GALATASARAY UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ENERGY EFFICIENCY ANALYSIS OF A GPS-BASED AUGMENTED REALITY APPLICATION

Ahmet KARAMAN

Jan 2016

ENERGY EFFICIENCY ANALYSIS OF A GPS-BASED AUGMENTED REALITY APPLICATION

(GPS TABANLI ARTTIRILMIŞ GERÇEKLİK UYGULAMASINA AİT ENERJİ VERİMLİLİĞİ ANALİZİ)

by

Ahmet KARAMAN, B.S.

Thesis

Submitted in Partial Fulfilment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

INSTITUTE OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Jan 2016

This is to certify that the thesis entitled

ENERGY EFFICIENCY ANALYSIS OF A GPS-BASED AUGMENTED REALITY APPLICATION

prepared by **Ahmet KARAMAN** in partial fulfilment of the requirements for the degree of **Master of Science in Computer Engineering** at the **Galatasaray University** is approved by the

Examining Committee:

Yrd. Doç. Dr. Gülfem ALPTEKİN (Supervisor)
Department of Computer Engineering
Galatasaray University -------------------------

Yrd. Doç. Dr. İ. Burak PARLAK
Department of Computer Engineering
Galatasaray University -------------------------

Yrd. Doç. Dr. Ayşe TOSUN
Department of Computer Engineering
İstanbul Technical University -------------------------

Date: -------------------------

ACKNOWLEDGEMENTS

I would like to thank to my colleague and project partner Doğa Erışık, my supervisor Dr. Gülfem Işıklar Alptekin and Dr. Özlem Durmaz İncel for their contributions and support. I would also like to thank to Turkish Ministry of Science, Industry and Technology and Yapı Kredi Bank for giving me this opportunity and their support. Yapı Kredi Bank provided the real GPS data of their merchants and supported the mobile application development.

This work is supported within the scope of SAN-TEZ program of the Turkish Ministry of Science, Industry and Technology under grant number 0307.STZ.2013-2.

January 2016
Ahmet KARAMAN

TABLE OF CONTENTS

LIST OF SYMBOLS

AR	**:** Augmented Reality
GPS	**:** Global Positioning System
YKB	**:** Yapi Kredi Bank
SARAS	**:** Sensor-Based Augmented Reality Application Software
LBA	**:** Location Based Application
UI	**:** User Interface
CPU	**:** Central Processing Unit
DBMS	**:** Database Management System

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Augmented reality (AR) applications on mobile devices have been used quite widely in recent years. The main function of these applications is to build an interaction with the real world environment by taking an image on the device. In this way, the user can determine the behaviour of real-life, based on the feedback received from the device. Although this technology is more widely used in the military field, it is also useful in the tourism and entertainment fields. There are several commercial mobile applications under development that have similar functions such as the proposed one in this thesis. Many of these applications that use mobile phone's GPS, compass and other sensors, show information on the screen and they identify user's whereabouts and direction. This brings a number of challenges. For example, GPS gives accurate results outdoors, but it does not work properly at indoor environments. In order to produce more accurate results on the device, an internet connection is usually required. Besides, sensors for direction information must be on at all time. Accordingly, the information placed on the screen is constantly changing locations. All of these functions increase the process density and hence battery consumption on device.

In this work, an Android-based AR application which is designed and developed as a part of a research project was used. The application provides information about one of the biggest Turkish banks' (Yapı Kredi Bank) merchants, branch information, and related campaigns on the display. The main objective of the application is to enhance the interaction between the mobile clients as well as the potential mobile clients by providing information about sales campaigns. The developed application uses GPS, compass, gyroscope, accelerometer sensors and it utilizes a display placement algorithm. As a part of this thesis, the server side of the SARAS application, which includes the database as well, is designed and developed. Additionally, the QR-code scanning property is added to the client side on the mobile application. However, the main focus of the thesis is the examination of the resources and energy consumed by the application. The performance with the other most known algorithms and applications are also compared, and then propose improvements.

Firstly, the basic resource controls, such as closing the resources that are used in the application and controlling the battery life have been applied. Then power consumption tools and methods in Android platforms are examined. The improvements suggested by these tools are applied on the SARAS application. These improvements reduced the resource consumptions up to %30. Detailed energy analysis has been performed under different GPS and sensor processing settings. With the results of this detailed analysis, the effects of the sensors and GPS to the energy consuming have been clearly demonstrated. The comparison tests were done with the similar commercial applications. The result of these tests reveal that the resource consumption of SARAS application is comparable to the given commercial applications.

RÉSUMÉ

Applications de réalité augmentée sur les appareils mobiles ont été utilisées assez largement au cours des dernières années. La fonction principale de ces applications est de construire une interaction avec l'environnement du monde réel en prenant une image sur l'appareil. De cette façon, l'utilisateur peut déterminer le comportement de la vie réelle basée sur les commentaires reçus de l'appareil. Bien que cette technologie est plus largement utilisé dans le domaine militaire, il est également utile dans les domaines du tourisme et de divertissement. Il existe plusieurs applications mobiles commerciales en cours de développement qui ont des fonctions similaires, tels que celui proposé dans cette thèse. Beaucoup de ces applications qui utilisent le GPS, la boussole de téléphone mobile et d'autres capteurs, afficher des informations sur l'écran et ils identifient les allées et venues et la direction de l'utilisateur. Cela apporte un certain nombre de défis. Par exemple, le GPS donne des résultats précis à l'extérieur, mais il ne fonctionne pas correctement à des environnements intérieurs. Afin de produire des résultats plus précis sur le dispositif, une connexion Internet est généralement nécessaire. En outre, des capteurs pour des informations de direction doivent être sûr en tout temps. En conséquence, l'information placée sur l'écran est en constante évolution emplacements. Toutes ces fonctions augmenter la densité de processus et donc la consommation de la batterie sur l'appareil.

Dans ce travail, nous utilisons une application de réalité augmentée basé sur Android qui est conçu et développé dans le cadre d'un projet de recherche. L'application fournit des informations sur l'une des plus grandes banques turques (Yapı Kredi Bank) marchands, des informations de branche, et des campagnes connexes sur l'affichage. L'objectif principal de l'application est d'améliorer l'interaction entre les clients mobiles, ainsi que les clients mobiles potentiels en fournissant des informations sur les campagnes de vente.

L'application développée utilise le GPS, boussole, gyroscope, capteurs d'accéléromètre et il utilise un algorithme de placement d'affichage. Dans le cadre de cette thèse, le côté serveur de l'application SARAS, qui inclut la base de données ainsi, est conçu et développé. En outre, la propriété de balayage par code QR est ajouté au côté client de l'application mobile.

Toutefois, le principal objectif de la thèse est l'examen des ressources et de l'énergie consommée par l'application. Nous comparons également les performances avec les autres algorithmes et applications les plus connues, et ensuite proposer des améliorations.

En premier lieu, les contrôles de ressources de base, telles que la fermeture des moyens qui sont utilisés dans l'application et le contrôle de la durée de vie de la batterie, ont été appliquées. Ensuite, des outils et des méthodes dans les plates-formes Android consommation d'énergie sont examinés. Les améliorations proposées par ces outils sont appliqués sur l'application SARAS. Ces améliorations ont réduit les consommations de ressources jusqu'à 30%. L'analyse de l'énergie par la suite détaillé a été effectuée sous différents réglages de GPS et de traitement du capteur. Avec les résultats de cette analyse détaillée des effets des capteurs et le GPS au consommant de l'énergie avaient été clairement démontrés. Aussi les tests de comparaison ont été effectués avec les applications commerciales similaires. Le résultat de ces essais montre que la consommation de ressources de l'application SARAS est comparable à ces applications commerciales.

ÖZET

Mobil cihazlar üzerindeki arttırılmış gerçeklik uygulamaları son dönemde oldukça yaygın olarak kullanılmaya başlanmıştır. Bu uygulamaların temel işlevi, cihaz üzerindeki görüntüyü alarak gerçek dünyadaki ortamla birleştirmektir. Bu şekilde kullanıcı cihazından aldığı geri bildirimlere göre gerçek hayattaki davranışlarını daha bilinçli şekilde yönlendirebilmektedir. Bu teknoloji askeri alanda daha yaygın olarak kullanılmasına karşın, turistik ve eğlence gibi amaçlarla da kullanım alanı bulmaktadır. Bu tezde önerilen uygulama ile benzer işlevlere sahip geliştirilme aşamasında olan birçok ticari mobil uygulamalar bulunmaktadır. Bu tür arttırılmış gerçeklik uygulamalarının çoğu, telefonun GPS, pusula ve diğer algılayıcılarından yararlanarak kullanıcının bulunduğu yeri ve yönü belirleyerek ekran üzerinde göstermektedir. Bu işlemler, bir takım zorlukları da beraberinde getirmektedir. Örneğin, GPS açık alanlarda çok yakın mesafelerde bile doğru sonuçlar vermekte, ancak iç mekanlarda düzgün çalışmamaktadır. Daha doğru sonuç üretebilmesi için cihaz üzerinde internet bağlantısına ihtiyaç duyulabilmektedir. Ayrıca yön bilgisini elde etmek için algılayıcıların sürekli olarak çalışıyor durumda olması gerekmektedir. Ancak bunlar sayesinde, ekran üzerinde yerleştirilen bilgilerin yerleri sürekli olarak güncellenebilmektedir. Bütün bunlar, cihaz üzerindeki işlem yoğunluğunu ve dolayısıyla pil tüketimini arttırmaktadır.

Bu tezde, Türkiye’nin en büyük bankalarından biri olan Yapı Kredi Bankası ile iş birliği yapılmış ve uygulama için gerçek veriler kullanılmıştır. Bankaya ait üye iş yerleri, şubelerini ve ilgili kampanyaları ekran üzerinde gösteren Android tabanlı bir artırılmış gerçeklik uygulaması tasarlanmıştır. Geliştirilen bu uygulama üzerinde farklı GPS ve algılayıcı işleme yöntemleri ve ekrana yerleştirme algoritmaları uygulanarak, bunların kaynak tüketimleri incelenmiştir. Ayrıca diğer uygulamalar ve algoritmalar ile karşılaştırmalar yapılarak uygulama için iyileştirmeler yapılmıştır.

İlk olarak, kullanılan kaynakların kapatılması ve pil ömrü kontrolü gibi temel kısımlar iyileştirilmiştir. Sonrasında Android platformlardaki performans analiz araçları ve metotları incelenmiştir. Bu araçlarla önerilen geliştirmeler SARAS uygulaması üzerinde uygulanmıştır. Yapılan bu iyileştirmeler sonucunda kaynak tüketimlerinde yaklaşık %30 azalış sağlanmıştır. Bunun yanı sıra geliştirilen uygulama farklı modlarla ve GPS teknikleriyle çalıştırılarak kaynak tüketimi üzerine nelerin etki ettiği detaylı şekilde incelenip analiz edilmiştir. Ayrıca benzer ticari uygulamalar ile de karşılaştırma testleri yapılmıştır. Uygulanan testlerin sonucunda benzer ticari uygulamalar ile kaynak tüketimi açısından çok yakın sonuçlar verdiği gözlemlenmiştir.

