



Ramkumar Patturaj

SERVER FOR A BEST SALE APPLICATION

SERVER FOR A BEST SALE APPLICATION

Ramkumar Patturaj
Master's thesis
Spring 2014
Degree Programme in Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Master of Information Technology

Author: Ramkumar Patturaj

Title of Master's thesis: Server for a Best Sale Application

Supervisor: Dr. Kari Laitinen

Term and year of completion: Spring 2014

Number of pages: 52

The aim of this Master's thesis was to build a web server providing an Rest API endpoint which could be used to develop platform independent applications displaying online advertisement content for mobiles and tablets. Reducing wastage of paper, minimizing the cost of advertisement, improving the accessibility of the best discount offer for the end user/customer and maintaining the statistical data of online advertisements are some of the practical problems discussed in this thesis.

To achieve that aim it was decided to develop a web service using J2EE with a Java programming language. A tomcat servlet container was used as the web server and MySQL database for storing the information. The advertisements information obtained from shop website as RSS content are polled, parsed and saved in database. This information is available for the application developer as Rest API endpoints pertaining to every shop categorized based on city.

Although this project was an internal work during the Master's thesis project, the online advertisement concept still carries a significant potential for future upgrading and a wide utilization in the mobile market. The application and the thesis concept if realized will have overwhelming response from the shop and the end users.

Keywords:

Best Sale, offer/discount, mobile application, RSS, Content Management, Database

PREFACE

This thesis represents the culmination of work and knowledge that I have gained while working for Nokia Oyj, Oulu Finland.

Firstly, I would like to express my sincere thanks to Nokia Oyj, for all the good things that has happened to me; Mr Pasi Turpeinen, Product Program Manager, for the opportunities and support; Mr Heikki Savola-Palokangas, Project Manager, for his trust and belief in my abilities; Mr Jani Mallila, Architect and Product Manager, for his guidance and careful explanation of the architectural details of the work and for providing me the steepest learning curve of my career; Mr Olli Miettinen for his technical guidance at all times during the work

My sincere thanks goes to Dr Kari Laitinen for his humble, yet truly admirable ability to help writing this work in better words.

Last, but by no means least, I thank my parents and my family, specially my loving wife, Yoganandhi Raja for her constant support and wit; and patiently accepting few of my outbursts with soothing smiles.

Oulu, Finland, December 2013

Ramkumar Patturaj

CONTENTS

ABSTRACT	3
PREFACE	4
ABBREVIATIONS	7
LIST OF FIGURES	9
LIST OF TABLES	10
1 INTRODUCTION	11
2 CONCEPTUAL DESIGN – A BEST SALE APPLICATION	13
3 UI – BEST SALE APPLICATION	14
4 SERVER – BEST SALE APPLICATION	15
4.1 Used Technologies	15
4.2 Core – Server for a Best Sale Application	16
4.2.1 Polling Concept	16
4.2.2 Threads	19
4.2.2.1 Concurrency	19
4.2.2.2 Java Threads	19
4.2.2.3 Life Cycle	20
4.2.2.4 Thread Pooling	21
4.2.3 RSS Content	23
4.2.3.1 Benefits of RSS	24
5 CONTENT MANAGEMENT PORTAL	27
5.1 Login Page	27
5.2 Search Page	29
5.3 Health Check Page	30
5.4 Stats Page	31
5.5 Add URL Page	32
5.6 Add Login Page	34
6 REST API ENDPOINTS	35
6.1 Rest Endpoint – Shops List API	37
6.2 Rest Endpoint – Items List API	38
7 DATABASE	40
7.1 Table – Users	40

7.2	Table – Products	40
7.3	Table – Shops	41
7.4	Core-Server Execution Logic	42
8	FURTHER RESEARCH AND STUDY	49
9	CONCLUSION	51

ABBREVIATIONS

Term	Description
RSS	<p>Rich Site Summary</p> <p>Uses a family of standard web feed formats to publish frequently updated information: blog entries, news headlines, audio and video.</p>
SDK	<p>Software Development Kit</p> <p>Is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.</p>
IOS	<p>iPhone Operating System</p>
URL	<p>Uniform Resource Locator</p> <p>A uniform resource locator, abbreviated URL, also known a web address, is a specific character string that constitutes a reference to a resource. In most web browsers, the URL of a web page is displayed on top inside an address bar.</p>
PC	<p>Personal Computer</p>
XML	<p>Extensible Markup Language</p> <p>Is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.</p>
HTTP	<p>The Hypertext Transfer Protocol</p> <p>Is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.</p>

CPU	Central Processing Unit
JVM	Java Virtual Machine Is a virtual machine that can execute Java byte code. It is the code execution component of the Java platform.
OS	Operating System
API	Application Programming Interface Specifies how some software components should interact with each other
SQL	Structured Query Language Is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS).

LIST OF FIGURES

FIGURE 1 Conceptual Design – Best Sale Application.....	13
FIGURE 2 Polling Conceptual Diagram-1.....	16
FIGURE 3 Polling Conceptual Diagram-2.....	17
FIGURE 4 Thread Lifecycle Diagram.....	19
FIGURE 5 Thread Stages Conceptual Diagram.....	20
FIGURE 6 Thread Polling Conceptual Diagram.....	22
FIGURE 7 Content Management – Login Page.....	26
FIGURE 8 Content Management – Welcome Page.....	27
FIGURE 9 Content Management – Search Page.....	28
FIGURE 10 Content Management – Health Check Page.....	29
FIGURE 11 Content Management – Statistics Page.....	30
FIGURE 12 Content Management – Add URL Page.....	31
FIGURE 13 Content Management – Add URL Failure Page.....	32
FIGURE 14 Content Management – Add Login Page.....	32
FIGURE 15 Core Server flow diagram.....	42
FIGURE 16 Content Management – Add URL Failure Page.....	43

LIST OF TABLES

TABLE 1 Database Table – Users.....	39
TABLE 2 Database Table – Products.....	40
TABLE 3 Database Table – Shops.....	40
TABLE 4 Database Table - Products.....	42
TABLE 5 Database Table – Products.....	44
TABLE 6 Database Table – Products.....	47

1 INTRODUCTION

The basic assumption of this 'Server for a Best Sale Application' is that mobile phones and handset devices are widely available among the public.

A special offer for a product/item/goods in a shop is advertised in newspapers. Each shop deliver these papers free of charge at our door steps. Shops/Stores in the city spend money on making these advertisement papers reach our door steps. Most of the time a general customer is not interested in reading all the sales advertisement every day. The general outcome is the wastage of paper and is the actual intention of making the customer aware of discounts and offer is lost. If a customer is interested in purchasing a particular product, he/she needs search through all the advertisement papers of each store/shop to find the right bargain price. In another instance the customer is at a shop and believes that the bargain price provided there is not the best, he/she might want to know about the bargain price of the same product sold in a different shop nearby. The customer needs to physically travel to the other shop and compare the prices. In this thesis I propose a practical solution to solve this problem.

The advertisements will be available on the mobile handsets and can be accessed at any point of time. The customer can subscribe to a shop and monitor its product and advertisements. The updates from stores will be listed by cities. The content will be provided in a standard XML format that could be used to create an application for Mobile Phones, Tablets and Desktop PCs.

Practical Issues Resolved

- 1) Usage of application can help to reduce the wastage of paper
- 2) Shops/Stores can reduce the advertisement cost and prevent postal charges
- 3) Shops/Stores need to update their website RSS content with the discounts and offer sales. Server for a Best Sale Application will retrieve the content periodically.
- 4) Shops/Stores will be facilitated with the statistics count of number of customer views on their offer
- 5) Shops/Stores will be able to view the offer and sale of their competitor in business

- 6) Shops/Stores can view the statistics of all the Shops/Stores in the locality, which could help them to rate their own offers and discount prices.
- 7) General customer can save time checking through all the advertisement papers to see the best bargain price of a particular product they intent to purchase
- 8) General customer will have all the offers and sale information handy

2 CONCEPTUAL DESIGN – A BEST SALE APPLICATION

The conceptual design (figure 1) is based on the assumption that the advertisements from shop's website pertaining to best offers and discounts are displayed as RSS feed contents. This RSS content is polled regularly by the Server for a Best Sale Application and the new content is saved to the database. The Core Server, explained in detail in further sections of the thesis, will process the database content. The Core Server provides the best offers and discounts content as Rest API endpoints, which are available as a public content in the Internet. The application developers can make use of this Rest API content to publish them to end users and customers. The applications can be developed on any mobile platforms or as a web application. Mobile platforms that are popular today are iOS, Android, Windows and Nokia SDK. Web applications can be accessed from PCs and tablets.

