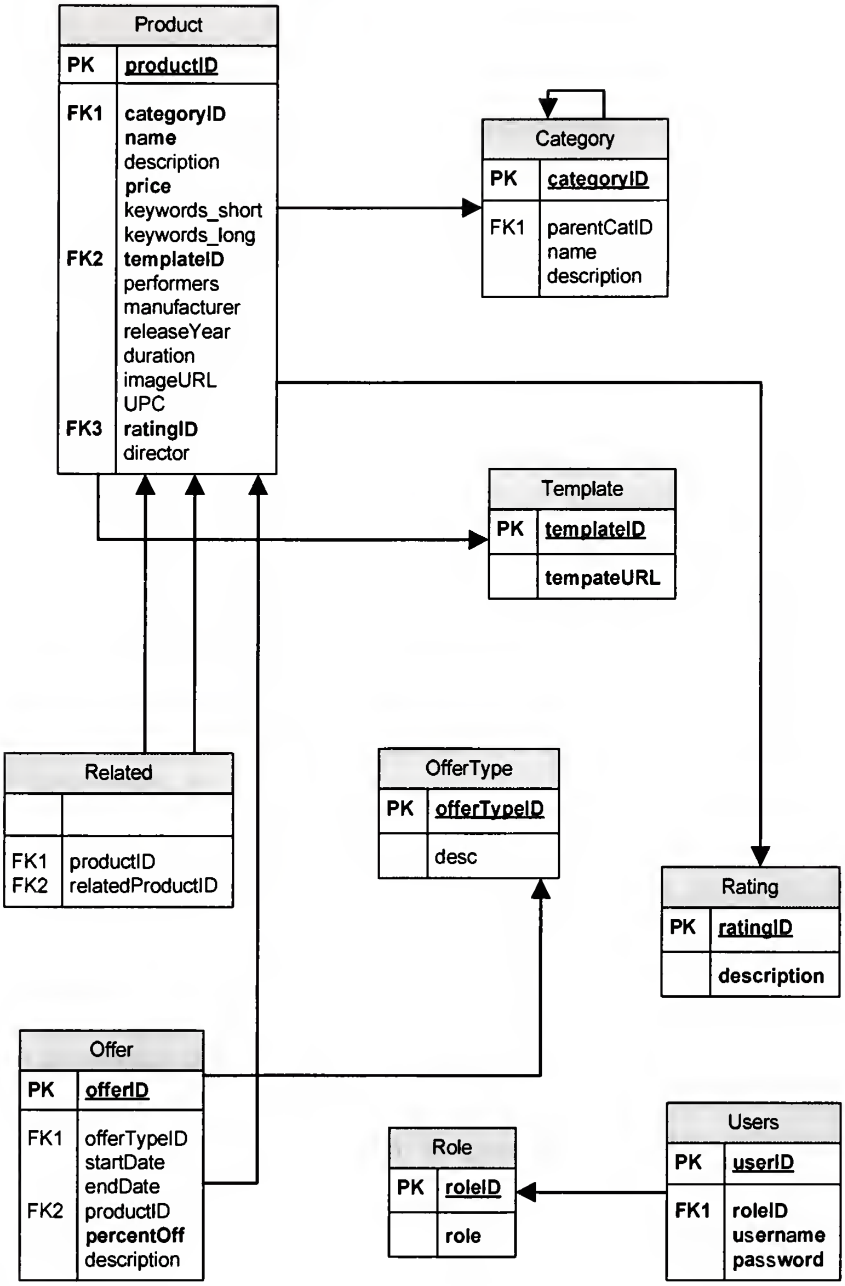


**Figure 21: Schema of the Inventory Database**

## Inventory Data Administration

The Inventory database of the store holds the necessary data of the products with and without promotions in the store. The data is not limited to a very small set, but contains a large number of products. Though the administrator of the store's database can access and edit the database, the task is often hard and cumbersome. The following points illustrate the difficulties that can be encountered.