1. INTRODUCTION

Augmented reality (AR) applications help the users interact with their surroundings via the mobile devices. Many of these applications; using phone's GPS (Global Positioning System), compass and other sensors (gyroscope, accelerometer etc.), show information on the display identifying the user's whereabouts and provide direction/navigation information. System identifies the location from GPS sensor, the orientation from accelerometer and the direction from magnetic field sensors, and uses these to provide information on the mobile display. The surrounding information can be gathered by web services via internet connection or from the local data stored in the mobile device.

LAYAR is one of the example applications in the relevant market of AR applications (LAYAR, 2009). This application provides information related to the locations of the phone user, from very large popular worldwide data sources such as Twitter, Instagram, Facebook and using the geocode information in the data. Various information is provided: for example the predefined structures, such as information on restaurants or rental property close to the user's location can be provided and displayed as a layer on the user's screen. LAYAR application supports a number of services and an annual license fee must be paid (LAYAR Pricing, 2009). Another example application Turkcell Pusula/ En Yakın (TURKCELL, 2012), identifies nearby points like pharmacy, entertainment etc. and shows the proper route to reach this point on Google Maps. The Wikitude application (WIKITUDE, 2008) works with camera support on most of the devices, and shows nearby places and which twitter users have been in these places via indicating the direction to them.

In most of the studies in the literature that utilize sensors on smart phones, data is collected on the phone and more powerful processors in the data processing is performed off-line on a computer.

When the limited processing capabilities and battery capacity of the phones are taken into consideration, such a scheme may not be efficient especially requiring real-time performance. Thus, for the development of real-time applications resource constraints on the phones should be considered. The main purpose of this study is to develop a mobile AR application with energy efficient methods considering resource consumption under different GPS and sensor processing settings.

In this work, an AR application entitled 'SARAS (Sensor-based Augmented Reality Application Software)', which shows bank merchants and branch information on the display, is focused (Erışık, 2016). Particularly the resource consumption of the application is analysed and it is made an energy efficient and resource friendly application by considering battery controls and performance improvements. SARAS works on Android based mobile devices and displays certain information to clients or potential clients who pass close to the point of interests (POIs). Main objective of the application is to enhance the interaction between mobile clients and potential mobile clients with bank points (ATM, branch, merchants, etc.).

The application works as follows: the user gets the viewing angle (or framing) within the camera to be launched in SARAS by looking at a direction in shopping centres, on the street or on the road (highway, city). If there are POIs related to the back in the viewing angle (also inside framing), this information appears as a list on the screen. If a point is selected from the list, detailed information (such as details of the campaign) is displayed. If the user is in a multi-storey shopping centre, floor distinction is made by 3D effect. Besides, SARAS program recognizes an emblem of any POI (augmented reality) by using the barcode information on the emblem and shows the information about the barcode on the screen. Bank business units decide on the detail of information that can be displayed to the users. SARAS is an alternative channel for them.

In the scope of this thesis, the location information of the bank merchants is provided by the bank, and application uses this data. Due to security restrictions application cannot access to core banking database, instead a sample of the data from a pilot area is used. Due to this restriction, a new server is installed at Galatasaray University for the sample data. The application reaches this data via web services.

In spite of the increase in processing power, feature set, and sensing capabilities, the smartphones continue to suffer from battery life limitation, which hinders the active utilization of LBA's (location based application) (Zhenyun, 2011). Unfortunately, GPS, the core enabler of LBAs (Ben Abdesslem, 2009), is power-intensive, and its aggressive usage can cause a complete drain of the battery within a few hours (Constandache, 2009). Numerous solutions have been proposed to improve the battery life of the mobile devices (Anand, 2003), but little rigor and attention (Shih, 2002) have been devoted to the battery-efficient use of LBAs. The LBA developers are suggested to reduce the use of GPS by increasing the location-update intervals (say, to more than a minute), thus allowing GPS hardware to sleep between successive location-updates (Viredaz, 2003). Such a simple solution can improve battery life by forcing applications to request location information less frequently, but it has fundamental limitation (Zhenyun, 2011). In AR applications location fixes must be taken more frequently. At the same time, the other sensors must be working all the time.

The main contribution of this work is to analyse the energy and resource usage of AR applications, particularly the SARAS application. First of all a detailed performance analysis was performed to find the battery and power consumption by using Android platform performance tracing tools. It was observed that keeping the resources on and calling the location-distance function for each point of interest were the most CPU consuming factors. So, it was provided that closing resources on the activity changes and calculating location distance on the web service call. After these improvements the application performs %30 less energy consuming. At the same time different GPS and sensor processing, and display placement algorithms applied to the application and resource consumption were examined. In addition, by making comparisons with other applications, overall resource consumption on AR applications have been analysed.

Besides, the resource analysis of the SARAS application, the server side of SARAS, namely the database and the web services have been developed within this thesis. Additionally, at the client side, the QR-code scanning property of the mobile application has been developed as a contribution.

The remainder of the thesis is organised as follows. Section 2 gives the AR concept and the details of the SARAS application. Similar applications and related literature are presented in this section. Section 3 presents the key design principles and components of the SARAS application. Section 4 represents the detailed results of the performance analysis and comparisons, and Section 5 concludes the thesis.

2. AUGMENTED REALITY CONCEPT AND THE DEVELOPED APPLICATION: SARAS

The term augmented reality is used to describe a live direct or indirect view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data. AR is based on techniques developed in virtual reality and interacts not only with a virtual world but has a degree of interdependence with the real world (Mehdi, 2010). The hardware components like Head-Mounted Displays (HMDs) are used generally for augmentation (Janin, 1993). However, in order to avoid limiting AR to specific technologies, AR can be defined as systems that have the following characteristics: combines real and virtual; interactive in real time; and registered in 3-D (Mehdi, 2010). This definition allows other technologies, such as mobile technology, by using the fundamental components of AR (Azuma, 2001).

Tracking and the real time rendering are also the core parts of augmented reality systems as well as 3D registration and display (Bimber, 2010). That means augmented reality should be interactive in real time. When you try to achieve an augmented image in HMD based or a point of interest in GPS based, accurate tracking is very important, that because the user must see a believable image and he or she must be informed in precisely way (Mehdi, 2010). For a moving user it can be harder to get precise matches in all type of augmented reality. The virtual object position must be determined within the environment of the user. This provides the computer generated object appear to be fixed (Bimber, 2010). In GPS based systems, complete tracking with a global coordinate system can be required. In such cases the system must distinguish between outside or inside tracking (Zhou, 2008). For example, the GPS sensors are used to track and triangulating the position of the mobile device in outside environments. For the inside environments system uses internal sensors fixed to mobile objects; a camera for vision based tracking, digital compass to track which way the phone is facing, an accelerometer to track acceleration.

However, these systems both have their drawback, as GPS for example is not as accurate inside buildings as outside and vision based tracking depends heavily on lighting conditions and visibility (Shatte, 2014).

2.1 Mobile Augmented Reality

As computers grow in power and decrease in size, new mobile, wearable, and pervasive computing applications are rapidly becoming feasible, providing people access to online resources always and everywhere (Shatte, 2014). AR already presents a particularly powerful user interface (UI) to context-aware computing environments. AR systems integrate virtual information into a person's physical environment so that he or she perceives that information as existing in their surroundings. Mobile AR systems provide this service without constraining the individual's whereabouts to a specially equipped area (Alem, 2011).

Mobile AR based on AR principles in truly mobile settings; that is, away from the carefully conditioned environments of research laboratories and special-purpose work areas. A few technologies must be used to make this possible; global tracking technologies, wireless communication, location-based computing and services (Mehdi, 2010). However, other sensor technologies must be used to define calibration and coordination system truly like accelerometer or gyroscope. For location-based AR the position of objects on the screen of the mobile device is calculated using the user's position (by GPS or Wi-Fi), the direction in which the user is facing (by using the compass) and accelerometer.

Mobile AR applications have been used widely last years. Emerging technologies in mobile computing also support this. The developed application SARAS is also a mobile and location based AR application. In the next sections firstly SARAS and then other commercial applications will be examined in detail.

2.2 SARAS Project

2.2.1 Overview

For this thesis, an Android based augmented reality mobile application entitled 'SARAS (Sensor-based Augmented Reality Application Software)' was developed. SARAS application utilizes a location-based AR approach like Wikitude (WIKITUDE, 2008). It also supports marker-based approach by QR code property like the LAYAR browser.

Figure 2.1 SARAS Application

Basically, the aim of the application is to inform the user about YKB merchants and campaigns in these merchants. In addition, YKB branches and ATM points can be searched by using the application. The user gets the viewing angle within the camera to be launched in SARAS by looking at the direction in shopping centres, on the street or on the road (highway, city). If there are YKB points in the viewing angle, this information appears on the camera as buttons in relative to the location. A point is selected from the screen and detailed information (such as details of the campaign) is displayed. The colour of the point buttons varies depending on whether the campaign information or the categories, such as the type of the merchant: gas station, book store, etc. Users can filter the category or the distance info if they want from the menu in the application. SARAS also supports a maps view, QR code property for indoors, and different GPS modes. In the next parts, the main screen and menu sections in the application and the project components are examined in detail.

2.2.2 Screens and Menu

Figure 2.2 shows the main screen components of the SARAS application.

1. This is the button to access menu. A user can control filters, login, map etc. using this menu. To access this menu the user can also use your device's settings button if you have one.

2. This is how bank merchants are displayed on the screen. There are three categories and depending on these categories background colour of the button is changed.

3. This shows that the Google play services are active or not. Google Play Services are used to take GPS fixes. The details about it will be explained in Section 3.

4. These are the GPS modes. User can change the frequency of updates using this radio group. This is explained in Section 3.4.

Figure 2.2 Main Screen Components

5. This is a compass. It points to the North direction. When you click on it you access the calibration information page. This page shows how to calibrate your device. The user should calibrate your device when this compass is not correctly pointing North.

2.3 Other Commercial Applications

2.3.1 Layar

Layar is a global leader in AR and interactive print market (LAYAR, 2009). They have created a mobile browser called Layar App. The browser allows users to find various items based upon AR technology. As millions downloaded the Layar App for iOS and Android, this makes the Layar the world's most popular platform for AR.

The application uses accelerometer, built-in camera, compass and GPS. These are used together to identify the user’s location and field of view. From the geographical position, the various forms of data are laid over the camera view like inserting an additional layer.

Figure 2.3 Layar Application (LAYAR, 2009)

Data in the browser comes in the form of *layers*. Layers are REST web services serving geo-located points of interest in the vicinity of the user. Layers are developed and maintained by third parties using a free API. As of July 2015, Layar had over 6000 layers.

The Layer Content Sources that provide the content to be viewed in the Layar Reality Browser, like Flickr.com for example. Layer Content Sources are not necessarily separated from the Layar Service Providers, but in general is different logical entities, as existing geo-coded databases and web services don't support the Layar Developer API.