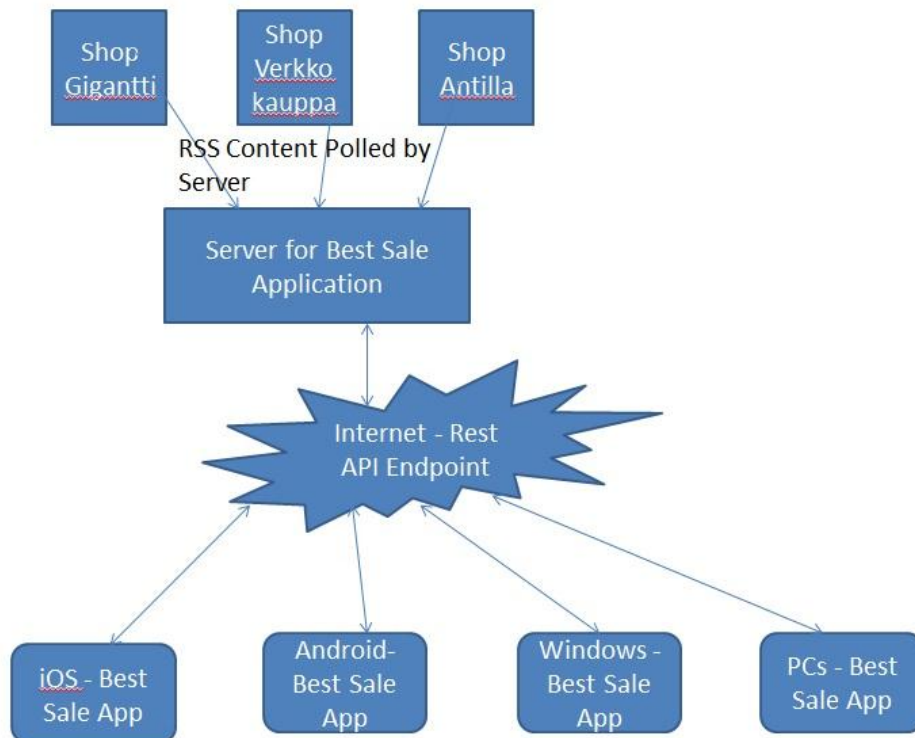


FIGURE 1 Conceptual Design – Best Sale Application

3 UI – BEST SALE APPLICATION

UI – Mobile Advertisement Application is a valuable solution that helps in advertising special offers at a faster pace and prevents any wastage of paper. The application will also save postal charges. The application in mobile will be listing the various (current) special offers of different stores categorized by cities/places.

Mobile – Advertisement Application can be created for various mobile platforms such as Symbian S40, Symbian S60, Meego, Nokia Asha SDK, Windows, iOS and Android. Apart from this we can create an application for Tablets and PCs. It can be used as follows:

- UI design should be done based on a display specification to best fit the screen. (e.g. Mobile, Tablets and Desktop Pcs)
- The application must be made free to improve the acceptance of a general customer
- When opening the application the user is prompted to select the **Region** (for e.g. Uusimaa, Pirkanmaa or Lappi)
- Now the various cities available in the selected region are displayed. The user can select the **City** (e.g. Pohjanmaa region on selection list the cities like Oulu, Oulun Salo, Kempele)
- When selecting the specific cities, various shops will be listed. Now the user can select a shop based on his/her choice, (e.g. when selecting Oulu, shops like Stockman, Gigantti, Verkokauppa will be listed)
- The user must be provided an option to unselect shops from the list displayed.
- Now the user can select the shop of his/her choice and view the various special offers available in it
- Back, Home and Exit icons will be available for a Mobile Application to make navigation easier

The UI- Best Sale Application is explained in depth in the Thesis paper of Chittu Gnanamoorthy Balakumar titled “Best Sale Mobile Application Development”.

4 SERVER – BEST SALE APPLICATION

The Server for a Best Sale Application incorporates components for managing the shops/stores, retrieving the offer or sale information from websites and providing this information in a standard XML format as Rest endpoints. The shops/stores need to update their offer and sale information on their website. The Server for a Best Sale Application will be managed by using the Content Management Portal (explained below). The Content Management Portal helps the administrator of Server for a Best Sale Application to add/administrate the shops in every. The shops/store owners will have a user login to the Content Management Portal where they can check the statistics of the customer view. The Core Server for a Best Sale Application will continuously retrieve information from the RSS link of each registered shop. This information will be stored in the database and made available using memcache. A user request received by the Core Server will be responded a standard XML format, agreed upon with the Application developer. The user request here corresponds to HTTP (Hyper Text Transmission Protocol) received from various devices like Mobile Application, Tablets and Desktop PCs.

4.1 Used Technologies

In the development of a Best Sale Application server many technologies were used. The following list covers the most important technologies.

Java: The core server and polling technique was implemented using Java. The application was developed using a Struts framework.

J2EE: The Content Management Portal and the Rest API endpoints were implemented using the J2EE technique. A Model-View-Controller architecture was used.

JQuery: JQuery is an open source library that was used in this application. JQuery simplifies the usage of JavaScript.

JavaScript: The Content Management Portal and the page validations were implemented using JavaScript.

MySQL: MySQL is an open source database, which was used to create database tables for the application. The content was saved and retrieved from the database using the MySQL queries.

Tomcat: Tomcat is a servlet container and web server used for the Best Sale Application server.

Netbeans: The integrated development environment used for the development of the application is Netbeans.

Git: The source code was stored and maintained using the GIT open source repository.

4.2 Core – Server for a Best Sale Application

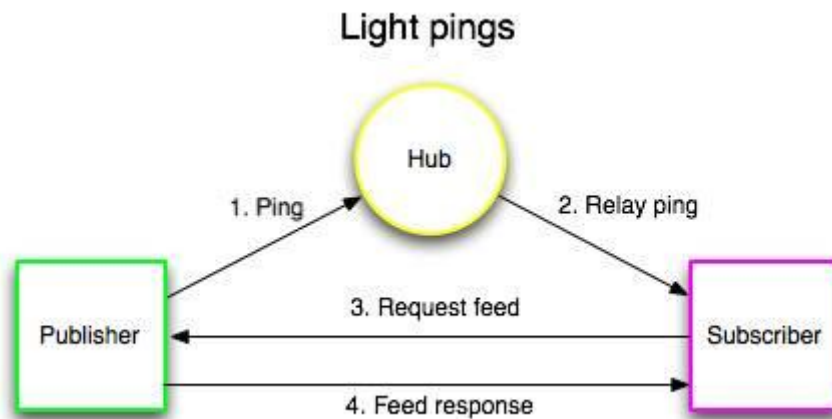
The Core-Server of the Best Sale Application forms a vital component of the paper. The Core-Server was developed based on a polling concept. The content provided by the shop website URL must be observing RSS standards. These concepts were explained to create a better understanding of the Core-Server methodology.

4.2.1 Polling Concept

Polling/Pulling refers to a continuous data synchronizing using a periodic call. In other words this means actively sampling the status of an external device. Pull coding or client pull is a style of network communication where the initial request for data originates from the client, and then is responded to by the server. In real life situation though, usually, customers will look for a site and visit only if it provides helpful and attractive contents and display. Pull requests form the foundation of network computing, where many clients (Core-Server of Best Sale Application) request data from centralised servers (website URLs of each shop). Pull is used extensively on the Internet for HTTP page requests from websites. In Polling synchronizing is done using a busy-wait technique. It can be explained with a simple example: When an I/O operation needs to be performed; the client keeps contacting the server periodically to identify the status. Polling in general refers to a technique where the client calls the server periodically to identify data. If the data is present then the content is retrieved successfully. The time period between a request from

a client to a server (called the polling from here on) is determined by the configuration of the backend processes.

Most web feeds, such as RSS are technically pulled by the client. With RSS, the user's RSS reader polls the server periodically for new content and the server does not send information to the client unrequested, shown in figure 2.



*FIGURE 2 Polling Conceptual Diagram-1
(Java Tutorial , 2014, date of retrieval 06.03.2014)*

The concept of Pull Technology was implemented in the Core Server of Best Sale Application. The Admin uses the Content Management Portal to save/enter each website URL of the shops registering with us. The Core-Server checks the validity of these websites. Here we mean that each website URL has a standard RSS content. Once the verification is completed, the website URL is added to the database. Each shop website URL is considered as a server and information is Pulled using threads (explained later).The Core server will have one thread running created and actively running for each shop website URL. Each thread will be sending an HTTP request to the server i.e. shop website URL once a day. The Core Server for a Best Sale Application was created using the Threading concept in Java with a periodic duration of 24hours.

Some assumptions need to be made to make sure that the load on the server is under control. We presume that every shop available in Finland will update new offers and sale once a day. So in order to prevent the number of polling attempts to get new offer information, we are setting the

time period as 24hours. With every attempt of Pulling we expect to read a new content and update our database, shown in figure 3.

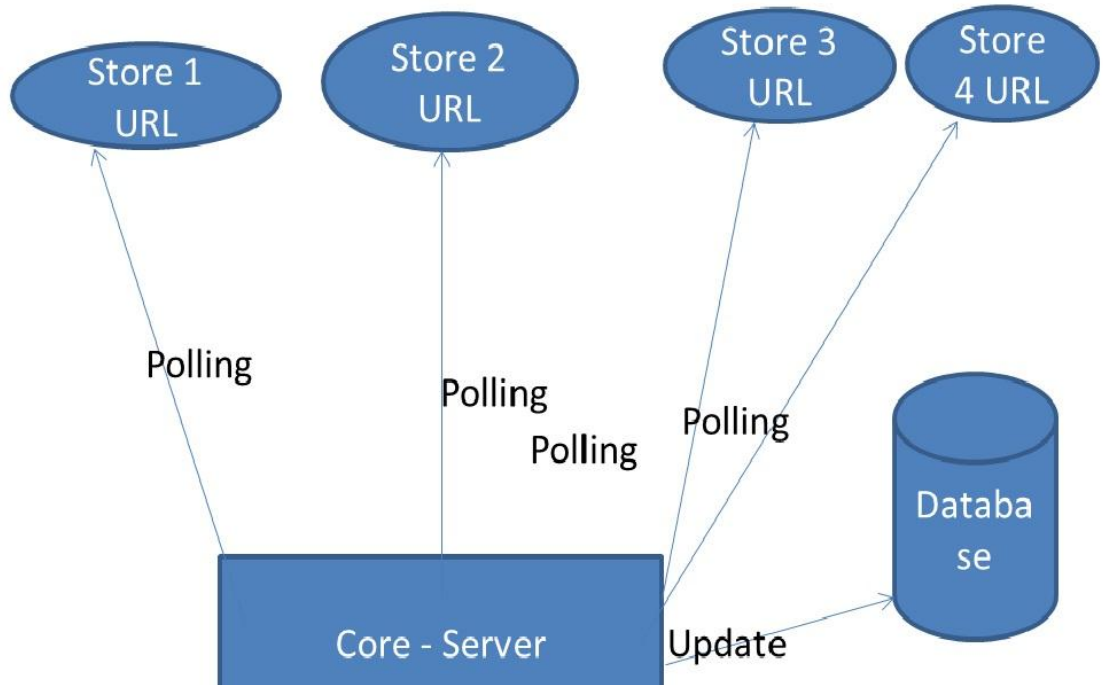


FIGURE 3 Polling Conceptual Diagram-2

4.2.2 Threads

4.2.2.1 Concurrency

Computer systems are capable of doing more than one task at a time. We can type some documents, while the music file keeps playing in the background and a document could be printed at the same time. For example, that streaming audio application must simultaneously read the digital audio off the network, decompress it, manage playback, and update its display. Within an application like word processor the response for keyboard should be simultaneous. Software that can do such things is known as concurrent software.

