ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

Enhancing client side search using Django and image cache

Venkatesh Raghul E, Niranjana G, Vignesh G
Department of CSE, SRM University, Chennai, India
***Corresponding author: E-Mail: venkatesh_elango@srmuniv.ac.in**

ABSTRACT

The basis of this paper is to create an online information system for the users to be updated about the current mark down in the market. This paper involves developing a website to help the owners of a brand/company to monitor the outlets that are functioning under them. The website enables the owners to register by entering their credentials, after which, the owners will be able to access the various features of the website. The owners will be able to add the outlets/franchise by which a separate username and password would be generated for each outlet/franchise. Using this username and password the franchisee would be able to access the website and fill in their details and the owners could keep track of it. The website is optimized using client side search (Django) and image cache. Where client side search will ease the creation of database driven websites and emphasizes on reusability of components and image cache increases the speed of retrieving or loading an image. Further this may be used as an app so that the users would be able to get the location of various outlets and view the offers/discounts availed in each outlet thereby providing necessary information to the user for effective decision making. The system uses REST API architecture to make the client server communications faster thereby creating a high performance environment for the users.

KEY WORDS: Client search, Image cache, REST API

1. INTRODUCTION

Traditionally the applications are mostly confined into online sales whereas this paper deal with providing valuable information to the users so as to save a lot of time looking around in various places. This helps two set of users, namely, the owners can login into the website and keep track of their outlets and secondly the users who will be able to access the information through the android application . This not only deals with the functional aspects but also looks into the nonfunctional aspects of the software. The web application is designed using phpmyadmin in which the database connectivity is achieved and the android application is done using android studio. Basically the system is designed using master slave architecture. The master slave architecture is a design in which the slave is controlled and managed by a central unit known as the master. All the changes and modification done to the slave will be known to the master in this type of architecture.

As it is a software product the functional and nonfunctional aspects of the product are further enhanced in this system. The functional aspects will be explained in the forthcoming pages whereas the nonfunctional aspects such as optimization, speed up, memory etc. are enhanced in this system. The nonfunctional aspects are further enhanced using two techniques such as client side search and image cache which will be elaborated later. The software product has its own life cycle starting from the requirement phase to the maintenance phase which is followed by this system as well. In the requirement phase the documentation of all the modules are noted down in order to produce a system that satisfies the entire set of requirements. The requirement phase is considered as an important task because the end product is designed based on the set of requirements that are provided. In the design phase the working and operation of various modules that are present in the requirements are designed. The requirements identified in the Requirements Analysis Phase are transformed into a System Design Document that accurately describes the design of the system and that can be used as an input to system development in the next phase. In the implementation phase the various modules that are designed are implemented or rather developed using various programming platforms.

Literature survey: The objective of this android and web application was to analyze the working and functionalities of various markets that are used in our daily basis. The web application was done with the motive of analyzing the working and functionalities of the subordinates of a particular market whereas in contrast the android application was designed exclusively for the users so as to provide them with the knowledge of the market in the current trend.

Mobile application for finding the ATMs: This paper delineates a mobile application that is developed for locating the nearby ATMs quickly. The application shows the ATMs of various banks from Rijeka and Croatia around your locality. It also provides the shortest distance for the chosen ATM from your location. Several mapping libraries that are available for Android applications such as, Google Maps API, ArcGIS Runtime API, OSMDroid API and MapsForge, are used in this application. The Google Maps Android API uses OpenGL ES version 2 to render the map. If OpenGL ES version 2 is not installed, map will not appear.

The application is launched by passing a method called start Activity () to the system that opens a window with user interface displaying all banks that have ATMs in Rijeka. For finding locations of each ATM, geo-coordinates posted on the official documents of each of the banks were used. Furthermore, the application offers all sorts of information available for the selected ATM and advices on where you can withdraw your money free of charge.