2.3.2 Wikitude

Wikitude is a mobile augmented reality technology provider (Wikitude, 2008). Founded in 2008, Wikitude initially focused on providing location-based AR through the Wikitude World Browser App. After then, by launching Wikitude SDK, it supports image recognition and tracking, and geolocation technologies.

Figure 2.4 Wikitude Application (WIKITUDE, 2008)

The Wikitude SDK is the Wikitude's core product. It includes image recognition and tracking, 3D model rendering, video overlay and location based AR. The Wikitude SDK is available for Android and iOS operating systems, and is optimized for several smart eyewear devices. The Wikitude application was the first publicly available that used a location-based approach to AR.

Content in the Wikitude World Browser App is mostly user-generated. Its content can be added by a web interface. In addition, web services are available to register the delivery of dynamic data.

2.3.3 Turkcell Pusula

Another location-based AR application sample is from Turkey. Turkcell is the leader GSM operator in country that provides many services to their customers. Turkcell Pusula is the one of them.

Figure 2.5 Turkcell Pusula App (TURKCELL, 2012)

Turkcell Pusula application provides user to find the nearby locations of the places like pharmacy, hospital, police stations, banks and restaurants. To find a location, location category is picked from the menu or can be searched by name. Then the places in this category are listed in maps view. By using radar view, the location of the places can also be appeared on camera view.

In Table 2.1, properties of these applications and differences from SARAS application are included as a summary.

Application	Data Source	QR Code Property	Licence	Resource Consumption
LAYAR	REST web services	Exists	Annual licence for the commercial use	Considered
Wikitude	User generated	Not	Free	-
Turkcell Pusula / En Yakın	Identified by Turkcell	Not	Free	-
Google Googles	User generated	Exists	Free	-
Field Trip	Web services	Not	Free	-
Tuscany +	Regional data	Not	Free	-
SARAS	Bank data	Exists	Free	Considered

Table 2.1 Similar Applications Comparison

2.4 Related Academic Work

In previous works in the literature, the general battery drain factors and common augmented reality application properties were discussed. For example, Mehdi Mekni - Andre Lemieux mentioned about the challenges of augmented reality applications (Mehdi, 2010). Methodologically they defined the principles of these applications and the challenges and finally provided the future trends. Also van Krevelen and R. Poelman talk about the basic principles, hardware and software components of the augmented reality technologies and limitations in their study (van Krevelen, 2010). They emphasize the human interaction part of the hardware components of these technologies. Finally, they talk about the usage areas and the limitations in hardware domain. Zhenyun et al. studied about the energy efficiency of location sensing (Zhenyun, 2011). Firstly they put forth of battery effects of location sensing then they suggest new location sensing methods. They claim that these methods improve the battery life by up to 75% by reducing the number of GPS invocations. This work is close to our study in terms of location based sensing. However, in our work, other factors in AR applications are also considered and different GPS and sensor processing methods are analysed in terms of energy consumption.

On the other hand, not only basic AR challenges but also mobile and Android operating system specific studies exist in the literature. Sarmiento et al. examined the performance of Android systems for AR applications (Sarmiento, 2012). They discussed the performance of synthetic image rendering, tracking sensors for indoor places and Android graphics in order. For the image capturing, they offer to use native code processor to improve velocity of execution and they analyse the performance with the different image formats and capture frequencies. For the tracking sensors (accelerometer, compass) they measured the values while the user is walking and for the GPS they analysed the network connections on the Android platforms. This part of the study is close to our work. But, they only investigated specific sensors using only one device. Since they use image rendering based augmented reality for the testing, they mostly concentrated on image capture and Android graphics performance. Finally, they presented the experimental results and conclusions over the basic test application. Wagner et al. also studied the AR on mobile phones over the image rendering based methods (Wagner, 2009). Similarly, they tested different formats of images and processing frequencies. They also examined the performance of rendering, memory, bandwidth usage and networking with different type of image rendering programs (OpenGL and Direct 3D). Another interesting research in this area is based on cloud computing (Chen, 2011). Chen et al. claim that AR over cloud computing has a great potential and the power of cloud computing can solve the limitation problems of AR applications. Some computational tasks can be done on the cloud with service as a server method. All this mobile-based AR researches are based on the image rendering conventional applications.

3. SARAS ARCHITECTURE

SARAS is a location-based AR application. It shows the POIs on the screen according to GPS information of the points and the direction of the user. In this application, POIs refer to Yapı Kredi merchants and branches. If any POI exists in the viewing angle, it is shown on the display. Display is the camera view of the mobile device.

The algorithm that is used in the application works as follows; i) finding the direction of user, ii) find out which point of interests in the viewing angle, iii) calculate the screen positions of them and shows them on the screen as buttons. They are shown on the screen with the different colours according the categories and existence of the campaigns. The detail of the algorithm can be examined in Erışık's thesis (Erışık, 2016). In this chapter, the core architecture of the application is mentioned and it is explained how the data is stored in the system and the data flow is examined in detail.

SARAS application uses the bank data. This data is stored in a remote database. Application reaches the data via RESTful web services. When the application is started, all data related to the points of interest are taken and loaded to the local phone database. According to the search filter parameters the points are searched in the local database with the algorithm mentioned before. For the detailed information of the points of interests, campaigns or branch queue info application calls the remote database services again. The architecture is shown in Figure 3.1.

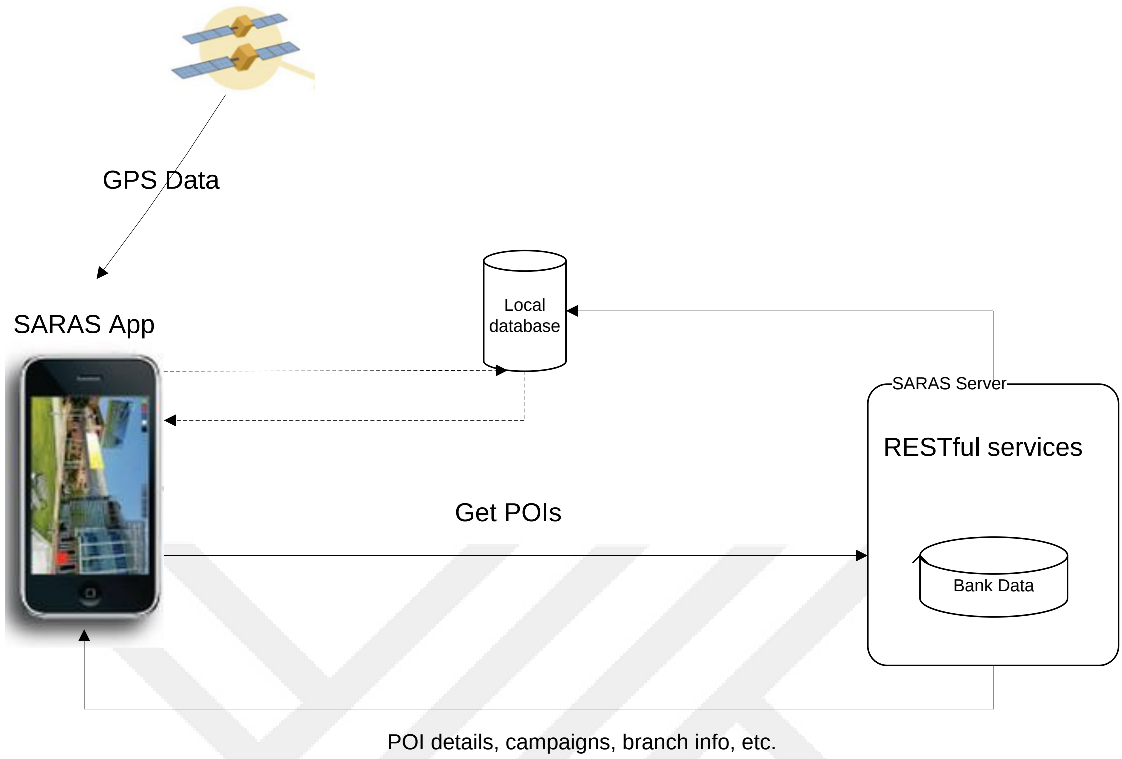

Figure 3.1 Basic Architecture of SARAS

The augmentation data source is the bank merchants and the campaigns. They are loaded via the restful web services. The remote server and the web services are the components of the application. Besides, the front camera, current location info and the direction of the device are needed. So device camera API's, graphics API and the motion sensors are used to achieve this. They must be working continuously in the application. For an extended battery lifetime and good performance, the resources must be used efficiently by the application.

3.1 Web Services

The content is provided by web services in the application. RESTful web services arecreated for this purpose. A server is located in Galatasaray University to deploy web services. Access permission to the 80 port of server is given. A remote desktop user is defined to connect and develop on the machine. The properties of the server are as follows;

- Intel Xeon 2.60 gHz processor
- 8 gb ram
- 64 bit Windows Server 2008 R2 Standard

MySQL has been installed on the server. MySQL Server 5.6 with Workbench 6.1 has been installed as well.

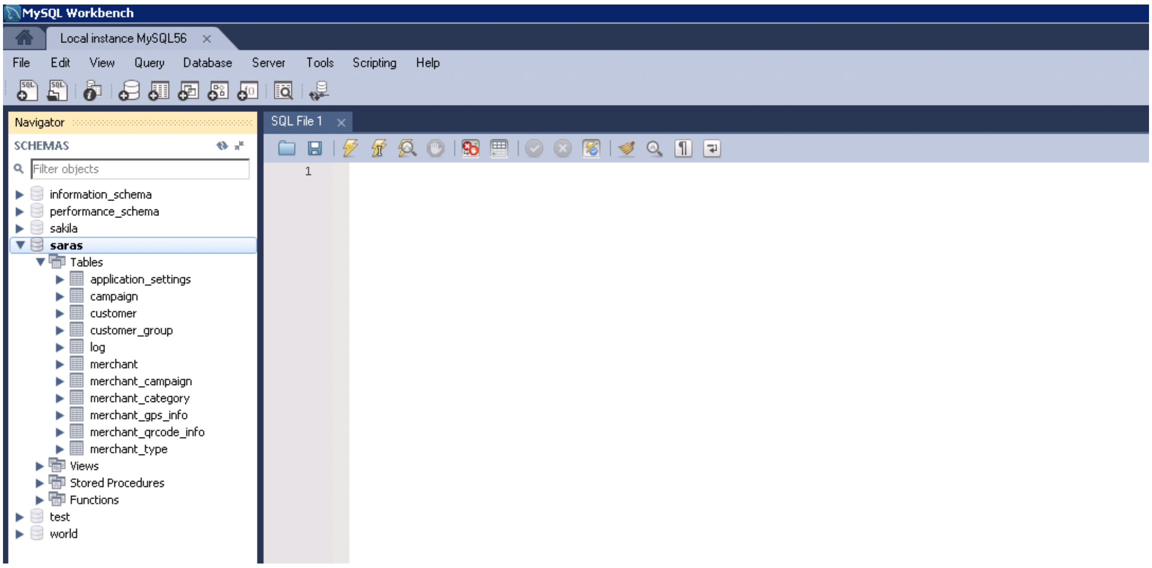

Figure 3.2 MySQL Workbench and SARAS Schema

The tables that are used in the project were created in the SARAS schema. A sample set of the Yapı Kredi Bank merchants and branch data have been loaded into merchant and category tables. In addition, a PHP frontend page is designed to add new data into the tables. Sample data is added to Customer and Campaign tables temporarily. The tables in SARAS schema are shown in Figure 3.3.