4.2.2.2 Java Threads

Java provides a built-in support for a multithreaded programming. Java Threads concept is clearly explained as follows in Java Tutorial (2014, date of retrieval 06.03.2014)

A multithreaded program contains two or more parts that can run concurrently. Each part of such a program is called a thread, and each thread defines a separate path of execution. Multithreading is a specialized form of multitasking. Multithreading requires less overhead than multitasking processing.

I need to define another term related to threads: process: A process consists of the memory space allocated by the operating system that can contain one or more threads. A thread cannot exist on its own; it must be a part of a process. A process remains running until all of the non-daemon threads are done executing. Multithreading enables you to write very efficient programs that make the maximum use of the CPU, because idle time can be kept to a minimum.

4.2.2.3 Life Cycle

A thread goes through various stages in its life cycle. For example, a thread is born and started then it runs, and dies. Following diagram shows a complete life cycle of a thread, shown in figure 4. (Tutorials Point, 2014, date of retrieval 06.03.2014)

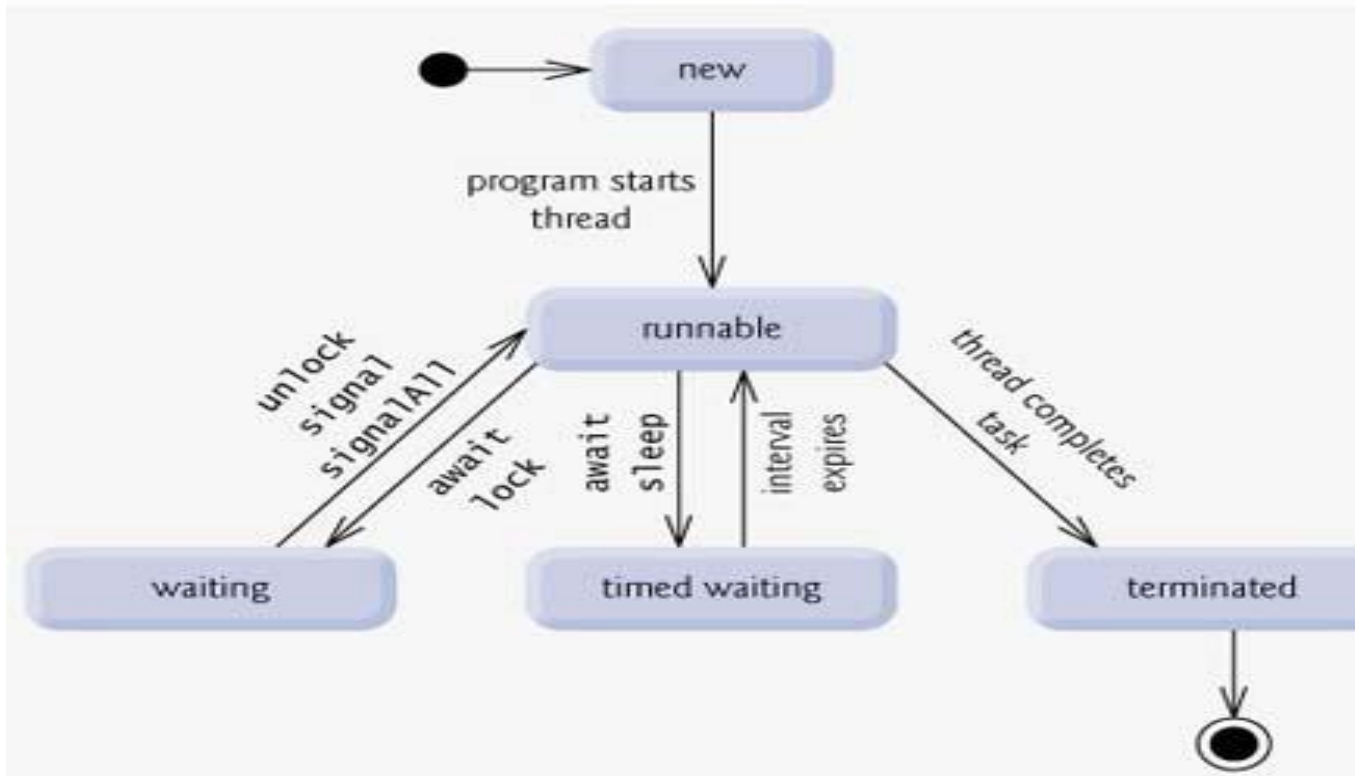


FIGURE 4 Thread LifeCycle Diagram
(Tutorials Point, 2014, date of retrieval 06.03.2014)

The above-mentioned stages are explained here and shown in figure 4:

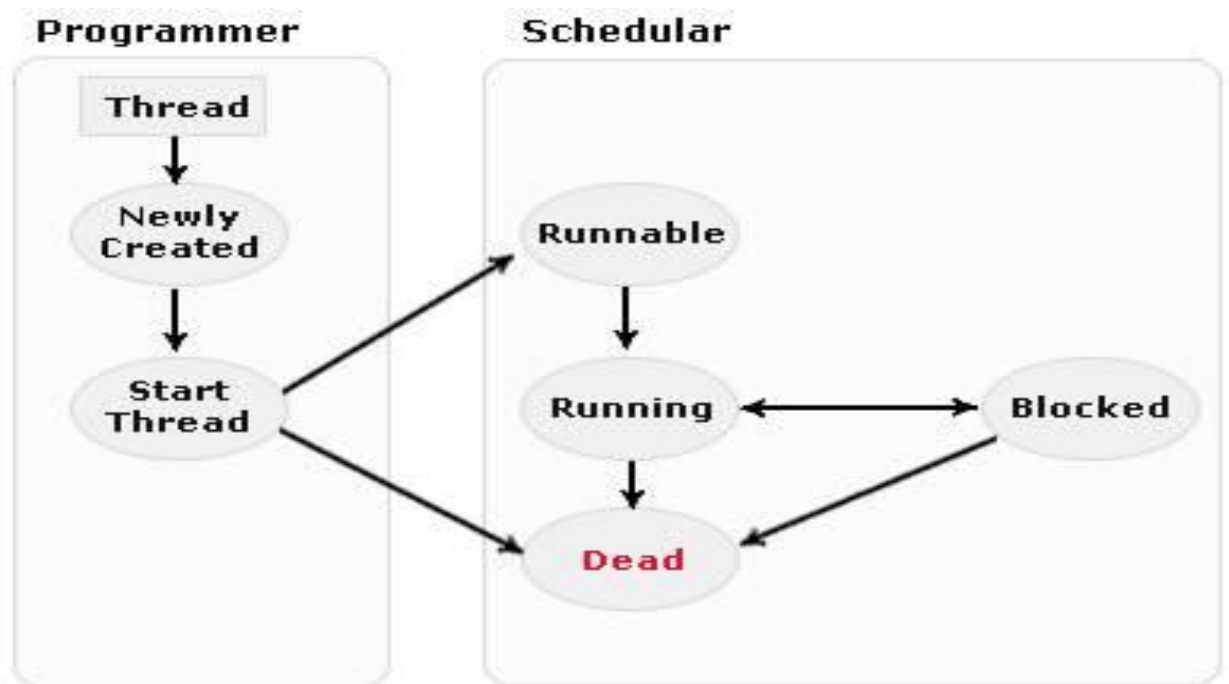
New: A new thread begins its life cycle in a new state. It remains in this state until the program starts the thread. It is also referred to as a born thread.

Runnable: After a newly born thread is started, the thread becomes runnable. A thread in this state is considered to be executing its task.

Waiting: Sometimes, a thread transitions to the waiting state while the thread waits for another thread to perform a task. A thread transitions back to the runnable state only when another thread signals the waiting thread to continue executing.

Timed waiting: A runnable thread can enter the timed waiting state for a specified interval of time. A thread in this state transition back to the runnable state when that time interval expires or when the event it is waiting for occurs.

Terminated: A runnable thread enters the terminated state when it completes its task or otherwise terminates.



*FIGURE 5 Thread Stages Conceptual Diagram
(Tutorials Point, 2014, date of retrieval 06.03.2014)*

4.2.2.4 Thread Pooling

Thread pooling refers to a technique where a pool of worker threads is created and managed by the application. When a new job arrives, instead of creating a new thread to service it, it is queued by the thread-pool manager and dispatched later to one of the available worker threads. The thread-pool manager manages the number of active worker threads based on available resources as well as load considerations, adding new threads to the pool or freeing some worker threads in response to the number of outstanding requests. The primary goals of thread pooling are managing the number of active threads in the system and reducing the overhead of creating new threads by reusing threads from a pool. Thread Pooling is clearly explained as follows in Java Tutorial (2014, date of retrieval 06.03.2014)

The primary argument in favour of managing the number of active threads in the system is: threads have a memory overhead since each one needs a certain amount of memory for its stack. Threads also add scheduling overhead, since the scheduler's work increases as the number of threads increases. Depending on the implementation of the Java Virtual Machine, each Java thread on certain operating systems may correspond to an OS thread, making Java threads extremely heavyweight, and may limit the total number of active threads that the JVM is allowed to create.