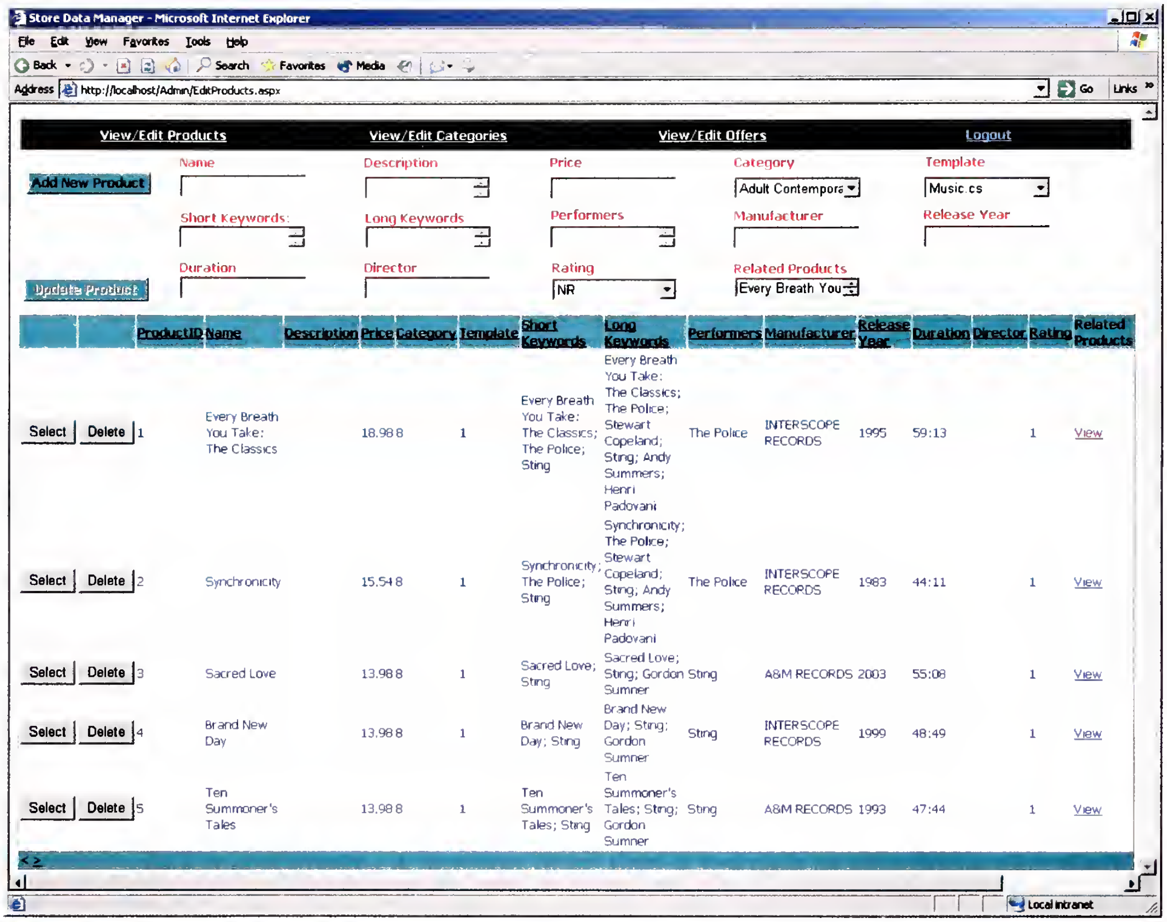
1. As the data may be spread over several tables, accessing the data in a single stroke requires one to write and run complex queries.

2. Entering the data spread over multiple tables is tedious and time consuming.

3. It is possible for erroneous data to be entered in the database as there are no additional application specific constraints that are validated for, apart from the ones offered by the database itself. For example, if an offer has a start date and an end date, though the data is entered in the correct format specified by the database, the end date might end up being entered as lesser than that of the start date. Checking for these constraints require additional work on the part of the administrator.