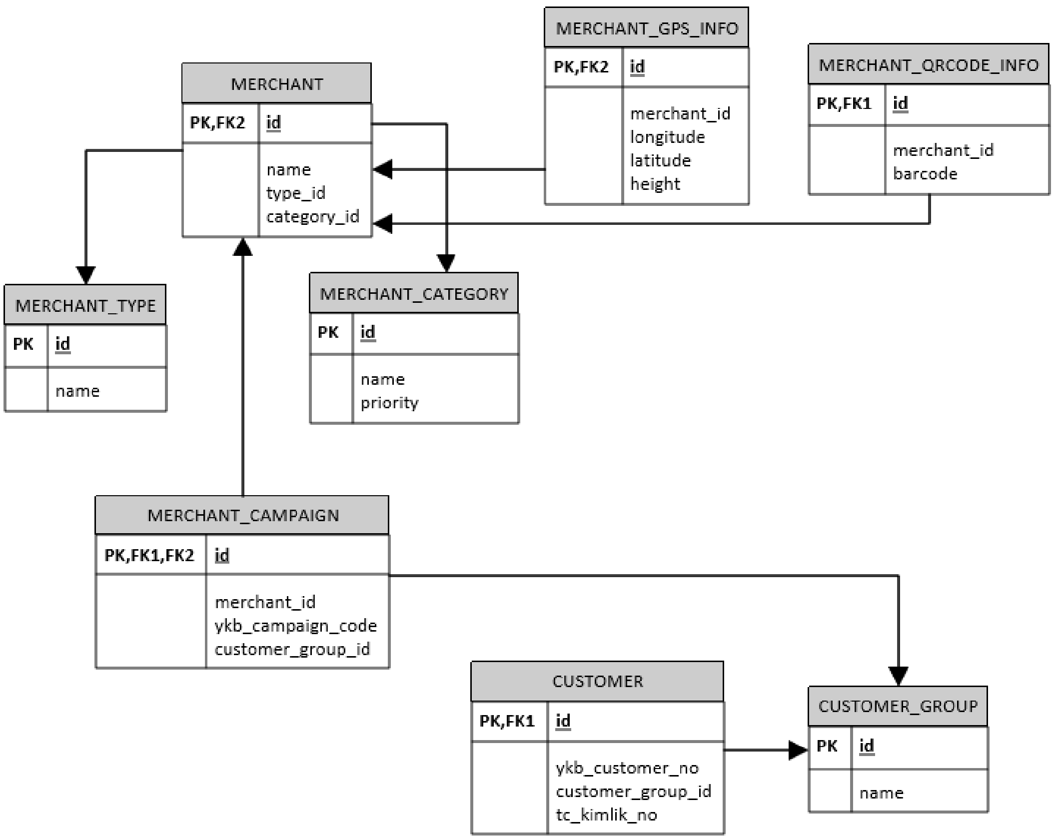

Figure 3.3 Tables in the SARAS Schema

MERCHANT

This is the main merchant table. It keeps merchant name and type/category information.

Name				
MERCHANT				
Column	**Tip**	**Not Null**	**PK**	**Purpose**
ID	NUMBER	Y	PK	Unique Identifier
NAME	TEXT	Y		merchant name
TYPE_ID	NUMBER	Y	FK	Refers to type table
CATEGORY_ID	NUMBER	Y	FK	Refers to category table

Table 3.1 MERCHANT Table

MERCHANT_TYPE

This table keeps merchant type information. Merchant types are merchant, branch or atm.

Name				
MERCHANT_TYPE				
Column	**Tip**	**Not Null**	**PK**	**Purpose**
ID	NUMBER	Y	PK	Unique Identifier
NAME	TEXT	Y		Type name

Table 3.2 MERCHANT_TYPE Table

MERCHANT_CATEGORY

This table keeps merchant category information. These categories can be defined by bank business. They are listed in the filter menu of SARAS application.

Name				
MERCHANT_CATEGORY				
Column	**Tip**	**Not Null**	**PK**	**Purpose**
ID	NUMBER	Y	PK	Unique Identifier
NAME	TEXT	Y		Category name
PRIORITY	NUMBER			Using for list ranking

Table 3.3 MERCHANT_CATEGORY Table

MERCHANT_GPS_INFO

This table is used to keep GPS information of merchants.

Name				
MERCHANT_GPS_INFO				
Column	**Tip**	**Not Null**	**PK**	**Purpose**
ID	NUMBER	Y	PK	Unique Identifier
MERCHANT_ID	NUMBER	Y	FK	Refers Merchant Table
LONGITUDE	NUMBER	Y		Longitude info
LATITUDE	NUMBER	Y		Latitude info
HEIGHT	NUMBER			Used for multi-storey places

Table 3.4 MERCHANT_GPS_INFO Table

MERCHANT_QRCODE_INFO

A unique barcode data is defined for each merchant that has QR code. This table keeps that data. QR code web service finds merchant information from this barcode data.

Name				
MERCHANT_QRCODE_INFO				
Column	**Tip**	**Not Null**	**PK**	**Purpose**
ID	NUMBER	Y	PK	Unique Identifier
MERCHANT_ID	NUMBER	Y	FK	Refers Merchant Table
BARCODE	TEXT	Y		Unique Barcode info

Table 3.5 MERCHANT_QRCODE_INFO Table

MERCHANT_CAMPAIGN

This is the main campaign table. The campaigns of the merchant are defined in this table. In the application merchant web service gets the data from MERCHANT_CAMPAIGN and MERCHANT_GPS_INFO tables. Campaigns can be defined to various customer groups.

Name
MERCHANT_CAMPAIGN

Column	Tip	Not Null	PK	Purpose
ID	NUMBER	Y	PK	Unique Identifier
MERCHANT_ID	NUMBER	Y	FK	Refers Merchant Table
YKB_CAMPAIGN_CODE	TEXT			Refers to bank campaign database
CUSTOMER_GROUP_ID	NUMBER		FK	Refers customer group table

Table 3.6 MERCHANT_CAMPAIGN Table

CUSTOMER

This table keeps customer information.

Name
CUSTOMER

Column	Tip	Not Null	PK	Purpose
ID	NUMBER	Y	PK	Unique Identifier
YKB_CUSTOMER_NO	NUMBER	Y		Refers to bank customer database
TC_KIMLIK_NO	TEXT	Y		Customer Id of Republic of Turkey
CUSTOMER_GROUP_ID	NUMBER		FK	Refers Customer Group Table

Table 3.7 CUSTOMER Table

CUSTOMER_GROUP

This table keeps customer groups.

Name				
CUSTOMER_GROUP				
Column	**Tip**	**Not Null**	**PK**	**Purpose**
ID	NUMBER	Y	PK	Unique Identifier
NAME	TEXT	Y		Customer Group name

Table 3.8 CUSTOMER_GROUP Table

XAMPP framework is used to prepare PHP web services. XAMPP is completely free; contains MySQL, PHP and Perl, open source and delivered by Apache. It can be installed for different servers like Windows, Linux. Detailed information can be taken from the link (ApacheFriends, 2011) and the download link can be accessed over there.

In first place, MySQL is downloaded and installed into the server. So MySQL in XAMPP is not used. XAMPP configuration and settings can be made from the control panel easily. It is sufficient to start Apache server in control panel for running PHP services and pages.

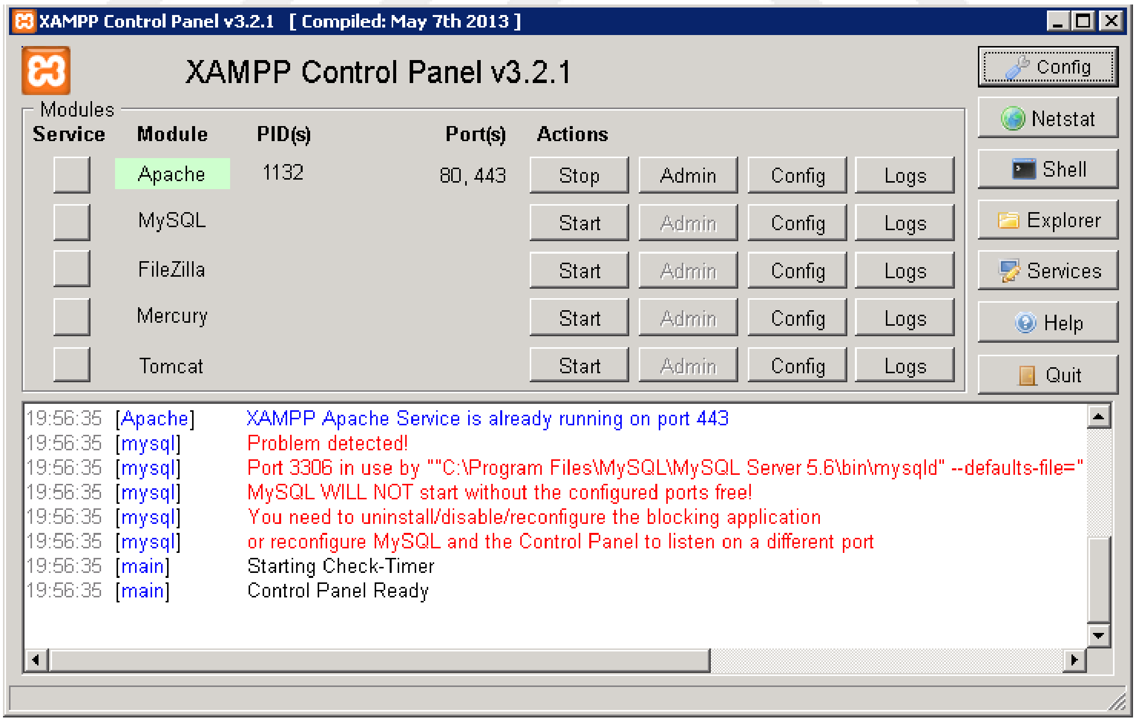

Figure 3.4 XAMPP Control Panel

PHP web services are prepared and published after the installation of the MySQL and Apache server. To create a new PHP page, it is required to put PHP extension file into xampp/htdocs directory. Two directories that named Services and Pages were created on the htdocs directory. PHP services were put under the Services directory.