To be clear, I'm not saying you don't need to manage the number of active threads in a system. After all, the benefits of multithreading do have diminishing returns once the number of threads contending for the available CPUs increases. If a server can process only about 1,000 simultaneous requests, it doesn't help to dispatch each incoming request as it's made. Often the requests must be queued and processed at a controlled rate to maintain the number of active requests below the server threshold. A common mistake, however, is to assume that dispatching queued requests automatically calls for the reuse of threads from a thread pool. Dispatching a request to a new thread and letting the thread die once the request is serviced achieves the same effect on managing the number of active threads in the system.

Thread creation also has an overhead that can be higher in many cases than the overhead of managing a thread pool. While the argument still applies, the relative performance impact has changed significantly over the years. The newer JVM implementations are optimized for creating threads; most use a combination of user-level threads (known as green threads) as well as system-level threads (or OS threads) to make creating threads much less expensive than in earlier implementations, shown in figure 6.

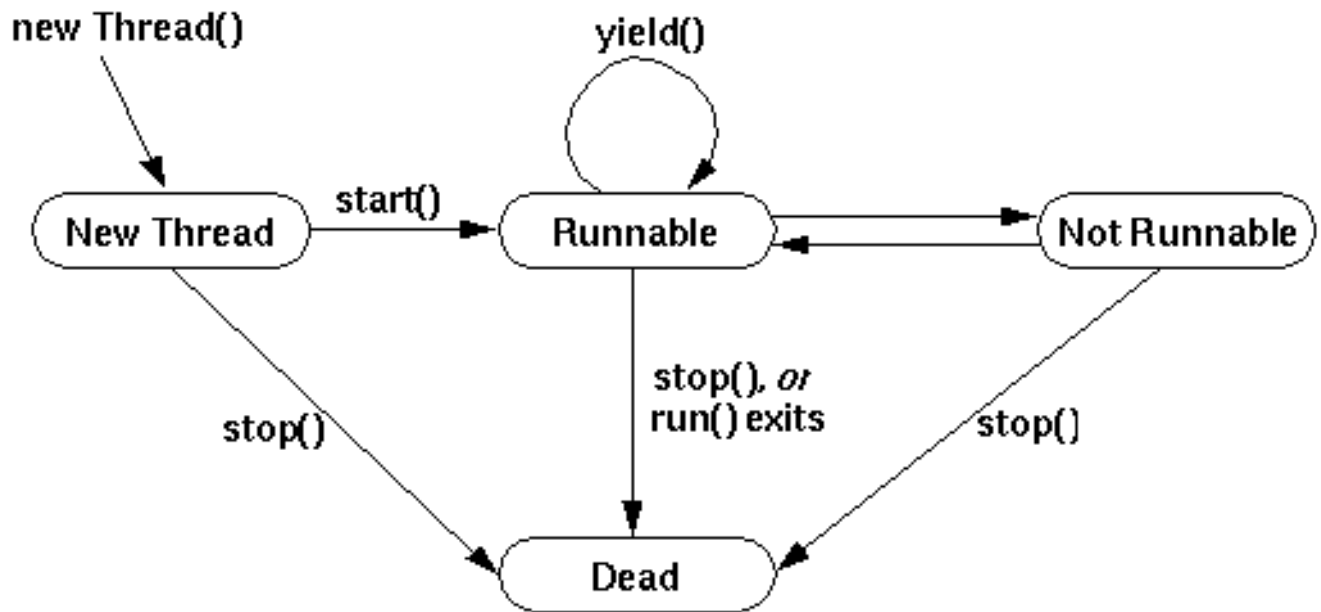


FIGURE 6 Thread Polling Conceptual Diagram
 (Java Tutorial , 2014, date of retrieval 06.03.2014)

4.2.3 RSS Content

RSS stands for Really Simple Syndication or Rich Site Summary however it is often referred to as the feed or news feed. In the early days, if you wanted to keep track of updates on your favourite website, then you had to bookmark it in your browser and manually visit it frequently to see if there were any updates. Wouldn't it be nice if you could tell your favourite websites to let you know every time they update? Well, this is exactly what RSS does. RSS allows users to keep track of their favourite websites without having to manually visit the website each time.

RSS allowed publishers to automatically syndicate their content, so that people can read it in their mailboxes, feed reader and/or other devices. RSS is basically an structured XML document that includes full or summarized text along with other metadata such as published date, author name, etc. RSS feeds can be read by using programs called "RSS reader", "feed reader", or "aggregator" which can be web-based, desktop-based, and even mobile-based. Here is what an RSS feed looks like