Product Aspect Ranking Techniques: A Survey: A product may have hundreds of aspects. Some of the product aspects are more important than the others and have strong influence on the eventual consumer's decision making as well as firms' product development strategies. Identification of important product aspects become necessary as both consumers and firms are benefited by this. Consumers can easily make purchasing decision by paying attention to the important aspects as well as firms can focus on improving the quality of these aspects and thus enhance product reputation efficiently. This provides the description of various techniques for product aspect identification and classification.

Android App Development: In recent years, the emergence of smart phones has changed the definition of mobile phones. Phone is no longer just a communication tool, but also an essential part of the people's communication and daily life. Various applications added unlimited fun for people's lives. It is certain that the future of the network will be the mobile terminal. Now the android system in the electronics market is becoming more and more popular, especially in the Smartphone market. Because of the open source, some of the development tools are free, so there are plenty of applications generated. This greatly inspired the people to use the android system. In addition, it provides a very convenient hardware platform for developers so that they can spend less effort to realize their ideas. This makes android popular and of course scope for further development. As the smart phones and android system getting popular, the activities like listening to music, watching videos surfing the internet etc. are moved from the computer to a phone now. The major attractive feature is the lack of interference of built-in advertisements which many of us hesitate to have which we experience when using computer systems. The development of the android application can not only be limited to the function, more attention should be paid to the user's experience .After studying and experiencing some previous android applications ,we decided to use the Java language, the Eclipse platform, android ADT and the android SDK to develop the mobile application for ordering food from restaurants named PikDish. This system has a nice interface and smooth operation. Besides that it won't steal any personal information and bring a wonderful user experience.

So while surveying as to on which platform these applications could be integrated, we selected android for its various advantages. The advantages are listed below.

Android is open: Because it is Linux based open source so it can be developed by anyone.

Multitasking: Android phones can run many applications; it means you can browse while listened to the song.

Easy access to the Android App Market: Android owners are people who love mobile phones application, with Google's Android App market we can download many applications for free.

Can install a modified ROM: We sometimes find an unofficial ROM. That is the version that was not in accordance with the specification release our cell phones, the last way is modification. Do not orry, there are many custom ROM that you can use on Android phones, and guaranteed not to harm your device.

Phone options are diverse: Android is different than the IOS, if the IOS is limited to the iPhone from Apple, then Android is available on mobile phones from various manufacturers, from Sony Ericsson, Motorola, HTC and Samsung. And each handset manufacturer also presents an Android phone in the style of each, such as Motorola with its Motoblur, Sony Ericsson with its Timescale. So you can freely choose the Android phone in accordance with the 'brand' favorite.

Widget: With the widgets on the home screen, you can easily access a variety of settings easily and quickly.

The following lists the Disadvantages

Need internet connection: Android requires an active internet connection. At least there should be a GPRS internet connection in your area, so that the device is ready to go online according to our needs.

Advertising: Application in the Android phones can indeed be obtained easily and for free, but the consequences in each of these applications, will always be ads on display, either the top or bottom of the application.

Wasteful Battery: Android more wasteful than any other operating system, because this operating system is a lotof "process" in the back ground that lead to the battery quicklydrains.

Many applications contain virus: virus inserted android applications like Counter Strike Ground Force, Puzzle, Photo Game, etc. Androis Application contain virus also present in the Android market. Android also has its fair share of disadvantages but when compared to other platforms android seems to be a very a better platform. Along with this we surveyed the popularity of the operating system. Market share of android which was mere 2.8% in 2009(initial stage), boosted to 68% till August, 2015 which is almost half the share of the total market. Our basic aim is to make the application reach as many people as possible and this goal is achieved by implementing the application on android.

2. MATERIALS AND METHODS

Introduction: System Analysis is a combined process dissecting the system responsibilities that are based on the problem domain characteristics and user requirements. The goal of the system analysis is to completely specify the technical details for the main concept in a concise and unambiguous manner.