RESTful web services were used in the project as mentioned before. It is very easy to prepare restful services with PHP. Json formatted data can be returned by json_encode function. The services prepared in UTF-8 encoded. At the beginning of the code UTF-8 meta tag is added. Firstly, connection control is made by providing MySQL connection, after that character encoding was set for queries and the data results were taken in array by running query. After that json object is created by looping in array data. Json formatted data is returned with json_object command. For all services in the project same structure was used. The services list is given as follows.

http://194.27.192.112/Services/getTypes.php -> Gets the merchant types.

http://194.27.192.112/Services/getCategories.php -> Gets the merchant categories (Types and Categories services were merged on the java layer.)

http://194.27.192.112/Services/getMerchants.php -> Gets the bank merchants by the GPS and distance filters. Latitude and longitude inputs are mandatory for this service. Distance and category inputs are optional. Distance info is 1000 meter if it is absent. If category filter is null, all type of merchants return. The algorithm that used in the service is shown in below (LTD is latitude, LON is longitude info).

```
SELECT m.id, m.name, m.category_id, c.name as category_name,
m.type_id, t.name type_name,
g.longitude, g.latitude,
( 6371 * acos( cos( radians(".$LTD.") ) * cos( radians( latitude ) ) *
cos( radians( longitude ) - radians(".$LON.") ) + sin(
radians(".$LTD.") ) * sin( radians( latitude ) ) ) ) AS distance
FROM merchant_gps_info g
LEFT OUTER JOIN merchant m ON g.merchant_id = m.id
LEFT OUTER JOIN merchant_category c ON m.category_id = c.id
LEFT OUTER JOIN merchant_type t ON m.type_id = t.id
WHERE ('". $TYPE . "' = '0' OR t.id IN (" . $TYPE . "))
HAVING distance < " . $DIST/1000 . " ORDER BY distance;
```

http://194.27.192.112/Services/getCampaigns.php -> Gets the merchant campaigns info. Merchant id is mandatory. The input parameters must be sent in GET type to the services.

http://194.27.192.112/Services/customerLogin.php -> Customer login service. It is prepared for login of the customers in application. The identity number is mandatory. It returns the customer data with the identity number entered.

http://194.27.192.112/Pages/merchant.php -> In addition to services, a simple php page is also prepared to add new merchant info. Merchant detail information like name, category, latitude-longitude information and address can be entered and saved via this page. Latitude-longitude information can be detected by google maps.

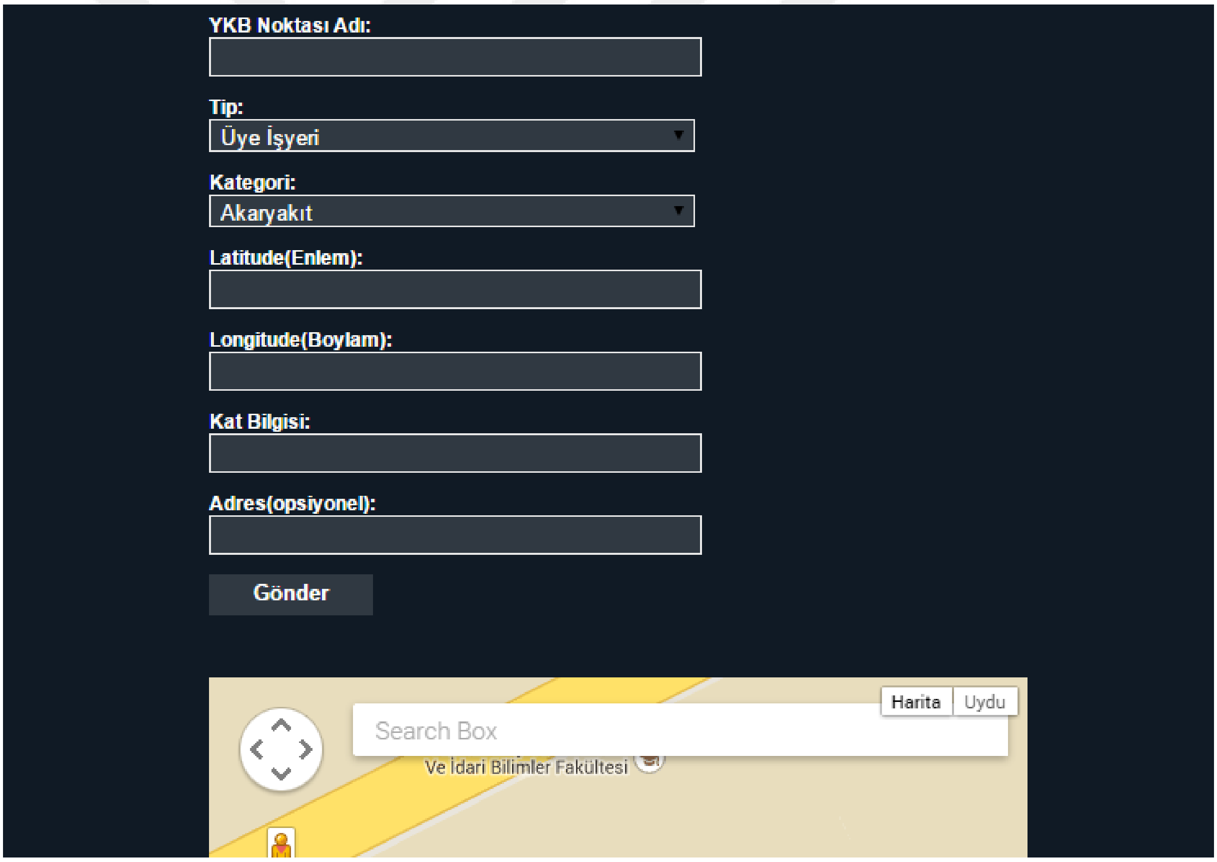

Figure 3.5 Add Merchant Page

3.2 Data Storage in Android

In the SARAS application, storing data locally is needed for two reasons:

1. *For keeping merchants information:*

Merchant information is fetched using web services when the program is started. However, connecting to a web service consumes battery and may be delayed. So keeping certain amount of merchants in the device's local storage can make the program run faster and save energy. For this storage saving a larger amount of data is needed and also it should be private to our program.

Therefore, merchants are fetched inside a diameter of 10 kilometres via a web service, and then they are kept in device's local storage. As the user changes location, if he gets out of the circle, the program reconnects and gets the data from the web service, clean the local database and refill it with the new merchants' information.

SQLite database is used for this objective. SQLite is an open source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation.

A database was created called saras.db using SQLite and the following tables were created using this database.

SARAS_MEMBER_SHOP

This table is used to store all the merchants inside a diameter of 10 kilometres. This table is filled using MerchantServices web service. It gets merchants of all types from the server. Type filter is not implemented in here because as the user changes his choices of type filter it is not needed to connect to a web service. Therefore, when he changes the filter for type the program connects to the local database and fetches new data quickly.

If it is the first time the application is being used, if the user presses the refresh button on menu, or if the user has changed the location and moved away 5 km from his first location this table is refreshed. The older values are dropped; with reconnecting to our server using the web service SARAS_MEMBER_SHOP table is refilled with new merchants which are located inside 5 km of diameter.

Name				
SARAS_MEMBER_SHOP				
Column	**Tip**	**Not Null**	**PK**	**Purpose**
MEMBER_SHOP_ID	NUMBER	Y	PK	Refers to Merchant id
MEMBER_SHOP_NAME	TEXT	Y		Refers to Merchant name
MEMBER_SHOP_LATITUDE	DOUBLE	Y		Refers to latitude of merchant
MEMBER_SHOP_LONGITUDE	DOUBLE	Y		Refers to longitude of merchant
MEMBER_SHOP_CATEGORY	TEXT	Y		Refers to category of merchant
MEMBER_SHOP_TYPE_ID	NUMBER	Y		Refers to type of merchant
COSLAT	DOUBLE	Y		Used for calculating distance
SINLAT	DOUBLE	Y		Used for calculating distance
COSLNG	DOUBLE	Y		Used for calculating distance
SINLNG	DOUBLE	Y		Used for calculating distance

Table 3.9 SARAS_MEMBER_SHOP Table

SARAS_MEMBER_CATEGORIES

This table is used for feeding category filter on the user menu. As the program is being opened first time, connects to the server using the web service. All the possible distinct type available are fetched and the table is filled with these values. So, the menu option for type filter can be filled using this table.

Name				
SARAS_MEMBER_CATEGORIES				
Column	**Tip**	**Not Null**	**PK**	**Purpose**
CATEGORY_ID	NUMBER	Y	PK	Refers to category id
CATEGORY_NAME	TEXT	Y		Refers to category name
CATEGORY_PRIORITY	NUMBER	Y		Refers to category priority

Table 3.10 SARAS_MEMBER_CATEGORIES Table

2. *For keeping filters chosen by user:*

User can change distance and type filter, and the program should remember these selections and gather data using these filters. So for this storage a smaller amount of data is needed to save and it should be private to our program. So Shared Preferences were used in the application.

Shared Preferences

Shared Preferences are used to save a relatively small collection of key-values. A Shared Preferences object points to a file containing key-value pairs and provides simple methods to read and write them. Each Shared Preferences file is managed by the framework and can be private or shared. The private one was used in application.

Following variables are created with Shared Preferences:

1. User's first entry to application control: SARAS_FIRST_ENTRY

This control is used to bring in front the login screen. If it is the first entry, meaning if this value is null, an alert dialog is shown for login. This screen asks for Turkish citizenship identity number. As the user enters his identity number, the program checks if the user is a bank customer or not. The campaign information can be different for bank customer groups.

If user is bank customer; SARAS_CUST_LOGIN parameter is saved using SharedPrefs with parameter value = 1. SARAS_CUST_GROUP_ID parameter is saved using SharedPrefs with parameter value = customer's group information. If user is not bank customer; SARAS_CUST_LOGIN parameter is saved using SharedPrefs with parameter value = 0. SARAS_CUST_GROUP_ID parameter is saved using SharedPrefs with parameter value = 0.

2. Distance Filter

Distance chosen by the user is also saved using Shared Preferences. Parameter's key value is DISTANCE_VALUE. Thus user's choices do not be losed even if the program is terminated. And also this parameter's value can be get inside different java classes. So during the program this parameter's value is used in web services and distance filter alert dialog.

3. Type Filters

Categories chosen by the users are also kept using Shared Preferences. For functionality of the program, all distinct categories are fetched at the initialization. They are loaded into our SQLite local database.

Afterwards our type filters are feeded with these values kept in our local database. But they should be kept in mind type filters chosen by user. They should not been lost even if the program is terminated.

For this reason Shared Preferences are used. All the categories are saved with keys with following format:

TYPE_VALUE_CATEGORY_NAME (for example for ATM it is TYPE_VALUE_ATM), and if it is chosen its value is 1 and if it is unchecked its value is 0. These parameters were implemented using this generic structure. Therefore, the new categories can be added in program and they do not lose older values.

4. Location Information

Our local database should be updated as the user changes his location. Therefore at the time that data is fetched from the web service the device's GPS values should be saved. Using MYFIRSTGPS_LONGITUDE and MYFIRSTGPS_LATITUDE parameter keys, latitude and longitude values are saved. Continuously the difference is calculated between these first values and the current GPS values. If this distance is bigger than (UPDATE_MIN_VALUE = 10000 meters) web service is called again, our table is cleaned which is keeping merchant information and refilled with new values. U sing Shared Preferences our new GPS (MYFIRSTGPS_LONGITUDE, MYFIRSTGPS_LATITUDE) values were written over old values.