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<rss version="2.0">

<channel>

```

```
<title>W3Schools Home Page</title>
<link>http://www.w3schools.com</link>
<description>Free web building tutorials</description>
<item>
  <title>RSS Tutorial</title>
  <link>http://www.w3schools.com/rss</link>
  <description>New RSS tutorial on W3Schools</description>
</item>
<item>
  <title>XML Tutorial</title>
  <link>http://www.w3schools.com/xml</link>
  <description>New XML tutorial on W3Schools</description>
</item>
</channel>
</rss>
```

The RSS basic concept is clearly explained as follows in Wordpress Beginners (2014, date of retrieval 06.03.2014)

RSS makes it easier for other people to subscribe to your content. Using an RSS feed, they can receive updates from your blog or website directly in their web browsers (except Google Chrome), feed readers, desktop applications, and many other devices and services. Subscribers don't have to remember your website URL because they can bring the content into a central location where they like to read all of their websites at once. Your content becomes portable and your subscribers can even take it with them for offline reading.

RSS feeds also help build and nurture a loyal following around your website and content. RSS subscribers are more likely to share your content around their social circles and spread the word about your website. By using RSS feeds, you can subscribe to your favourite websites and read them all at one place in your feed reader. As a content publisher, it helps you quickly scan through your favourite news and blog sites. It also enables you to sort your reading lists into categories and groups. By using RSS efficiently, you can really maximize your reading time.

4.2.3.1 Benefits of RSS

RSS is taking off so quickly because people like it. RSS is easy to use and it has advantages for a publisher as well as for a subscriber. Here are the advantages of RSS for subscribers as well as for publishers listed out. The advantages of RSS for subscribers and publishers are clearly explained as follows in Tutorials Point (2014, date of retrieval 06.03.2014)

RSS Advantages for Subscribers

RSS subscribers are the people who subscribe to read a published feed. Here are some of the advantages of RSS feeds for subscribers.

All News at one Place: *You can subscribe to multiple news groups and then you can customize your reader to have all news on a single page. It will save you a lot of time.*

News when you want it: *Rather than waiting for an e-mail, you go to your RSS reader when you want to read a news. Furthermore, RSS feeds display more quickly than information on Web sites, and you can read them offline if you prefer.*

Get only the news you want: *RSS feed comes in the form of headlines and a brief description so you can easily scan the headlines and click only those stories that interest you.*

Freedom from e-mail overload: *You are not going to get any email for any news or blog update. You just go to your reader and you will find updated news or blog automatically whenever there is a change on RSS server.*

Easy republishing: *You may be both a subscriber and a publisher. For example, you may have a Web site that collects news from various other sites and then republishes it. RSS allows you to easily capture that news and display it on your site.*

RSS Advantages for Publishers

RSS publishers are the people who publish their content through RSS feed. I would suggest you to use RSS if you want to get your message out and easily and if you want people to see what you publish, and you want your news to bring people back to your site. Here are some of the advantages of RSS if you publish on the Web:

Easier publishing: RSS is really simple publishing. You don't have to maintain a database of subscribers to send your information to them, instead they will access your feed using a reader and will get updated content automatically.

A simpler writing process: If you have a new content on your Web site, you need only write a RSS feed in the form of titles and short descriptions, and link back to your site.

An improved relationship with your subscribers: Because people subscribe from their side, they don't feel as if you are pushing your content on them.

The assurance of reaching your subscribers: RSS is not subject to spam filters, your subscribers get the feeds which they subscribe to and nothing more.

Links back to your site: RSS feeds always include links back to a Web site. This will increase lot of traffic towards your website.

Relevance and timeliness: Your subscribers always have the latest information from your site.

5 CONTENT MANAGEMENT PORTAL

Stores selling products will display their products and offers on their website. The stores website is where the information is collected from. The Best Sale Application – Server admin will enter every stores website into the Web Portal manually. This helps to prevent in-appropriate content and also prevents websites which are not active from being included. The information available on each website is parsed and details are categorized based on location, duration of offer, products and discount type. Location pertains to the city/state/country information. Duration pertains to start date and end date of this offer. The Portal can list the stores based on location (city/state/country) and based on product type. After the duration i.e. the product with an offer will be inactive. The Admin has the delete privileges and is responsible for cleaning the database using the web portal.

Guest login for the web portal is also provided for the Store owners. The view presented to the guest logins is restricted to view only. They could search the stores based on a location. Information and view of the various products will be displayed. The guest has view rights to all the stores and all the products of each store. This will help them understand the usability of our Best Sale Application.

Every entry in the server denoting the offer is saved based on the product. One entry for each product of a store. This entry provides information about a shop name, location, start date of offer, end date of offer, discount type and status. Status denotes information whether the offer is active or inactive. The Content Management Portal and the components are explained below.

5.1 Login Page

The Admin login is added/created by making an entry into the database. The guest login will be created for every shop owner so that they can check the statistics of the items available and number of views. The user login page is shown in figure 7.

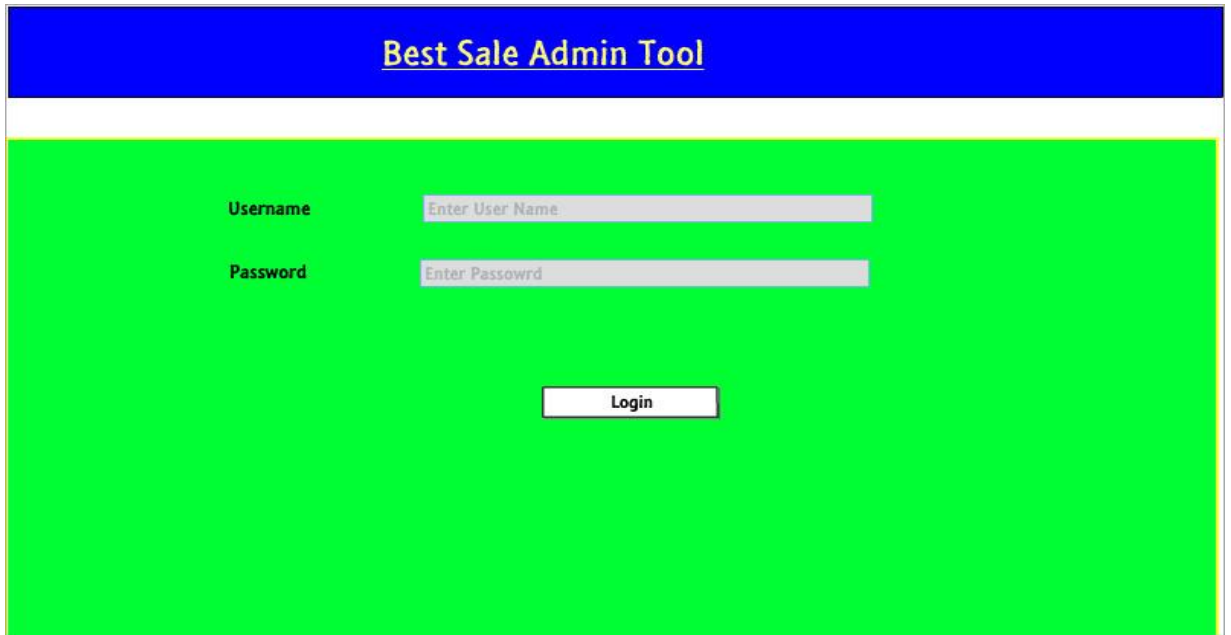


FIGURE 7 Content Management – Login Page

The figure 7 above shows the Best Sale Application – Server’s Content Management Portal. The Admin and the Guest would have the same view, but the Guest login will have limited privileges. Listed below are the privileges of login types.

Admin Login: Add Login, Health Check, Statistics, Add URL and Search

Guest Login: Search, Stats

On a successful Login the welcome page is displayed. The privileges are listed in the menu tab (on the left) i.e. (Admin/User), shown in figure 8

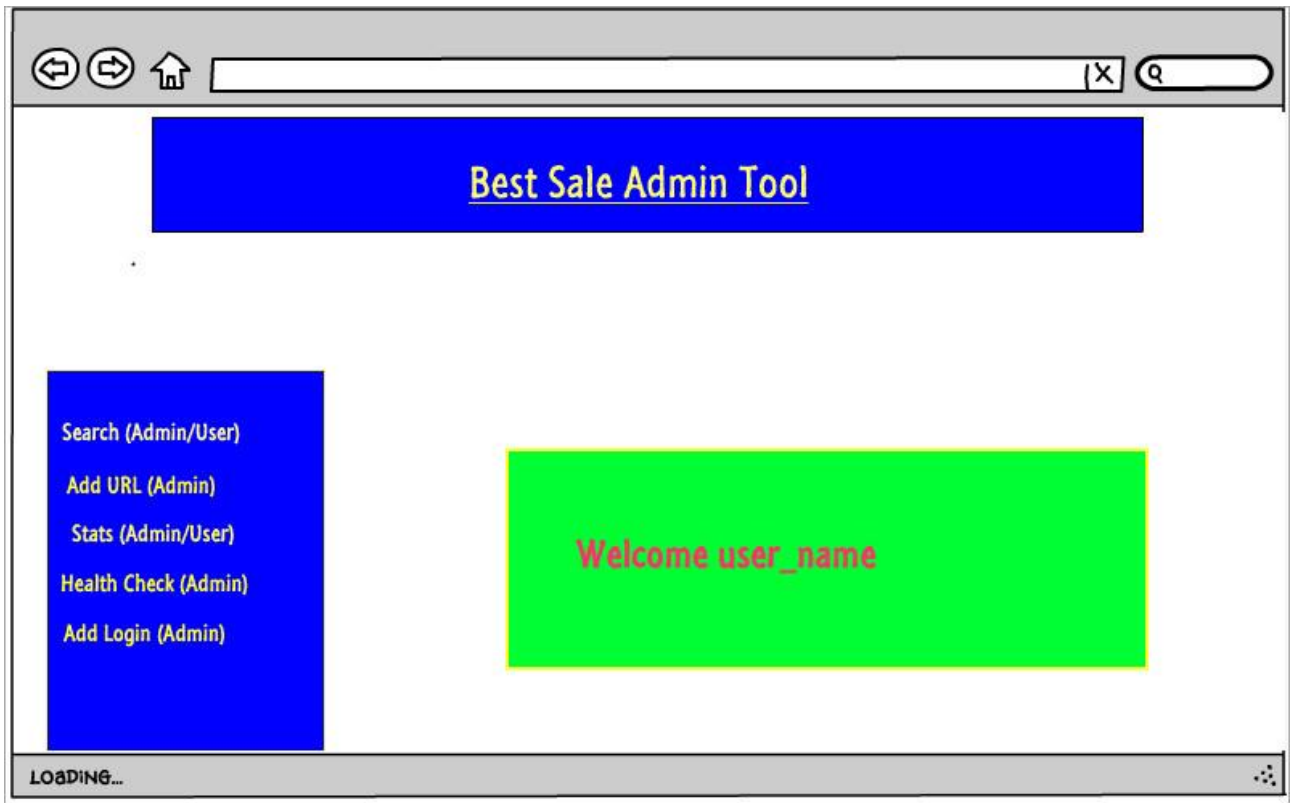


FIGURE 8 Content Management – Welcome Page

5.2 Search Page

The Search Page lists the offer products available in each city. The Search Page is visible for the Admin and the User login. The items can be searched and verified based on city and country. Initially the number of shops/stores available would gradually increase based on the usability of the application. Region is another parameter which could be used, at a later stage, to categorize the stores. Search Page of the Content Management Portal is, shown in figure 9.



FIGURE 9 Content Management – Search Page

5.3 Health Check Page

The Health Check Page is visible to the Admin Login. This page helps the Admin monitor the Stores/Shops added to the Portal. The basic idea is to help the Admin identify the Shop/Store URLs that are inactive with no data. This would help the admin to check the websites accordingly to verify the content. Figure 10 shows the Health Check Page. The methodology used identify the Website URLs that are not active is explained in the core server section of this thesis.

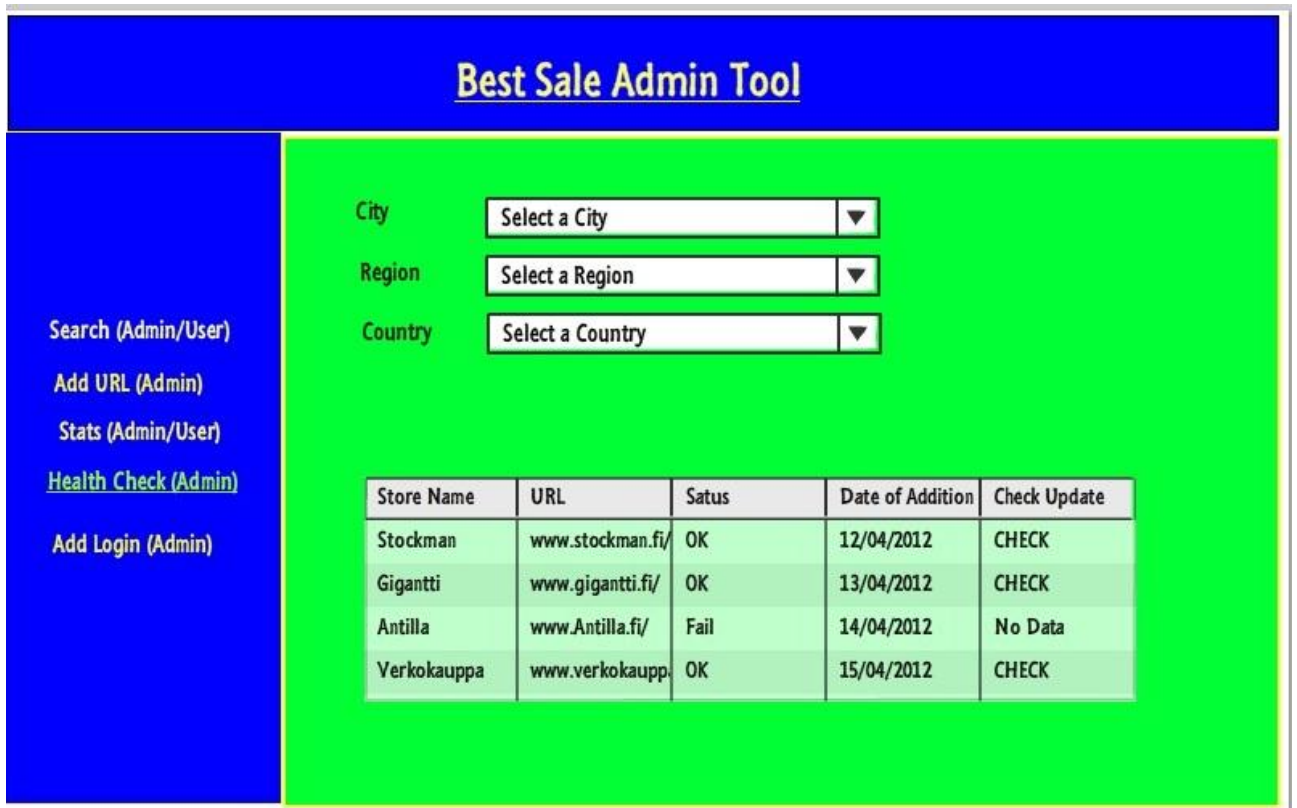


FIGURE 10 Content Management – Health Check Page

5.4 Stats Page

The Stats page lists the statistics of the user view. This page is visible for the User and the Admin login. The page displays the total user request received per shop. The Statistics is saved in the database. The Core Server logs these request for every request from the User. Picture of the Stats page is, shown in figure 11.



FIGURE 11 Content Management – Statistics Page

5.5 Add URL Page

The Add URL page is in Admin Login view. Here the Admin adds each Shop/Store website URL manually. The data entered here is verified by the Core Server (explained later in this paper). The Store data is stored in the database. The RSS content in the website is parsed and identified by the Core Server, shown in figure 12.

Best Sale Admin Tool

Search (Admin/User)
[Add URL \(Admin\)](#)
Stats (Admin/User)
Health Check (Admin)
Add Login (Admin)

Add URL Add URL here..

City Select a City ▼

Region Select a Region ▼

Country Select a Country ▼

Add

FIGURE 12 Content Management – Add URL Page

The Website URL added here is verified by the Core-Server explained below. If the shop website does not contain RSS content for the Core-Server to poll, then the URL is rejected. This helps the admin to recheck the website manually for RSS support, shown in figure 13.



FIGURE 13 Content Management – Add URL Failure Page

5.6 Add Login Page

The Add Login Page helps the Admin to create user logins for the guest i.e. Shop/Store Owners. The Add Login Page has Admin Login View, shown in figure 14



FIGURE 14 Content Management – Add Login Page

6 REST API ENDPOINTS

The Website URLs of shops/stores is stored in the database by the Admin using the Content Management Portal. These URLs are verified by the Core-Server (explained in Core-Server section). The RSS content is parsed to identify the information available in each store/shop. The Mobile Application developer needs to be provided with the XML content in a standard format which is uniform. The RSS content available in the various Store/Shop websites is not expected to be in the same format. The Core of Best Sale Application Server is expected to parse the content and identify the useful information. This is then saved to the database. Every product of each shopforms one row in the database (explained in the Database section). This information is retrieved and presented to the Mobile Application Developer as Rest API Endpoints. This Rest API Endpoints could be used to create application on any platform such as Android, IOS, Windows and Nokia Asha SDK

Below are couple of RSS content available at two different websites. The content shows that the format of presenting information is different in either case.

RSS Content Type 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" media="screen" href="/~d/styles/rss2full.xsl"?><?xml-stylesheet
type="text/css" media="screen" href="http://rss.antilla.com/~d/styles/itemcontent.css"?><rss
xmlns:media="http://search.yahoo.com/mrss/" version="2.0">
<channel>
<title>antilla.com - Top Stories</title>
<link>http://edition.antilla.com/index.html?eref=edition</link>
<description>antilla.com delivers up-to-the-minute news and information on the latest top stories, weather,
entertainment, politics and more.</description>
<language>en-US</language>
<copyright>Copyright 2013 Cable News Network LP, LLLP.</copyright>
<pubDate>Wed, 13 Nov 2013 03:56:14 EST</pubDate>
<ttl>10</ttl>
<image>
<title>antilla.com - Top Stories</title>
<link>http://edition.antilla.com/index.html?eref=edition</link>
<url>http://i.cdn.turner.com/antilla/e/img/1.0/logo/antilla.logo.rss.gif</url>
<width>144</width>
<height>33</height>
```

```

<description>antilla.com offers wide variety of products such as household items</description>
</image>
<atom10:link xmlns:atom10="http://www.w3.org/2005/Atom" rel="self" type="application/rss+xml"
href="http://rss.antilla.com/rss/edition" /><feedburner:info
xmlns:feedburner="http://rssnamespace.org/feedburner/ext/1.0" uri="rss/edition" /><atom10:link
xmlns:atom10="http://www.w3.org/2005/Atom" rel="hub" href="http://pubsubhubbub.appspot.com/"
/><item><title>Curtains</title><guid>http://edition.antilla.com/2013/11/11/justice/iranian-underground-
band-members-shot-dead/index.html</guid><link>http://edition.antilla.com/2013/11/11/justice/iranian-
underground-band-members-shot-dead/index.html?eref=edition</link><description> Offer - Flat 30%
Discounts </description><pubDate>Wed, 13 Nov 2013 03:04:47
EST</pubDate></item><item><title>Winter
Jerkins</title><guid>http://edition.antilla.com/2013/11/12/world/meast/israel-
liberman/index.html</guid><link>http://edition.antilla.com/2013/11/12/world/meast/israel-
liberman/index.html?eref=edition</link><description> Jack and Jones offer - 10%
</description><pubDate>Tue, 12 Nov 2013 12:02:37 EST</pubDate></item>
</channel>
</rss>