The above mentioned difficulties are in sharp contrast to using a compact interface to manage data as it offers more convenience and ease of use. In order to overcome these problems and to provide for efficient and effective Inventory data management, we developed an application that provides clear and well-defined interfaces for managing data. The Store Data Manager has separate interfaces that mostly map one-on-one with the tables in the Inventory database. The Interfaces gives the administrator better control over the management of Inventory data.

Figure 22 shows the screenshot of the interface to edit product information. Similar interfaces exist for editing categories and offers. As can be seen, the administrator can view complete information of products with ease. This interface facilitates adding new products as well as updating or deleting existing products. The data entered is validated against the framework and database specifications. This ensures that the data entered is not erroneous.



Address http://localhost/Admin/EditProducts.aspx

| | | | | |
|---|---|---|---|---|
| Name | Description | Price | Category | Template |
| | | | Adult Contempora | Music.cs |
| Short Keywords: | Long Keywords | Performers | Manufacturer | Release Year |
| Duration | Director | Rating | Related Products | |
| | | NR | Every Breath You | |

| | | ProductID | Name | Description | Price | Category | Template | Short Keywords | Long Keywords | Performers | Manufacturer | Release Year | Duration | Director | Rating | Related Products |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Select | Delete | 1 | Every Breath You Take: The Classics | | 18.98 | 8 | 1 | Every Breath You Take: The Classics; The Police; Sting | Every Breath You Take: The Classics; The Police; Stewart Copeland; Sting; Andy Summers; Henri Padovani | The Police | INTERSCOPE RECORDS | 1995 | 59:13 | | 1 | View |
| Select | Delete | 2 | Synchronicity | | 15.54 | 8 | 1 | Synchronicity; The Police; Sting | Synchronicity; The Police; Stewart Copeland; Sting; Andy Summers; Henri Padovani | The Police | INTERSCOPE RECORDS | 1983 | 44:11 | | 1 | View |
| Select | Delete | 3 | Sacred Love | | 13.98 | 8 | 1 | Sacred Love; Sting | Sacred Love; Sting; Gordon Sumner | Sting | A&M RECORDS | 2003 | 55:08 | | 1 | View |
| Select | Delete | 4 | Brand New Day | | 13.98 | 8 | 1 | Brand New Day; Sting | Brand New Day; Sting; Gordon Sumner | Sting | INTERSCOPE RECORDS | 1999 | 48:49 | | 1 | View |
| Select | Delete | 5 | Ten Summoner's Tales | | 13.98 | 8 | 1 | Ten Summoner's Tales; Sting | Ten Summoner's Tales; Sting; Gordon Sumner | Sting | A&M RECORDS | 1993 | 47:44 | | 1 | View |

< >

**Figure 22: Interface for Inventory Data Administration**

The following are some of the features of the Store Data Manager application.

- Convenient and easy-to-use interfaces that mostly map one-on-one with the database tables. As the interfaces displays the relevant data from related tables in order to aid the user, any necessary changes to the data in the relevant tables are filled automatically. For example, as a product can have other products related to it, the interface displays the related products. Thus, when instructed by the administrator to make the necessary data changes for a particular product, the application makes the necessary changes in the product data and also fills the data in the 'Related Products' field automatically to the corresponding table. This eliminates the need for a separate interface or a user initiation for making these changes.
- Data is verified against the framework as well as the database specifications before being entered into the database. The application checks the data for framework as well as database constraints and performs the required operations initiated by the administrator only when the constraints are verified. It provides a brief description of any problem in the data. This enables the administrator with the necessary information to make corrections to the data before re-submitting.

Though, the Store Data Manager is an application developed for effective Inventory data management, it is worth more than a mention in this thesis as it plays an important part during any necessary data changing operations.