3.3 QRCode

SARAS application supports the QRCode scanning property. In indoor places where the GPS is not working properly, the merchants have a QRCode placed on the YKB posters on their display windows. And if the user wants to see the campaigns in a merchant, he will use the application's QRCode property.

Zxing QRCode library (ZXING, 2015) was used in the thesis to read QRCodes. A new android activity that reading QRCodes with using this library was created. BarcodeScannerActivity activity was extended from CaptureActivity class. It is defined in AndroidManifest file also. It was provided that opening this activity via QRCode menu in application.

Figure 3.6 QRCode Reader

The QRCode information of the merchants is kept in MerchantBarcodeInfo table in the database. A new RESTful web service was created to access this info from the mobile application. Java layer that calls this web service was also prepared in application. Web service information is on the below.

http://194.27.192.112/services/getCampaignsQrcode.php?barcode=aaa111

A unique barcode code is necessary to call this service. For test purposes, “aaa111” coded record was defined in the database and it was connected to a merchant that has a campaign. By calling the service in this way merchant and campaign info are returned from the service. It was prepared that one campaign information returns for each call.

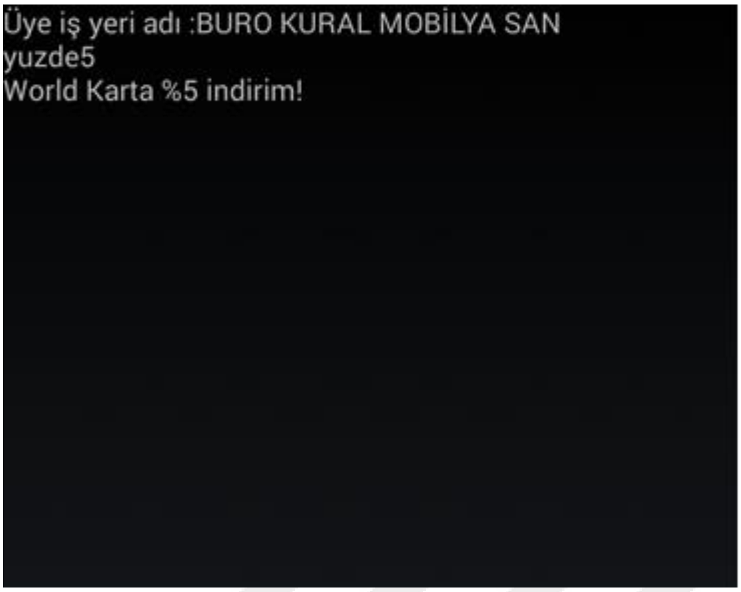

Figure 3.7 Merchant and Campaign Info in QRCode

3.4 Other Components

One of the main components in an augmented reality application is the camera. Device camera is always open in the application's main activity. The camera display is shown in the screen and the augmented data are placed on this display. In the SARAS application; camera display is created by using SurfaceHolder object. Finally, camera is released after using it for other applications. For accessing the camera permissions must be added to AndroidManifest file.

The other basic structure of augmented reality applications is to find the user's location. The application finds near point of interests according to this information. Location information is obtained using GPS. The GPS (Global Positioning System) is a space-based navigation system that provides location and time information in all weather conditions, anywhere on or near the earth where there is an unobstructed line of sight to four or more GPS satellites (Wikipedia, 2015). The system is used in many different areas like military, civil and commercial. It is freely accessible with any GPS receiver.

In the SARAS application; the GPS receiver is an Android device. Android location library is used for GPS properties. An available Google Play Services API is used if the device supports. Google Play Services provide google-powered features such as Maps, Google+, and more. Again permissions must be added to manifest file to access location libraries and google play services.

Three different GPS modes were added on the SARAS application for both Google Play Service version and Android Location Library versions. These modes are;

1. *Normal Mode*

This mode is the default mode. Application starts in this mode. The application is forced to take updates every 20 seconds. This should consume less energy. Detailed results will be explained in Section 4. This mode can be use when the user is walking.

2. *Drive Mode*

The application is forced to request updates every 1 seconds. This should definitely consume more energy. This mode can be used when the user is driving or moving quickly. But as the devices are not able to take location updates every 1 second, this mode's update range changes from device to device. For HTC XLARGE minimum update range is every 10 seconds and on SAMSUNG S4 MINI it is 15 seconds.

3. *No Updates Mode*

The application does not request location updates. This should consume less energy. This can be used when the user is stable. It assumes that the user is at the same place and make its operations based on the last GPS information taken.

The use of different GPS libraries and the different usage modes were tested. Detailed test results will be explained in Section 4.2. As mentioned before, these modes are only created for test purposes. Best way to achieve our goal for us is to take location updates as the user has displaced minimum 1 meter.

The other components are the sensors in augmented reality applications. To determine the orientation of an Android device it is benefited from sensors. Motion sensors like accelerometer, gyroscope and position sensors like geomagnetic field are used for this purpose. The SARAS application uses the orientation vector that is a combination of geomagnetic field and accelerometer. Figure 3.8 shows a summary of these.

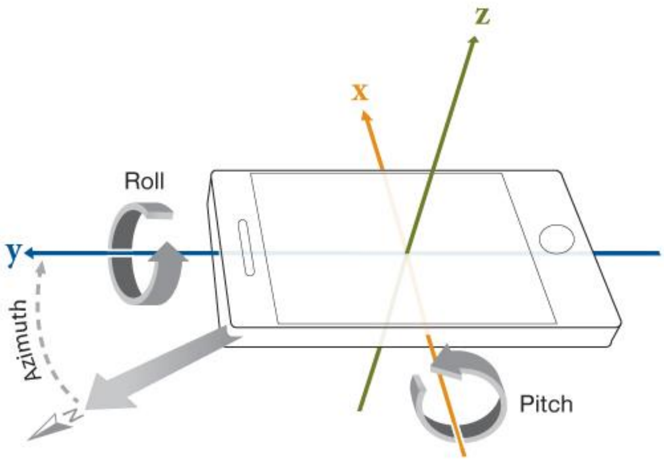

Figure 3.8 Orientation Vector (Orientation Vector, 2015)

4. PERFORMANCE ANALYSIS AND TEST RESULTS

The main purpose of this work is the energy efficiency analysis of the AR application, SARAS. Firstly, the SARAS application has been made energy efficient by putting basic controls. Next, the performance tests were applied in different modes. Finally, the comparison tests were done with similar commercial applications. This section gives the details of all these analysis and test results.

4.1 Basic Controls

The power consumption must be under control in mobile applications due to the battery limitations. Therefore, the resources, such as the sensors, connections, that had been activated before must be closed when the application or the activity finishing. This can be provided with Android activity lifecycle. Also the battery status must be checked while the application is running. Some functions must be finished if the battery level falls under a certain level. Even the application must be closed in very low level of battery.

For these reasons in SARAS application the battery status is controlled consistently and actions are taken according the battery level. *PowerStateChangedReceiver* named android receiver is coded for this control. At the start of the main activity, this receiver is registered. So the battery level is controlled while the application is running. *ACTION_BATTERY_CHANGED* Android intent is used for the battery control. This intent is passed to *isLowBattery* method in the Utility class and the battery level is controlled in there. Boolean variable is returned from the function. In the same way if it falls a higher level (currently it is %15), user confirmation is needed to open QRCode reader activity.

Figure 4.1 User Confirmation in Low Battery

The resources that are used in the *mainActivity* and the other activities that can be opened by menu, like *BarcodeScannerActivity* or *MapViewerActivity*, are closed in onPause () function by the scope of application life cycle. With the *onResume ()* function these resources are opened again. While passing to other activities several Boolean flags are defined in mainActivity. Thus the opening QRCode or map activity are understood and the resources are closed properly. These improvements decrease the battery consumption. Before closing the resources, the application was observed to be consuming %15 battery in 20 minutes, continuously operating with HTC One Mini. With these improvements, it consumes only %13. But the main impact of the development was seen at the background running. For 40 minutes test of background running, the CPU consumption decreased to 46 j from 183 j and the power consumption decreased to 36 mW from 90 mW. The detail test results can be seen at Figure 4.2.

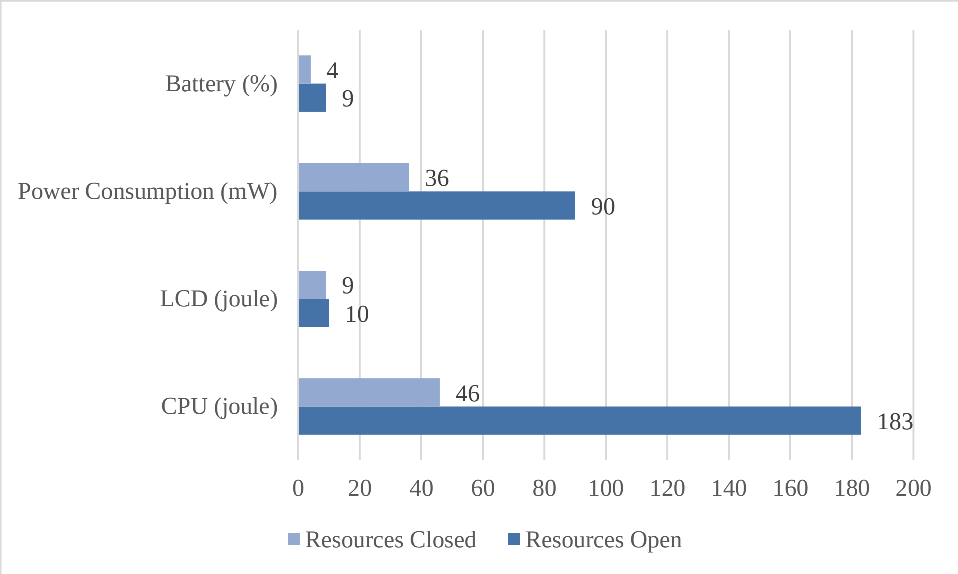

Figure 4.2 Background Running Test Results

4.2 CPU Profiling

The detailed performance analysis is performed after the basic controls. In this section the methods are explained that utilized for CPU profiling at thread level first and show the analysis of the profiling with the improvements added to the SARAS application as a result of this analysis. In Android four methods can be used for this profiling.

4.2.1 StrictMode

This control is used to control out the long processes in the main thread. If a long process or the direct network accesses the main thread and this control is open, Android throws exception and terminates the process.

4.2.2 Developer Options

In Android devices the developer options menu can be activated to follow the application performance or the power consumptions. For example by the “show CPU usage” menu CPU consumptions can be watched instantly.

4.2.3 LINT

Lint is an Android analysis tool for the common errors in Eclipse projects. It can be reached Android Tools -> Run Lint: Check for Common Errors by the right click on the project in Eclipse. For the SARAS project this is run and the errors, shown in Figure 4.2, are fixed. For example, SparseArray<String> using instead of Hashmap arrays or Float.valueOf(x) using instead of Double variables effects CPU, fixing layout problems effects LCD performance.