```

RSS Content Type 2:

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet title="XSL_formatting" type="text/xsl" href="/shared/bsp/xsl/rss/nolsol.xsl"?>

<rss xmlns:media="http://search.yahoo.com/mrss/" xmlns:atom="http://www.w3.org/2005/Atom"
version="2.0">
<channel>
<title>Kmarket News - Home</title>
<link>http://www.Kmarket.co.uk/news/#sa-ns_mchannel=rss&ns_source=PublicRSS20-sa</link>
<description>The latest stories from the Home section of the Kmarket News web site.</description>
<language>en-gb</language>
<lastBuildDate>Wed, 13 Nov 2013 08:56:04 GMT</lastBuildDate>
<copyright>Copyright: (C) KMarket see http://news.Kmarket.co.uk/2/hi/help/rss/4498287.stm for terms
and conditions of reuse.</copyright>
<image>
<url>http://news.Kmarketimg.co.uk/nol/shared/img/Kmarket_news_120x60.gif</url>
<title>Kmarket Items - Home</title>
<link>http://www.Kmarket.co.uk/news/#sa-ns_mchannel=rss&ns_source=PublicRSS20-sa</link>
<width>120</width>
<height>60</height>
</image>
<ttl>15</ttl>
<atom:link href="http://feeds.Kmarketi.co.uk/news/rss.xml" rel="self" type="application/rss+xml"/>
<item>
<title> Fruits </title>
<description> Fruits offer - Flat 40% Discounts </description>
<link>http://www.Kmarket.co.uk/news/health-24914385#sa-
ns_mchannel=rss&ns_source=PublicRSS20-sa</link>

```

```

<guid isPermaLink="false">http://www.Kmarket.co.uk/news/health-24914385</guid>
<pubDate>Wed, 13 Nov 2013 08:04:23 GMT</pubDate>
  <media:thumbnail width="66" height="49"
url="http://news.Kmarketimg.co.uk/media/images/71068000/jpg/_71068405_waiting.jpg"/>
  <media:thumbnail width="144" height="81"
url="http://news.Kmarketimg.co.uk/media/images/71068000/jpg/_71068406_waiting.jpg"/>
</item>
</channel>
</rss>

```

6.1 Rest Endpoint – Shops List API

The Shops List API is one of the two Rest APIs communicated to the Best Sale Application. The information presented by the Shop/Stores Website is presented in this format. Here the list of shops that are registered with Best Sale Application are listed. The shops List is presented categorized based on City. This will be the initial page on the Mobile Application. The Shops List API provide information about every shop (registered) available in a City. Each entry specifies the Shop name, Title and the XML URL. The XML URL such as Biltelma_Jyvaskylä or Vekkokauppa_Oulu helps the mobile application developer identify the URL address where the items (offers/discounts) list could be retrieved.

If the mobile application makes the request,

http://Best_Sale_Application:8080/BestSaleApp

the server will respond with the following XML description.

```

<?xml version="1.0" encoding="UTF-8"?>
<opml version="1.0">
  <head>
    <title>Finland Local Store Sale Deals</title>
  </head>
  <body>
    <outline title="Oulu" text="Oulu">
      <outline text="Antila" title="Antila" type="rss" xmlUrl="Antila_Oulu"/>
      <outline text="BudgetSports" title="BudgetSports" type="rss" xmlUrl="BudgetSports_Oulu"/>
      <outline text="Vekkokauppa" title="Verkkokauppa" type="rss" xmlUrl="Vekkokauppa_Oulu"/>
      <outline text="Gigantti" title="Gigantti" type="rss" xmlUrl="Gigantti_Oulu"/>
    </outline>

    <outline title="Jyvaskyla" text="Jyvaskyla">
      <outline text="Biltelma" title="Biltelma" type="rss" xmlUrl="Biltelma_Jyvaskyla"/>
      <outline text="InterSports" title="InterSports" type="rss" xmlUrl="InterSports_Jyvaskyla"/>
      <outline text="KMarket" title="KMarket" type="rss" xmlUrl="KMarket_Jyvaskyla"/>
      <outline text="SMarket" title="SMarket" type="rss" xmlUrl="SMarket_Jyvaskyla"/>
    </outline>
  </body>
</opml>

```

```
</outline>
</body>
```

6.2 Rest Endpoint – Items List API

The Items list Rest API Endpoint provides us the list of items (offers /discounts) available in each shop in a city. The Shop List API response helps the Mobile Application Developer to construct the request for the Items List API request. The xmlUrl parameter serves the purpose, e.g. xmlUrl="Antila_Oulu" is used to build a request. The Items List API endpoint listed below, provides a response in a standard XML format communicated with the Mobile Application Developer. Each Item provides information about an offer available in a city store, e.g. Title, Description, Pubdate, StartDate, Enddate and Status. Status helps us to identify the offer validity. New, Started and Ended are the states identifying the duration of the offer/discounts.

If the mobile application makes the request,

<http://Best Sale Application:8080/BestSaleApp/Antila Oulu>

the server will respond with the following XML description