## Use Cases

**Purpose:**

The purpose of this design document is to capture the high level requirements of the project "SMMART Client" in the form of use cases. The following are the use cases.

| Actor | Use Case |
|---|---|
| Consumer | Browse Offers |
| | Manage Preferences |
| | View Product |
| | Browse Category |
| | Search Inventory |
| | Edit Auto-Deleted Preferences |

**Use Case UC1: Browse Offers**

**Primary Actor:** Customer

**Stakeholders and Interests:**

Customer – Wants to view current available offers that are of interest to him in the present store.

SMMART Service – Wants to receive the preferences in the proper format in order to process the data and return proper results.

**Preconditions:** Customer is in close proximity to the store.

Success Guarantee (Postconditions): Network is found. Offers are displayed.

**Main Success Scenario:**

1. Customer starts the application.

2. System checks for the available store networks and displays them.

3. Customer selects a store network to connect.

4. System checks for stored preferences of the Customer. Preferences that are of no interest to the Customer are removed while those of less interest are reduced in weight. The preferences are sorted alphabetically.

5. System connects to the SMMART Service in the store network and communicates the Customer preferences.

6. System displays the current offers in the store that match the Customer preferences.

7. Customer selects a product to view the details.

8. System presents product information: View Product

**Extensions:**

*a. At any time System fails:

1. System issues a warning to the Customer of the failed condition and exits.

*b. At any time, Customer selects to edit preferences: Manage Preferences

*c. At any time, Customer selects to quit:

    1. System exits.

2a. System is unable to locate a network:

System displays a warning and then retries.

1a. Customer ignores the warning:

System exits.

4a. System does not find any stored preferences:

System informs the Customer to add preferences.

1a. Customer ignores the information:

1. System displays all available offers.

Customer adds the preferences: Manage Preferences.

6a. No offers match the Customer preferences:

System informs the Customer to change preferences.

1a. Customer ignores the information:

1. System displays all available offers.

Customer edits the preferences: Manage Preferences.

**Use Case UC2: Manage Preferences**

**Primary Actor:** Customer

**Stakeholders and Interests:**

Customer – Wants to add/update/delete preferences of interest.

**Preconditions:** None

**Success Guarantee (Postconditions):** Preferences are added/updated/deleted.

**Main Success Scenario:**

1. System displays the available preferences.

2. Consumer edits the preference.

3. The system sorts the preferences alphabetically.

4. System saves the information.

**Extensions:**

*a. At any time System fails:

1. System issues a warning to the Customer of the failed condition and exits.

*b. At any time, Customer selects to quit:

1. System exits.

*c. At any time, Customer selects to edit default maintenance settings.

1. System saves and applies the new settings.

*d. At any time, Customer selects to edit automatically deleted preference information: Edit Auto-Deleted Preferences

2a. Customer adds a new preference.

System adds the preference.

1a. Preference already exists.

1. System overwrites the existing preference.

2b. Customer selects an existing preference:

System displays the preference information.

Customer updates the information.

2c. Customer deletes a preference.

1. System removes the preference from the preferences.

**Use Case UC3: View Product**

**Primary Actor:** Customer

**Stakeholders and Interests:**

Customer – Wants to view the product details.

SMMART Service – Wants to receive the request details in a proper format to process the data and return proper results.

**Preconditions:** System is connected to a participating store's network.

**Success Guarantee (Postconditions):** Complete Product information is displayed. User preference details are updated with the corresponding product information.

**Main Success Scenario:**

1. System contacts the SMMART Service and displays the complete product information which includes associated offer and related products.

2. System checks the stored customer preferences for any preferences matching the product keywords and updates the preference information.

3. System sorts the preferences alphabetically and saves the information.

**Extensions:**

*a. At any time System fails:

1. System issues a warning to the Customer of the failed condition and exits.

*b. At any time, Customer selects to edit preferences: Manage Preferences

*c. At any time, Customer selects to quit:

    1. System exits.

1a. Customer chooses to explore a related product in detail: View Product.