Problem Definition: A website that enables the owners of a company to manage their outlets across various regions, thereby ensuring owner monitoring. The owners will be able to add the outlets and offers that are available. Further this may be converted into an android application which would display the offers available at various outlets.
System Overview: The purpose of system analysis is to produce the brief analysis task and also to establish complete information about the concept, behavior and other constraints such as performance measure and system optimization.
Objective: The objective of this paper is to create a website to help the owners of a brand/company to monitor the outlets that are functioning under them. The website enables the owners to register by entering their credentials, after which, the owners will be able to access the various features of the website. The owners will be able to add the outlets/franchise by which a separate username and password would be generated for each outlet/franchise. Using this username and password the franchisee would be able to access the website and fill in their details and the owners could keep track of it. The website is optimized using client side search (Django) and image catch. Where client side search will ease the creation of database driven websites and emphasizes on reusability of components and image catch increases the speed of retrieving or loading an image. Further this is converted into an app so that the users would be able to get the location of various outlets and view the offers/discounts availed in each outlet thereby providing necessary information to the user for effective decision making.
` The project was developed in the view of increasing the optimization and efficiency of the system rather than concentrating on the user interface and aesthetic features of the system. When the framework was developed it was limited only to a few features and many more features can be added in future. The security of the system can be implemented once the system is deployed in the market. It was basically developed targeting the android market and all the older markets are not taken into consideration.
Existing System: Mobile application that gives Android users the ability to find ATMs quickly. Currently, the application offers locations of ATMs for various banks in Rijeka, Croatia. It also provides the possibility of finding the shortest path to the chosen ATM, as well as helpful tips where they can withdraw the money free of charge. The idea for this application was born after realizing that many people were wasting time looking for a place to withdraw money. This application saves time to inhabitants as well as to visitors of the city.
Disadvantage
- Does not have a JavaScript framework to reduce complexity
- Size of the application is big
- Difficulty of testing JavaScript applications
- Synchronization of data between different user sessions
- High initial loading time
- Inability of search engines to index JavaScript applications

Proposed System: Proposed system focus on reducing the complexity, improving speed and responsiveness of the application. In proposed system, we are using,
- Client side search which helps in automated search thereby decreasing the time complexity of the system.
- REST API which enables faster client and server communications and increases the memory capacity of the system.
- Owner monitoring is possible in this system.
- The concept of image cache is used in this system which again improves the speed and memory capacity of the system.

System Design: In this project the system design is divided into frontend design and the backend design. This project consists of a web application and an android application which will have its own system design. Initially the web application was developed and the design process included the layout of all the web pages and their respective navigation on performing a particular operation. Further the database schema was designed for all the categories and modules of the web application. Database schema is nothing but a skeleton structure that represents the logical view of the database. It defines how each data is organised and provides the relationship between various data and tables in a database. Apart from all these the system architecture forms the important part of the design process. It is designed in such a way that it satisfies the non-functional requirements or quality requirements of the system.
System Architecture: A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. In this paper the system architecture is designed by taking reference from the master slave architecture. Master slave architecture is nothing but a centralized node that will have control over the other nodes that are connected to the central node. The central node is considered as the master and the other nodes that are connected to the master are the slaves to that master. The slaves are nothing but nodes that perform the work provided by the

master and revert back to the master after the work is done. Any changes that are made to the slaves will be updated to the master thereby ensuring that the masters are aware of the working of the slaves.