205 errors, 225 warnings

Description	Category	Location
"country_codes" is not translated in "af" (Afrikaans), "am" (Amharic), "ar" (Arabic), "be" (Belarusian)	Correctness:Messages	arrays.xml:3 in values (CaptureActivity)
Class referenced in the layout file, com.google.zxing.client.android.ViewfinderView, was not found i	Correctness	capture.xml:23 in layout-ldpi (CaptureA
The locale folder "he" should be called "iw" instead; see the java.util.Locale documentation (2 items	Correctness	tez
Not targeting the latest versions of Android; compatibility modes apply. Consider testing and updat	Correctness	AndroidManifest.xml:7 (tez)
The <activity> ar.AugmentedActivity is not registered in the manifest (7 items)	Correctness	AugmentedActivity.java:29 in ar (tez)
To get local formatting use getDateInstance(), getDateTimeInstance(), or getTimeInstance(), or use	Correctness	Database.java:193 in sqlite (tez)
Invalid layout param in a RelativeLayout: layout_weight	Performance	activity_main.xml:13 in layout (tez)
Custom view ar/AugmentedView has setOnTouchListener called on it but does not override perforn	Accessibility	AugmentedActivity.java:59 in ar (tez)
Replace "..." with ellipsis character (…, …) ? (6 items)	Usability:Typography	strings.xml:19 in values-tr (tez)
Found bitmap drawable res/drawable/launcher_icon.png in densityless folder (2 items)	Usability:Icons	launcher_icon.png in drawable (Captur
Use new SparseArray<String>(...) instead for better performance (3 items)	Performance	Utility.java:37 in core (tez)
Use Float.valueOf(20) instead (3 items)	Performance	ARData.java:33 in ar (tez)
Missing the following drawables in drawable-mdpi: launcher_icon.png (found in drawable-xhdpi, di	Usability:Icons	tez
Use a layout_height of 0dp instead of wrap_content for better performance (2 items)	Performance	capture.xml:39 in layout-ldpi (CaptureA
Nested weights are bad for performance	Performance	search_book_contents.xml:32 in layout
Possible overdraw: Root element paints background @color/encode_view with a theme that also pa	Performance	encode.xml:21 in layout-land (Capture/
The resource R.layout.app_picker_list_item appears to be unused (158 items)	Performance	app_picker_list_item.xml in layout (Cap
The following unrelated icon files have identical contents: common_signin_btn_icon_disabled_focu	Usability:Icons	common_signin_btn_icon_disabled_foc
[Accessibility] Missing contentDescription attribute on image (7 items)	Accessibility	app_picker_list_item.xml:7 in layout (Ca

Figure 4.3 Runnig LINT for SARAS

4.2.4 Systrace

While the application is running it can be started by selecting the “Start Method Profiling” on the running thread on DBMS. It provides results of the CPU consumptions in function level. So it can be checked that which function in the application consumes more CPU.

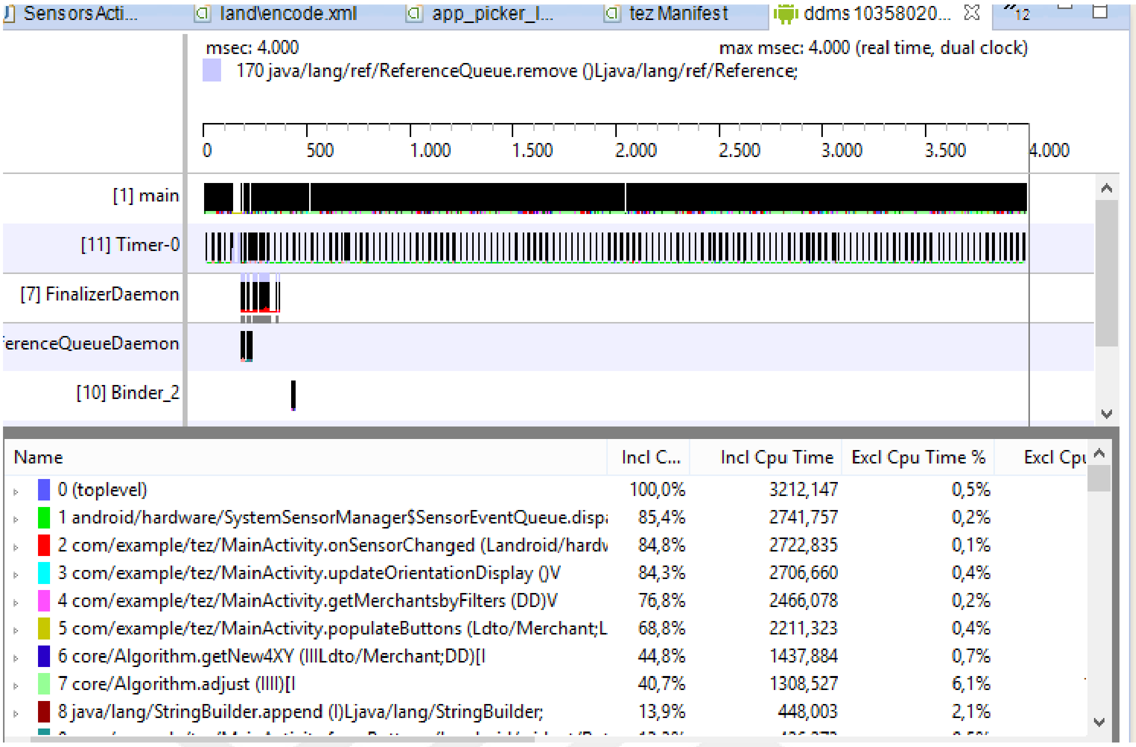

Figure 4.4 Tracking the CPU by functions in SARAS

Considering the results of the systrace, it can be seen that the sensors and screen placements functions consume more CPU compared to the other functions. It is expected because these functions are always running in the application life cycle. Hence, all these functions are examined again after these results. And improvements are made in *sensor running*, *getMerchantsByFilters* and *populateButtons* functions. For example; all information in the merchant web service is fetched to a local database at first and then they are filtered by the parameters in the menu (getMerchantsByFilters). Before, the distance filter control had been provided with the distance calculation for each point of interest by the android GPS library. It effected the performance negatively. So the SQLite query that searches the local merchant data has been improved with the distance info calculation. To provide this, COSLAT, SINLAT, COSLNG, SINLNG columns are added to the MERCHANT table. These are the cosine of the latitude, the sine of the latitude, the cosine of longitude and the sine of the longitude in order. These values are calculated and written to the table when the merchant information is taken from the web service. Equation 4.1 is used for the calculation.

$$(deg * Math.PI \,/\, 180.0); \text{ deg is the latitude or the longitude (1)}$$

In select query these pre calculated values are used with the other GPS information to figure out the distance. And the results are filtered with this distance information. Distance values are used in kilometre terms.

At the same time the bank merchant web service that is used in the application was returning just the GPS and name information. In the map view activity for the every merchant the campaign information were taken from the other web service. To prevent this, a Boolean flag that shows the campaign existed in the merchant is added to the merchant web service. In the map, points of interests are shown in different colours according to this flag. The strings that had been put on the screen for debugging purpose was removed. All these improvements contributed to decrease CPU and LCD consumption in application. While in three minutes short running of application the test results are close to each other, in long term running the results are more evident. The CPU consumption decreased to 266 j from 360 j and the LCD consumption decreased to 890 j from 1100 j for twenty minutes of usage.

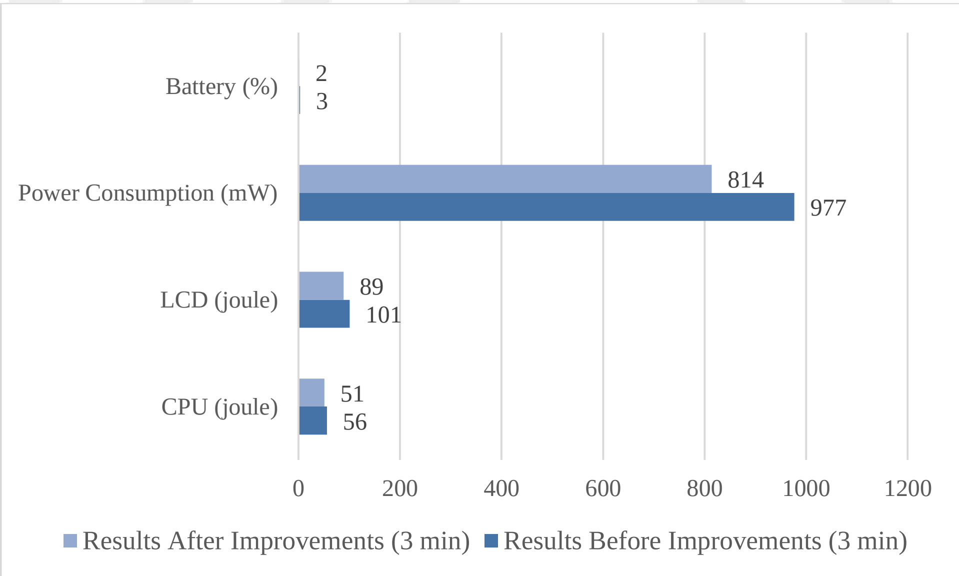

Figure 4.5 Performance Results after Improvements (3 min)

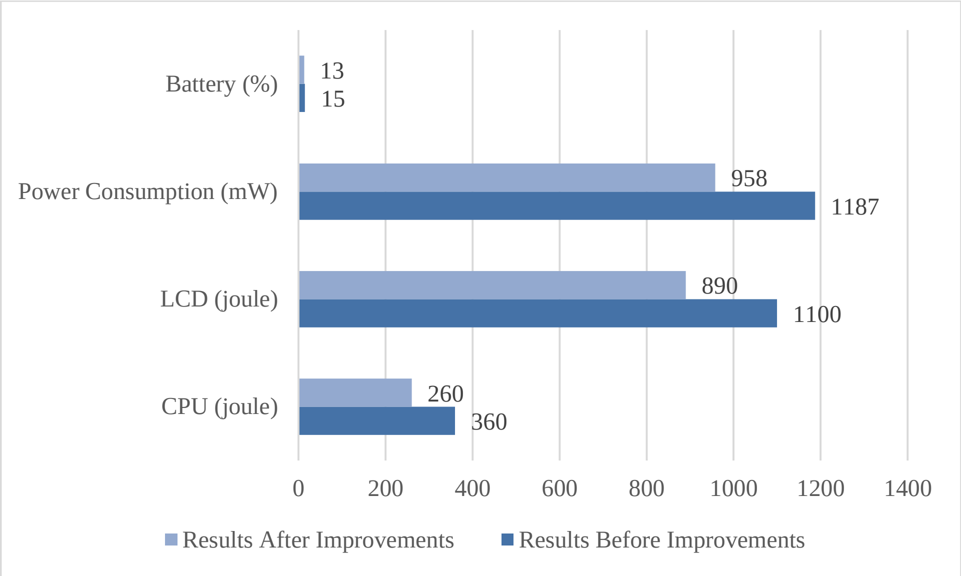

Figure 4.6 Performance Results after Improvements (20 min)

Figure 4.5 and 4.6 shows the results after performance improvements in 3 minutes and 20 minutes. These results were taken in normal mode with the google play services support. The battery in percentage, the CPU and LCD in joule and the total power consumption in mW unit.