1b. Customer chooses to explore a category: Browse Category

2a. No preferences match the product keywords:

System adds new preferences that correspond to the product information.

**Use Case UC4: Browse Category**

**Primary Actor:** Customer

**Stakeholders and Interests:**

Customer – Wants to view the sub-categories of a category or products associated with a particular category.

SMMART Service – Wants to receive the request details in a proper format to process the data and return proper results.

**Preconditions:** System is connected to a participating store's network.

**Success Guarantee (Postconditions):** Sub-categories or products belonging to a category are displayed.

**Main Success Scenario:**

1. System contacts the SMMART Service and displays the sub-categories of the category.

**Extensions:**

*a. At any time System fails:

1. System issues a warning to the Customer of the failed condition and exits.

*b. At any time, Customer selects to edit preferences: Manage Preferences

*c. At any time, Customer selects to quit:

    1. System exits.

1a. No sub-categories exist for the category:

System displays the products belonging to the category.

1b. Customer chooses to explore a category: Browse Category

**Use Case UC5: Search Inventory**

**Primary Actor:** Customer

**Stakeholders and Interests:**

Customer – Wants to search entire Store Inventory for products with or without offers.

SMMART Service – Wants to receive the search keywords in the proper format in order to process the data and return proper results.

**Preconditions:** System is connected to a participating store's network.

**Success Guarantee (Postconditions):** Products that match the search keywords are displayed.

**Main Success Scenario:**

1. Customer enters the search keywords.

2. System contacts the SMMART Service and communicates the search keywords.

3. System displays the products matching the search.

4. Customer chooses to explore an offer in detail: View Product.

**Extensions:**

*a. At any time System fails:

1. System issues a warning to the Customer of the failed condition and exits.

*b. At any time, Customer selects to edit preferences: Manage Preferences

*c. At any time, Customer selects to quit:

    1. System exits.

1a. Customer is interested in narrowing the search.

System provides Customer with a more advanced search option.

Customer enters the search details.

**Use Case UC6: Edit Auto-Deleted Preferences**

**Primary Actor:** Customer

**Stakeholders and Interests:**

Customer – Wants to either purge or restore the deleted preferences.

**Preconditions:** None

**Success Guarantee (PostConditions):** Keywords are purged or restored to the list of preferences.

**Main Success Scenario:**

1. System displays deleted preference (s).

2. Customer chooses to edit a preference.

3. System saves the information.

**Extensions:**

*a. At any time System fails:

1. System issues a warning to the Customer of the failed condition and exits.

*b. At any time, Customer selects to quit:

1. System exits.

2a. Customer restores a preference.

1. System restores the preference along with a default setting to the list of preferences.

2b. Customer deletes a preference.

System removes the preference completely.

2c. Customer restores all preferences at once.

System restores all the preferences along with a default setting to the list of preferences.

2b. Customer deletes all preferences at once.

System removes all the preferences completely.