In this project the same architecture is used so as to ensure that the owners are aware of the working of the outlets. The owners of a particular company are considered as the master in this architecture and the outlets that are added by the owners act as the slave to those owners. Any changes or updates that happen in the slave will be affected or informed to the master. This architecture is mainly used in this project to ensure owner monitoring. By using this architecture the owners will be able to monitor the outlets and reduce the risk of frauds. Any malicious activity or operations that are out of bounds of the rules of the company can be tracked with the help of this architecture. A database is "slaved" to a "master" when it receives a stream of updates from the master in near real-time, functioning as a copy. The "slave" must simply apply the changes that the master validated and approved. In principle, one could create a master-slave setup by transferring files really fast from one server to the other; but in practice, each data base has its own specialized replication. There are many reasons a replica makes queries return faster. One is that the master's CPU is less burdened by queries while the replica's CPU is less burdened by writes; so there is more CPU available to do work. Another is that data, while nominally stored to disk, is cached in RAM when that is possible; and indeed the OS does this whether the database asks for it or not and can be unpredictable about it. On the master, recently written portions of the database and anything used to support writes, for example, indexes used to check uniqueness will be in RAM all the time; these must share space with the portions of the database loaded to serve queries. The replica needn't load as much data to support its write load (it is writing to the same parts of the database, but can naively commit it without checking anything) and it makes more RAM available to serve queries. There is also the matter of disk access. When data is written on the master, it must be written to disk before the master can return a successful commit message to the application. To write to disk, the operating system must make the disk head available. If the disk head is far from the place where we need to write the updated or added record, then it must be moved. For the disk head to "seek" can be quite time consuming, relative to other operations. For this reason, database developers have cleverly chosen to have their databases write all changes one right after another, next to each other on the disk, until a few changes have been written. Thus the disk head needn't move at all from the place where it was when the last record was written. This log of records does not store them in the most efficient way -- the database "heap" is where the records are most efficiently stored -- but for a few records at a time, it's not a problem. Every so often, the database software takes all the logged changes and writes them into the heap. This latter step does, indeed, move the disk head; but at least it hasn’t moved it back and forth all over the heap Reads do not allow these kinds of shortcuts. To read data, we must go to its place on disk and read it out. This reintroduces pressure to move the disk head back and forth. A replica has its own disk head (or heads) and can thus take on some of this load of seeking the data from the various disk heads that forms the memory fragment of the system. One could realize these same benefits by getting a bigger server, too; one with more disks, CPU and RAM. Doing that without downtime is a little tough, though. And whereas a replica gives you an option in the event that the master goes down, one big database server is not any more reliable than one small database server.

As we can see in figure the database of the master is replicated to the slaves so that they share the same database contents and the master slave architecture is achieved. The data from the application server is stored into the master database and the slave database is simply replicated from the master database. In the slave database the write operation cannot be performed from the application server. So when this happens, the memory for the extra duplicate database is reduced. The slave database will only be able to read the data from the application server. Any updates made to the slave database will be updated on the master database.

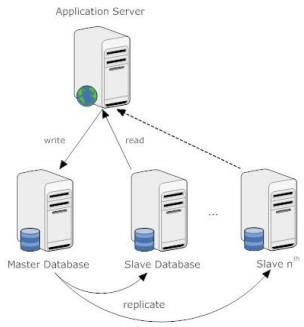

Figure.1.Client & Application Server Communication

Advantages

- It satisfies the ACID properties i.e. atomicity, consistency, isolation and durability.
- Each slave will contain only one master so it will be easy for the slaves to locate the master.
- No two slaves are associated and are independent under a particular master.

Disadvantages

- If the master fails all the slaves under that particular master fails.
- All the slaves are single master dependent and two masters are not associated at any point of time.

To overcome this disadvantages a master- master architecture can be used so that even if one master fails the other master will be able to replace the failure master. But when it comes to this system the usage of a master-master architecture is quite complex. It is complex because the consistency of the system will be reduced as each slave will be associated with one or more master. In that case it will be difficult to find the masters that are associated with a particular slave. Another major reason for the complexity is that this system contains various categories and is diverse, so implementing the master-master architecture will become difficult. If the system is smaller the master-master architecture would prove to worth implementing. For this particular system which is diverse this architecture is not advisable but it can be implemented. Once it's implemented a large amount of testing is required to ensure its consistency.