```
<rss xmlns:media="http://search.yahoo.com/mrss/" xmlns:atom="http://www.w3.org/2005/Atom"
version="2.0">
  <channel>
    <title>Antila - Oulu</title>
    <link></link>
    <description>Finland - Oulu Local Store Sale Deals</description>
    <language>en-gb</language>
    <image>
      <url>Image_of_the_store_to_be_provided</url>
      <title>Antila - Oulu</title>
      <link></link>
      <width>120</width>
      <height>60</height>
    </image>
    <ttl>15</ttl>
    <item>
      <title>Curtains</title>
      <description>Offer - Flat 30% Discounts</description>
      <pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
      <startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
      <endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
      <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
      <status>New</status>
    </item>
```

```

<item>
  <title>Winter Jerkins</title>
  <description>Jack and Jones offer - 10%</description>
  <link>url_of_the_product_to_be_provided</link>
  <pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
  <startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
  <endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
  <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
  <status>New</status>
</item>
<item>
  <title>Addidas</title>
  <description>Addidas items - 20% discount</description>
  <link>url_of_the_product_to_be_provided</link>
  <pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
  <startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
  <endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
  <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
  <status>New</status>
</item>
<item>
  <title>Baby Toys</title>
  <description>Baby Toys - 15% discount</description>
  <link>url_of_the_product_to_be_provided</link>
  <pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
  <startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
  <endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
  <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
  <status>New</status>
</item>
<item>
  <title>Antilla Cash Counter</title>
  <description>Take away</description>
  <link>url_of_the_product_to_be_provided</link>
  <pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
  <startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
  <endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
  <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
  <status>New</status>
</item>
</channel>
</rss>

```

7 DATABASE

The open source MySQL 5.5 database is used for this project. The information entered by the Admin using the Content Management Portal is stored in the database. The Core-Sever parses the RSS content on the Website URL and stores the Offer/Item information in the database. Statistics is a vital content stored, which that helps in presenting our Best Sale Application for money generation in future. The Tables and the Data stored in them are listed below.

7.1 Table – Users

This table stores the user account content for accessing the Content management Portal. The rights column specifies whether the user is Admin or a Guest User, as shown in Table 1.

TABLE 1. Database Table - Users

Field	Type	Null	Key	Default	Extra
Name	varchar(20)	YES		NULL	
Username	varchar(50)	YES		NULL	
Password	varchar(100)	YES		NULL	
Rights	varchar(10)	YES		NULL	

7.2 Table – Products

The Products table lists each item available in every shops of the city. The Products table provides information about the Start Date, End Date and the Status of the offer/discount. Shop_Id is the link to the Shops table listed next. The table content is entered by the Core-Server. The Status column identifies whether the offer is New, Started and Ended. Admin can remove the rows (items) when the Status column is marked as Ended. This can be done manually initially and can be replaced by a script running on the database server. The removal of expired items can be automated, as shown in Table 2.

TABLE 2. Database Table - Products

Field	Type	Null	Key	Default	Extra
Product_Name	varchar(20)	YES		NULL	
Description	varchar(100)	YES		NULL	
Shop_id	varchar(20)	YES		NULL	
City	varchar(20)	YES		NULL	
State	varchar(20)	YES		NULL	
Country	varchar(20)	YES		NULL	
Published_date	date	YES		NULL	
Start_Date	Date	YES		NULL	
End_Date	Date	YES		NULL	
Status	varchar(20)	YES		NULL	

7.3 Table – Shops

The Shops table lists the shops available in each city. The Week column identifies the statistics value and the Usage column denotes the number of user request count that is received, as shown in Table 3.

TABLE 3. Database Table - Shops

Field	Type	Null	Key	Default	Extra
Shop_Name	varchar(20)	YES		NULL	
Shop_id	varchar(20)	YES		NULL	
City	varchar(20)	YES		NULL	
State	varchar(20)	YES		NULL	
Country	varchar(20)	YES		NULL	
Usage	Number	YES		NULL	
Week	varchar(10)	YES		NULL	

7.4 Core-Server Execution Logic

The Core-Server logic is explained in this section. Flow diagram is shown in figure 15.

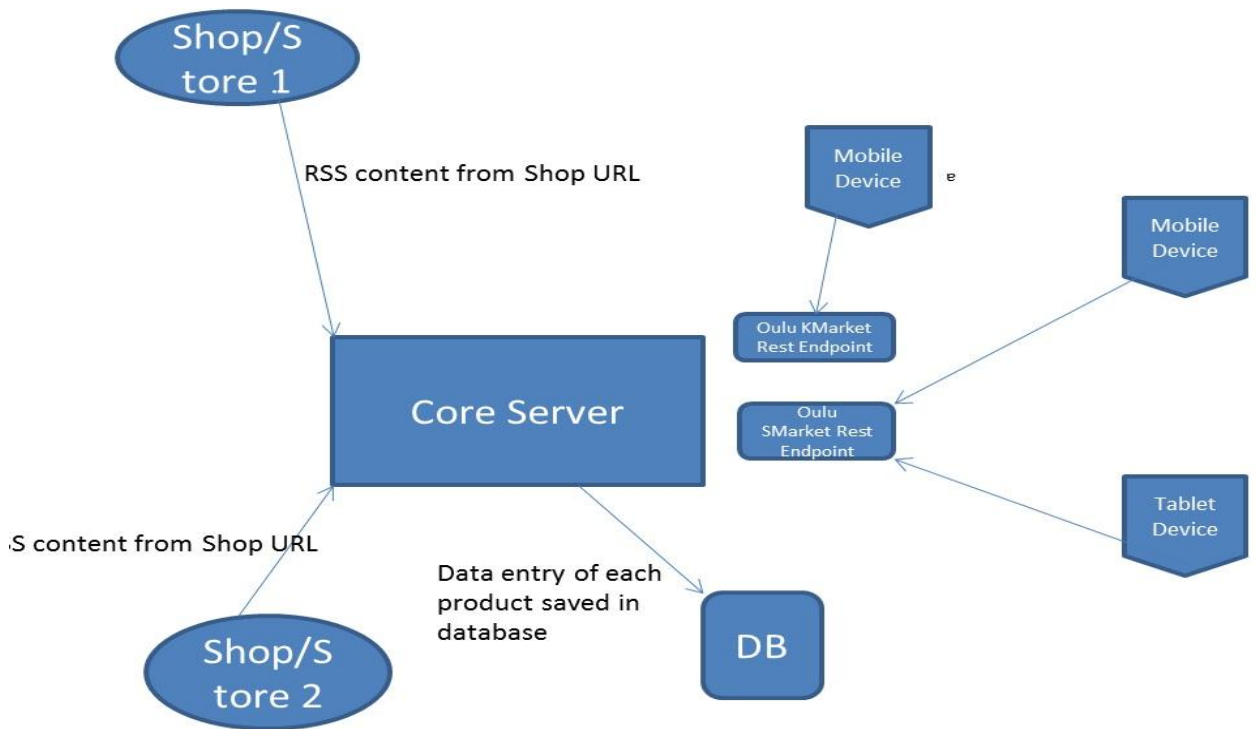


FIGURE 15. Core Server flow diagram

The admin adds each shop website URL to our Content Management Portal. The Core-Servers traverses the URL to identify the RSS link. If the RSS link is not available, then the Core-Server returns error which is reported on the Content Management Portal as “URL Not Responding”, as shown in figure 16



FIGURE 16. Content Management – Add URL Failure Page

The Core-Server adds the RSS link of the website URL to the database. This entry creates a new record in “SHOPS” table. The row indicates the corresponding shop. The “Shop_Id” corresponds to a 6-digit alpha-numeric random number representing the shop, as shown in Table 4.

TABLE 4. Database Table - Products

Shop_Name	Shop_Id	City	State	Country	Usage	Week
KMarket	cbFF1a	Oulu		Finland	0	45

The Core-Server sends a request to the RSS link and in response the RSS content is retrieved. The RSS content pertains to the offer and discount sales available at the store. The content is parsed using the Java XML parser libraries (open source) and the useful discounts information is

stored in the database. The table used for storing the discounts and offer information is "PRODUCTS"

Day 1: RSS content received from the website URL

Let's assume that the RSS content polled by the Best Sale Application's core server on day 1 is as follows,

```
<rss xmlns:media="http://search.yahoo.com/mrss/" xmlns:atom="http://www.w3.org/2005/Atom"
version="2.0">
  <channel>
    <title>KMarket - Oulu</title>
    <link></link>
    <description>Finland - Oulu Local Store Sale Deals</description>
    <language>en-gb</language>
    <image>
      <url>Image_of_the_store_to_be_provided</url>
      <title>KMarket - Oulu</title>
      <link></link>
      <width>120</width>
      <height>60</height>
    </image>
    <ttl>15</ttl>
    <item>
      <title>Fruits</title>
      <description>Fruits offer - Flat 40% Discounts</description>
      <link>url_of_the_product_to_be_provided</link>
      <pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
      <startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
      <endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
      <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
      <status>New</status>
    </item>
    <item>
      <title>ES Products</title>
      <description>ES Products offer</description>
      <link>url_of_the_product_to_be_provided</link>
      <pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
      <startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
      <endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
      <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
      <status>New</status>
    </item>
    <item>
      <title>Meat Discount</title>
      <description>Meat Discount Offer</description>
```