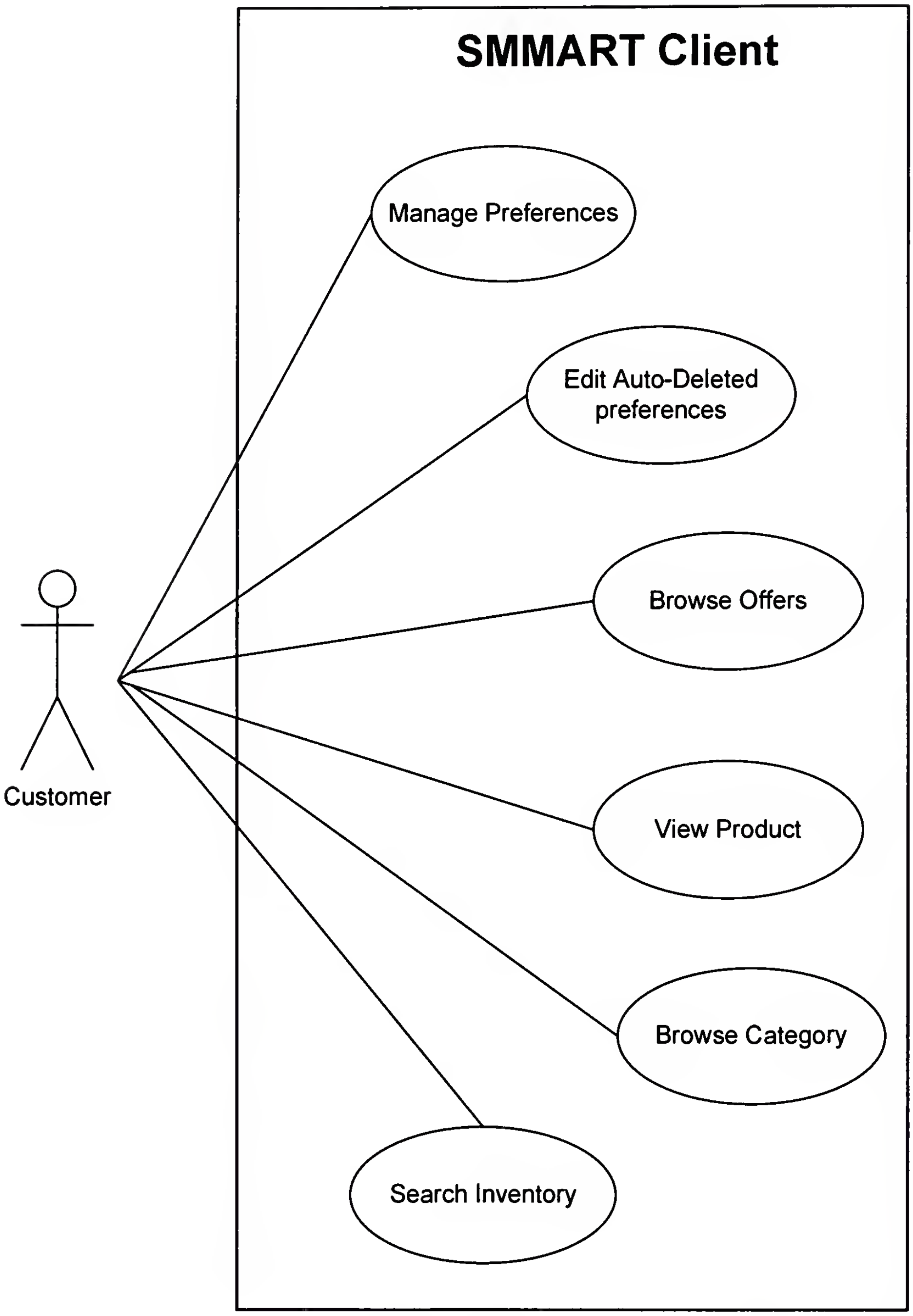


**Figure 23: Use Case diagram for SMMART Client**

# SQL Scripts for creating tables for Inventory Database

a) Category

```
CREATE TABLE Category (
categoryID int PRIMARY KEY IDENTITY (1, 1) NOT NULL ,
parentcategoryID int FOREIGN KEY REFERENCES Category ,
name varchar (50) ,
description varchar (200)
);
```

b) Template

```
CREATE TABLE Template (
templateID int PRIMARY KEY IDENTITY (1, 1) NOT NULL ,
templateURL varchar (400) NOT NULL
);
```

c) Rating

```
CREATE TABLE Rating (
ratingID int PRIMARY KEY IDENTITY (1, 1) NOT NULL ,
description varchar (5) NOT NULL
);
```