System Implementation: System implementation is a phase that is performed after the design process of a software product. The implementation in this application is divided into two; first is the implementation of the web application followed by the implementation of the android application. The implementation of the web application was done using PHP, JavaScript and HTML. The various modules of the software were programmed in the above mentioned programming languages. The above mentioned programming languages are used because of its versatile nature in developing a web application. php myadmin was used to create the database for storing the data into the respective tables. The system was divided into various modules, after the implementation of each module aggregation of all the modules are done to form the final product. Implementation phase consisted of incorporating various optimization features such as

- Client side search
- Image cache
- REST API

The above mentioned optimisation feature implementation are followed by the basic system implementation and their respective working structure.

3. RESULTS AND DISCUSSION

Client Side Search: In order to understand about client side search we need to know what is client side. Client side is nothing but the operations that are performed by the client in a client server relationship. The web application or other application that appears on the local computer or workstation is known as client side and would access the use of the server for certain operations. The client side is basically developed using client side scripting techniques using which the client side can be developed. It is basically writing a code that will run on the client using languages that can be executed by the browser such as JavaScript. JavaScript is highly flexible and it is used to facilitate client side search in this project. Client side search is nothing but an automated search that can be done on the client side without connecting to the server. For example it is similar to that of the Google search engine wherein the search is automated and once we start typing it would guess what our requirement would be and suggest certain web pages that would almost match our requirement. The similar idea is implemented in this project wherein the search bar is enabled with client side search and as soon as we type it would lower down on the various possibilities of our requirement and finally match with our requirement. This client side search is done to improve the speed and performance of the web application thereby improving the non-functional requirement of our product. Client side search is enabled with help of Django which is a web development framework used to improve and enhance the ease of developing a website. It's majorly done to save time and improve on the performance of the web application.

As we see in the example the jquery. quicksearch.js is the header for the search script which enables the use of quicksearch(). Quicksearch() is a function that is used to perform client side searching in this project. This function is used to return the search item faster when compared to the normal search where the entire text has to be typed in the search bar and then the result would be retrieved. As soon as few letters are typed the result would be suggested and the search is made faster. This concept is introduced in order to improve the performance of the web application.

Image Cache: Image cache is a small PHP class that accepts .png, .jpg, .gif extension files which can be compressed, moved, cached in the user browser. It will then show the new location of the image in the image tag. The image tag gives the source or location from which the image has to be loaded. It has two attributes that is the <src> and <alt>. Using image cache would change the source of the image that would be altered in the image tag. The major purpose of using this function is to improve or shorten the page load time. Page load time is one of the major factors in using an application. If the page load time is slow then the users would find it boring to use the application. In order to facilitate fast access of the pages image cache has been used in this project.

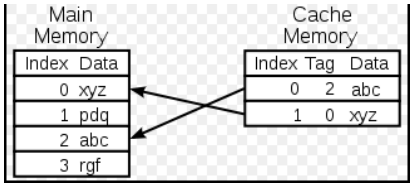

Figure.2.Image cache

In image cache the images that are regularly accessed will be stored in the cache memory so as to access the web pages faster. For example consider abc and xyz are the frequently accessed images, these images will be stored in the cache memory and on further invocation of these images the data will be retrieved from the cache memory instead of the main memory thereby facilitating faster access time of the web page.

As seen in the above example the echo function is used to print or retrieve the data or image from the cache. The images location changes dynamically after each click on the image and the new location will be stored on the image tag.

Rest API: In computing, representational state transfer (REST) is the software architectural style of the World Wide Web. More precisely, REST is an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements. Through the application of REST architectural constraints certain architectural properties are induced: Performance, Scalability, Simplicity, Modifiability, Visibility, Portability, and Reliability.

To the extent that systems conform to the constraints of REST they can be called RESTful. RESTful systems typically, but not always, communicate over Hyper Text Transfer Protocol (HTTP) with the same HTTP verbs (GET, POST, PUT, DELETE, etc.) that web browsers use to retrieve web pages and to send data to remote servers.

The name "Representational State Transfer" is intended to evoke an image of how a well-designed Web application behaves: a network of web pages (a virtual state-machine), where the user progresses through the application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and rendered for their use.