```

<link>url_of_the_product_to_be_provided</link>
<pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
<startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
<endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
<media:thumbnail width="66" height="49" url="Image_of_the_product"/>
<status>New</status>
</item>
<item>
<title>Cake Fresh</title>
<description>Cake Fresh offer</description>
<link>url_of_the_product_to_be_provided</link>
<pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
<startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
<endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
<media:thumbnail width="66" height="49" url="Image_of_the_product"/>
<status>New</status>
</item>
<item>
<title>KMarket Cash Counter</title>
<description>Free Take Away</description>
<link>url_of_the_product_to_be_provided</link>
<pubDate>Mon, 12 Aug 2013 13:25:07 GMT</pubDate>
<startDate>Mon, 10 Dec 2013 13:25:07 GMT</startDate>
<endDate>Mon, 20 Dec 2013 13:25:07 GMT</endDate>
<media:thumbnail width="66" height="49" url="Image_of_the_product"/>
<status>New</status>
</item>
</channel>
</rss>

```

Entry made in database

The entry saved to the database is listed here

TABLE 5. Database Table - Products

Product_Name	Description	Shop_Id	City	State	Country	Published_date	Start_Date	End_Date	Status
Fruits	Fruits offer - Flat 40% Discounts	cbFF1a	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New
ES Products	ES	cbFF1	Oulu		Finland	2013-08-12	2013-12-	2013-	New

	Products offer	a					10	12-20	
Meat Discount	Meat Discount Offer	cbFF1 a	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New
Cake Fresh	Cake Fresh offer	cbFF1 a	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New
KMarket Cash Counter	Free Take Away	cbFF1 a	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New

The core-server creates a thread which continuously sends a request to the RSS link of the website URL (once every day). The content retrieved every day is parsed and identified. The new entries are stored in the database. Duplicate entries are identified using the “Start_Date” and “End_Date” of the offer/discount, as shown in Table 5.

Day 2: RSS content received from the website URL

We assume that the RSS content polled by the Best Sale Application’s core server on day 1 is as follows,

```
<rss xmlns:media="http://search.yahoo.com/mrss/" xmlns:atom="http://www.w3.org/2005/Atom"
version="2.0">
  <channel>
    <title>SMarket - Oulu</title>
    <link></link>
    <description>Finland - Oulu Local Store Sale Deals</description>
    <language>en-gb</language>
    <image>
      <url>Image_of_the_store_to_be_provided</url>
      <title>SMarket - Oulu</title>
      <link></link>
      <width>120</width>
      <height>60</height>
    </image>
    <ttl>15</ttl>
    <item>
      <title>Frozen Vegetables</title>
      <description>Frozen Vegetables - Flat 40% Discounts</description>
```

```

<link>url_of_the_product_to_be_provided</link>
<pubDate>Mon, 13 Aug 2013 13:25:07 GMT</pubDate>
<startDate>Mon, 11 Dec 2013 13:25:07 GMT</startDate>
<endDate>Mon, 21 Dec 2013 13:25:07 GMT</endDate>
<media:thumbnail width="66" height="49" url="Image_of_the_product"/>
<status>New</status>
</item>
<item>
  <title>Ice Cream</title>
  <description>Ice Cream offer</description>
  <link>url_of_the_product_to_be_provided</link>
  <pubDate>Mon, 13 Aug 2013 13:25:07 GMT</pubDate>
  <startDate>Mon, 11 Dec 2013 13:25:07 GMT</startDate>
  <endDate>Mon, 21 Dec 2013 13:25:07 GMT</endDate>
  <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
  <status>New</status>
</item>
<item>
  <title>Chocolates</title>
  <description>Chocolates Offer</description>
  <link>url_of_the_product_to_be_provided</link>
  <pubDate>Mon, 13 Aug 2013 13:25:07 GMT</pubDate>
  <startDate>Mon, 11 Dec 2013 13:25:07 GMT</startDate>
  <endDate>Mon, 21 Dec 2013 13:25:07 GMT</endDate>
  <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
  <status>New</status>
</item>
<item>
  <title>Chips</title>
  <description>Chips offer</description>
  <link>url_of_the_product_to_be_provided</link>
  <pubDate>Mon, 13 Aug 2013 13:25:07 GMT</pubDate>
  <startDate>Mon, 11 Dec 2013 13:25:07 GMT</startDate>
  <endDate>Mon, 21 Dec 2013 13:25:07 GMT</endDate>
  <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
  <status>New</status>
</item>
<item>
  <title>SMarket Cash Counter</title>
  <description>Free Take Away</description>
  <link>url_of_the_product_to_be_provided</link>
  <pubDate>Mon, 13 Aug 2013 13:25:07 GMT</pubDate>
  <startDate>Mon, 11 Dec 2013 13:25:07 GMT</startDate>
  <endDate>Mon, 21 Dec 2013 13:25:07 GMT</endDate>
  <media:thumbnail width="66" height="49" url="Image_of_the_product"/>
  <status>New</status>
</item>
</channel>

```

</rss>

Entry made in database

The entry saved to the database is listed here

TABLE 6. Database Table - Products

Product_Name	Description	Shop_Id	City	State	Country	Published_date	Start_Date	End_Date	Status
Frozen Vegetables	Frozen Vegetables - Flat 40% Discounts	dcGG2ba	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New
Ice Cream	Ice Cream offer	dcGG2ba	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New
Chocolates	Chocolates Offer	dcGG2ba	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New
Chips	Chips offer	dcGG2ba	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New
SMarket Cash Counter	Free Take Away	dcGG2ba	Oulu		Finland	2013-08-12	2013-12-10	2013-12-20	New

8 FURTHER RESEARCH AND STUDY

The completed implementations and recommended solution presented in this thesis is only the beginning of a voyage. Refactoring is a continuous process and synchronization problems are very common in a multithreaded environment. The discussed solutions in this thesis are not exhaustive. There will be problems popping up and they will have to be addressed.

Build Environment

Jenkins is an open source continuous integration tool written in Java. Jenkins provides continuous integration services for software development. It is a server-based system running in a servlet container such as Apache Tomcat. It supports SCM tools including CVS, Subversion, Git, Mercurial, Perforce, Clear case and RTC, and can execute Apache Ant and Apache Maven based projects as well as arbitrary shell scripts and Windows batch commands. The primary developer of Jenkins is Kohsuke Kawaguchi. Released under the MIT License, Jenkins is free software. Builds can be started by various means, including being triggered by commit in a version control system, scheduling via a cron-like mechanism, building when other builds have completed, and by requesting a specific build URL (Jenkins, 2014, date of retrieval 06.03.2014).

Jenkins build environment is useful to build the code commits to maintain the source. In a work environment code committed by each team member should not affect the build environment of the source repository.

Java Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an

individual method. Unit tests are created by programmers or occasionally by white box testers during the development process.

Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. Its implementation can vary from being very manual (pencil and paper) to being formalized as part of build automation. JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit. JUnit is linked as a JAR at compile-time; the framework resides under package junit.framework for JUnit 3.8 and earlier, and under package org.junit for JUnit 4 and later. (Java Unit Testing, 2014, date of retrieval 06.03.2014)

Test based approach is a healthy coding environment. Creating test cases and making sure each one of them pass is mandatory before the code is submitted to the repository. This prevents build issues and creates a more performing environment.

Memcached

Memcached is a general-purpose distributed memory caching system that was originally developed by Danga Interactive for LiveJournal, but is now used by many other sites. It is often used to speed up dynamic database-driven websites by caching data and objects in RAM to reduce the number of times an external data source (such as a database or API) must be read. Memcached runs on Unix, Linux, Windows and Mac and is distributed under a permissive free software license. Memcached's APIs provide a giant hash table distributed across multiple machines. When the table is full, subsequent inserts cause older data to be purged in least recently used (LRU) order. Applications using Memcached typically layer requests and additions into RAM before falling back on a slower backing store, such as a database. (Memcached, 2014, date of retrieval 06.03.2014)

The Best Sale Application server needs to be more responsive and serve more client requests. Memcached is a quite useful technique in helping the core servers from client request loads. Memcached also reduces network calls.

9 CONCLUSION

The Server for a Best Sale Application project has a scope of money generation, for all buyers at anytime and anywhere who can access Internet via Mobile. The shop owner can save a lot of money on paper printing and postal charges in delivering the sale advertisement paper. Also, their offers reach the buyer much faster than through a traditional advertisement paper delivered to buyers home. The user can check the bargain price of a product in another shop when he/she is inside a shop where for the same product there are no good offers.

I feel there is definitely a scope of success for such applications on a market. Marketing the product would be the key.

REFERENCES

1. Java Tutorial .2014. Introduction to Concurrency. Date of retrieval 06.03.2014. <http://docs.oracle.com/javase/tutorial/essential/concurrency/>
2. Tutorials Point .2014. Java Threads Life Cycle. Date of retrieval 06.03.2014. http://www.tutorialspoint.com/java/java_multithreading.htm
3. Wordpress Beginners .2014. RSS introduction and usage. Date of retrieval 06.03.2014. <http://www.wpbeginner.com/beginners-guide/what-is-rss-how-to-use-rss-in-wordpress/>
4. Tutorials Point .2014. RSS advantages for Publishers and Subscribers. Date of retrieval 06.03.2014. <http://www.tutorialspoint.com/rss/rss-advantages.htm>
5. Jenkins .2014. First release of Jenkins published by Kohsuke Kawaguchi. Date of retrieval 06.03.2014. <https://www.java.net/blog/kohsuke/archive/20070514/Hudson%20J1.pdf>
6. Java Unit Testing .2014. JUnit Cookbook by Kent Beck and Erich Gamma. Date of retrieval 06.03.2014. <http://junit.sourceforge.net/doc/cookbook/cookbook.htm>
7. Memcached .2014. Introduction to Memcached. Date of retrieval 06.03.2014. <https://code.google.com/p/memcached/wiki/NewOverview>