d) Product

```
CREATE TABLE Product (
productID int PRIMARY KEY IDENTITY (1, 1) NOT NULL ,
categoryID int FOREIGN KEY REFERENCES Category NOT NULL,
name varchar (50) NOT NULL ,
description varchar (200) ,
price float NOT NULL ,
keywords_short varchar (300) ,
keywords_long varchar (300) ,
templateID int FOREIGN KEY REFERENCES Template NOT NULL ,
performers varchar (200) ,
manufacturer varchar (100) ,
releaseYear int ,
duration varchar (10) ,
imageURL varchar (200) ,
UPC varchar (20),
ratingID int FOREIGN KEY REFERENCES Rating NOT NULL ,
director varchar (200)
);
```

e) OfferType

```
CREATE TABLE OfferType (
offerTypeID int PRIMARY KEY IDENTITY (1, 1) NOT NULL ,
description varchar (200)
);
```

f) Offer

```
CREATE TABLE Offer (
offerID int PRIMARY KEY IDENTITY (1, 1) NOT NULL ,
offerTypeID int FOREIGN KEY REFERENCES OfferType NOT NULL ,
productID int FOREIGN KEY REFERENCES Product NOT NULL ,
startDate datetime NOT NULL,
endDate datetime NOT NULL,
percentOff int NOT NULL,
description varchar (200)
);
```

g) Related

```
CREATE TABLE Related (
productID int FOREIGN KEY REFERENCES Product NOT NULL ,
relatedProductID int FOREIGN KEY REFERENCES Product NOT NULL
);
```

h) Role

```
CREATE TABLE Role (
roleID int PRIMARY KEY IDENTITY (1, 1) NOT NULL ,
role varchar (30) NOT NULL
);
```

i) Users

```
CREATE TABLE Users (
userID int primary key IDENTITY (1, 1) NOT NULL ,
roleID int FOREIGN KEY REFERENCES Role NOT NULL ,
username varchar (30) NOT NULL ,
password varchar (30) NOT NULL
);
```

## Stored Procedures for Inventory Database

a) Procedure Name: sp_catMgr_getCategories1

```
CREATE PROCEDURE sp_catMgr_getCategories1 @productID int

AS

SELECT categoryID

FROM product

WHERE productID = @productID

GO
```

b) Procedure Name: sp_catMgr_getCategories2

```
CREATE PROCEDURE sp_catMgr_getCategories2 @categoryID int

AS

SELECT categoryID, parentCategoryID, name

FROM category

WHERE categoryID = @categoryID

GO
```

c) Procedure Name: sp_catMgr_getCategory

```
CREATE PROCEDURE sp_catMgr_getCategory @categoryID int

AS

SELECT categoryID, parentCategoryID, name

FROM category

WHERE parentCategoryID = @categoryID

GO
```

d) Procedure Name: sp_catMgr_getProduct

```
CREATE PROCEDURE sp_catMgr_getProduct @categoryID int
AS
SELECT productID, name
FROM product
WHERE categoryID = @categoryID
GO
```

e) Procedure Name: sp_prodMgr_getProduct

```
CREATE PROCEDURE sp_prodMgr_getProduct @productID int
AS
SELECT p.productID, p.categoryID, p.name, p.price, p.keywords_short, p.templateID,
p.performers, p.releaseYear, p.duration, rating = r.description, p.director, p.imageURL,
startDate = ISNULL (convert (varchar, o.startDate, 101), ' '),
endDate = ISNULL (convert (varchar, o.endDate, 101),' '),
percentOff = ISNULL (o.percentOff, 0)
FROM Product p INNER JOIN Rating r ON p.ratingID = r.ratingID
LEFT OUTER JOIN Offer o
ON p.productID = o.productID
WHERE p.productID =@productID
GO
```

f) Procedure Name: sp_prodMgr_getRelatedProduct

```
CREATE PROCEDURE sp_prodMgr_getRelatedProduct @productID int
AS
SELECT p.productID, p.name FROM Related r INNER JOIN Product p
```

```
ON r.relatedProductID = p.productID

WHERE r.productID = @productID

GO
```