REST allows for a minimum amount of data to be passed using the same well-established mechanisms that define the web without a lot of the encumbrances introduced by fatter protocols. This allows programmers to more rapidly build programs that access and act upon data exposed via APIs, even in environments with unreliable network speeds and limited computing power.

REST works almost exactly like a website in a browser. A resource is exposed to a program via a URL. The program can access that URL and receive data about the resource, not unlike when you type in a URL to your browser and get a web page back. Well-designed RESTful APIs include additional links the program can follow to request related information - similar to how you would click on a link to visit a new page - or to submit data to update the given resource - similar to how you would fill out a web form to create a new account for a web site.

The tradeoff for all of this flexibility is a lack of strongly set standards. REST really describes the method by which the data is transferred, but implementers have been left mostly on their own to figure out how this data should look. That's rapidly changing, especially as the number of mobile devices accessing data across the network increases, but it will be a while before these formats coalesce into something resembling true standards.

REST is often used in mobile applications, social networking Web sites, mash up tools, and automated business processes. The REST style emphasizes that interactions between clients and services is enhanced by having a limited number of operations (verbs). Flexibility is provided by assigning resources (nouns) their own unique Universal Resource Identifiers (URIs). Because each verb has a specific meaning (GET, POST, PUT and DELETE), REST avoids ambiguity.

4. CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion: As explained earlier this application is a unique application in the market. This application would prove to be a boon to the users. This application was developed to create high degree of user interactivity rather than other traditional applications. This project targets two types of users like the entrepreneurs and the common man. The basic and initial targets are the entrepreneurs for whom the web application was developed. The major use of this application for the entrepreneurs is owner monitoring. The various features like client side search and image cache increases the performance of the web application making it easier for the users. The faster and ease of working in the web application are the highlights of this project. The user interface is designed in such a way that the user will have no trouble using this application thereby making it user friendly.

The second type of users is the end users for whom the android application was developed. As said earlier android is a growing market and its widely used by the people when compared to other platforms. This application

consists of the REST API architecture which helps in faster communication between the client and the server. This architecture is not used by many android applications thereby increasing the efficiency than other applications. The android application can be used by the end users to view the offers that are present in various outlets of various organisations. The above offer information is entered in the web application by the first set of users. The end users will be able to view the various outlets that are available in their surroundings. By using this application the users can save the time they take to travel to various places and find out the information about the offers in various outlets. The user interface of the android application is designed in such a way that the end users would find it easy to use.

Future Enhancements: The features that are displayed in the android application is limited only to a few components. There is scope for many more additional options that can be implemented or incorporated in the application. To be more precise the android application displays only the various offers that are available to the users. In future the location, images of the products and various other features can be added to the application making it more attractive and create a good platform in the market. Basically the framework of the application is in the early development phase and more testing has to be performed before deploying it in the market.

Another major aspect that must be done in the future is increasing the security of the system. The security of the system is confined to authorization and authentication of the user and the system is not secured from external attacks. External attacks are nothing but the attacks that are made by a third party who finds the weakness of a system and enters into the system. They can inject malicious content into the system and disrupt the working of the system. So the system should be added with android security features so as to keep it safe from external environment.

As it is the future of the system looks really bright and it has good scope of improvement and enhancements. Once the enhancements are implemented this system would prove to be a leader in the market.

REFERENCES

Barish G, Obraczka K, World wide web caching ; trends and techniques, IEEE Communications Magazine, 38, 2000, 178-185.

Gero Decker, RESTful Petri Net Execution,Web Services and Formal Methods, Springer-Verlag LNCS, 5387, 2009, 73-87.

Tsai SS, Chen D, Takacs G, Chandrasekhar V, Singh JP and Girod B, Location coding for mobile image retrieval, MMCC, 8(1), 8-7.

Wang Jun-ming, WAP Technology,The Gate of Mobile Mulmedia. Mobile Communication, 2001.