The Power Tutor application (PowerTutor, 2011) was used for the measurements. PowerTutor is an application for Android platform phones that displays the power consumed by major system components such as CPU, network interface, display (LCD), and GPS receiver and different applications. The application allows software developers to see the impact of design changes on power efficiency. Application users can also use it to determine how their actions are impacting the battery life. PowerTutor uses a power consumption model built by direct measurements during careful control of device power management states. This model generally provides power consumption estimates within 5% of actual values. A configurable display for power consumption history is provided. In this work, PowerTutor is used to analyse the CPU, display and battery life for 20 minutes usage of SARAS and the other similar applications.

4.3 Performance Analysis with Different Settings

As mentioned in previous sections the SARAS application supports two GPS methods. Google play services and Android library. The application can be run in different sensor settings like normal mode or drive mode. The application was tested in these different methods and modes. Different connection types and mobile devices were used for testing. HTC One Mini mobile phone were used in general comparisons. Nexus 7 Android tablet was used in device comparison. The effects on the performance were examined of them. Similar algorithms and applications like LAYAR were tested. Now, all these test results will be examined in detail as comparative way.

4.3.1 Google Play Service & Android Location

SARAS application uses Google Play Services library to take GPS fixes as default. But, some devices do not support this library. In this case application uses the Android Location library. First of all these two GPS methods are tested in terms of resource consumptions. The test results are shown in Figure 4.7.

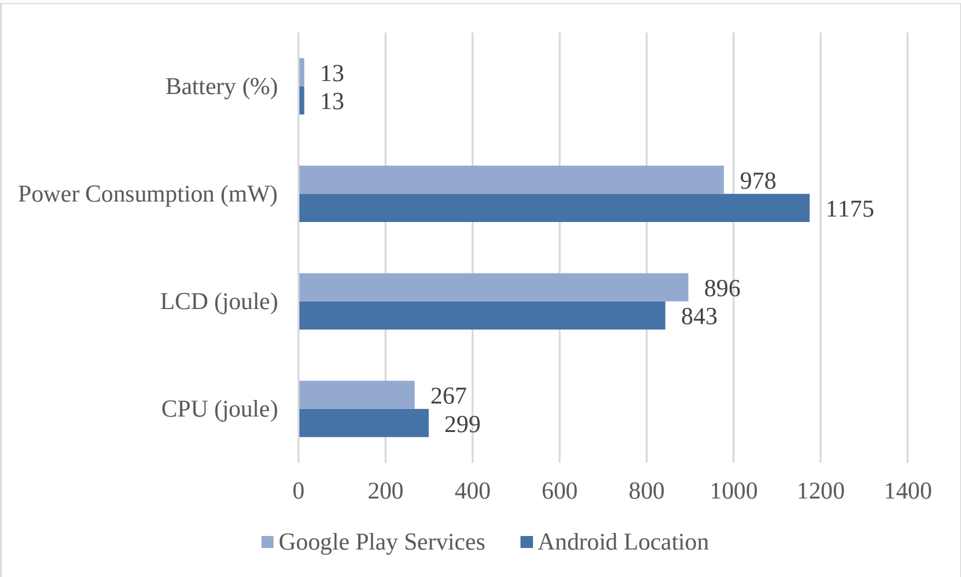

Figure 4.7 Comparison of Google Play Service & Android Location

As shown in Figure 4.7, using Google Play Service library consumes less CPU and power consumption. LCD values change according to the number of points of interest displayed on the screen. These values are close to each other. The only exception is the battery life. Battery life does not differ between the two library usages.

4.3.2 Normal Mode & Drive Mode

People can use the SARAS application while walking or going by car or any other transportation method. GPS update intervals must be different in these different cases, as explained in Section 3.4. SARAS takes the GPS fixes 10 times faster in the drive mode. Figure 4.8 shows the test results of this comparison.

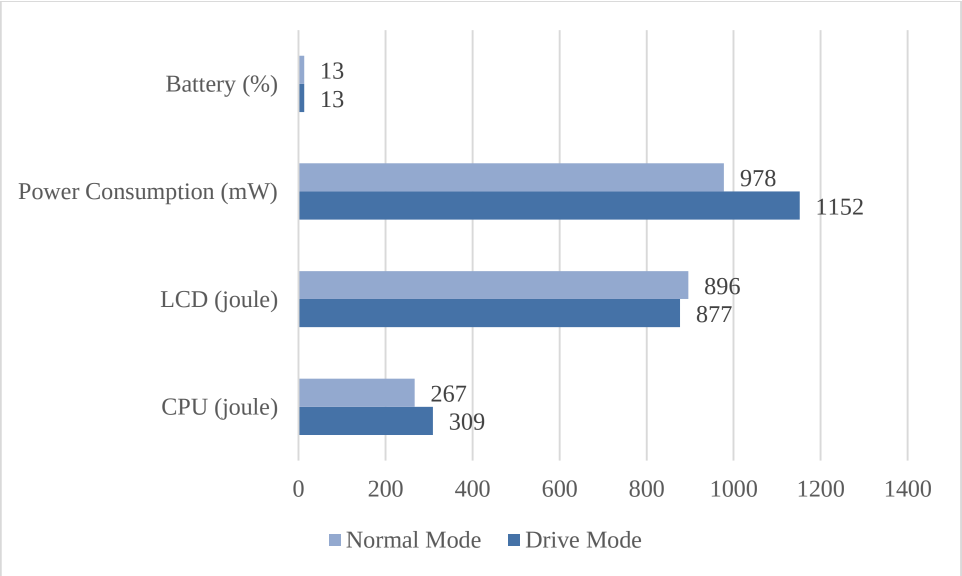

Figure 4.8 Comparison of Normal Mode & Drive Mode

As shown in test results GPS update intervals effects the performance in inversely proportional. But, LCD results doesn't change much in this case. These results are taken with the Google Play Services support and WIFI internet connection. With Android location library using in drive mode consumes 315 j of CPU and 1184 mW of power.

4.3.3 Connection Type

Connection type is also effective on the energy consumption on AR applications, because the application needs internet connection for running. So, SARAS application was tested on WIFI and 3G connection types. The results show that on 3G connection battery and CPU consumption is much more than the WIFI connection. Detailed results can be seen in Figure 4.9.

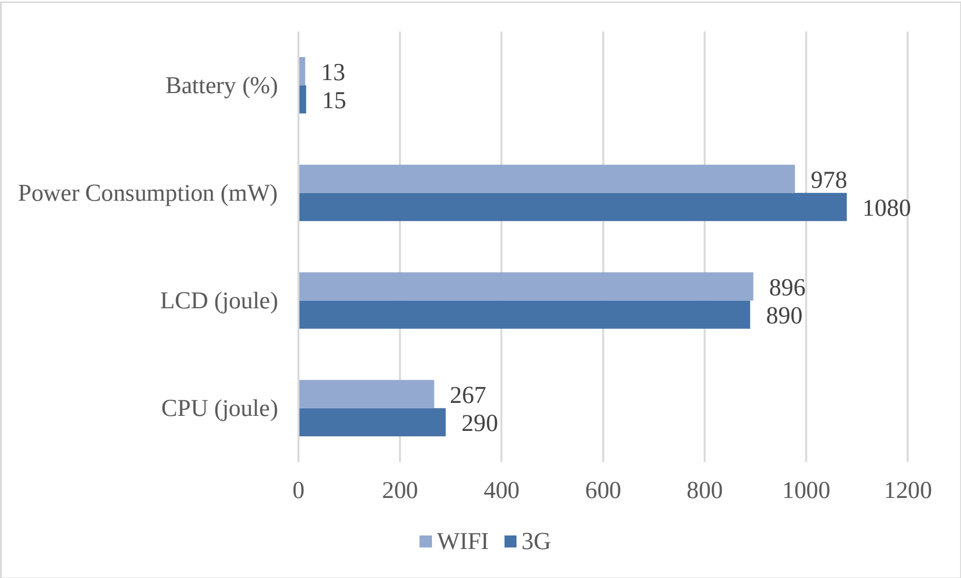

Figure 4.9 Connection Type Comparison

The reason of this is that the device connects to a remote base station while using mobile data and this connection ports are always changed on the move. Whereas in the WIFI connection device connects to a closer connection device (access point).

4.3.4 Web Services

SARAS application calls the merchant web service at the start of the application. It is not called again while the user changes location by 10 km as a default value. To analyse the effect of the web services on the energy consumption, tests were conducted with different web service call frequencies. For example, a test is conducted by calling in minutes or calling at a lower range of location changing in drive mode. For 20 minutes, test interval merchant web service was called 20 times in the first case. By movement of the car in drive mode, it was called 4-5 times.

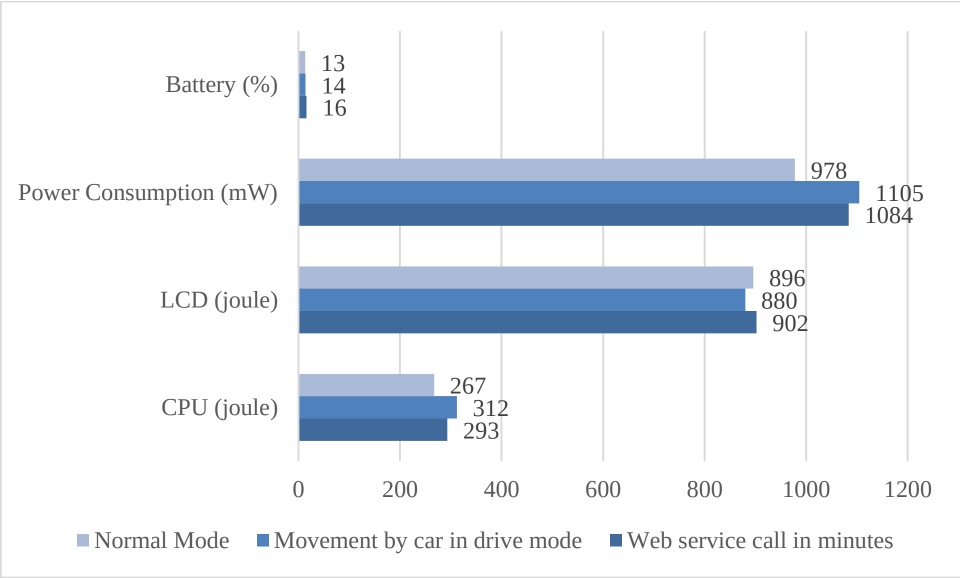

Figure 4.10 Effect of Web Service Call Frequency on Energy

Figure 4.10 shows that battery consumption is growing while web service call frequency increases. In the drive mode, 3G connection type is used. So, CPU and power consumption values are much more than the other case.

4.3.5 Testing with Android Tablet

SARAS application also was tested on Android tablet to see the impact on performance of the device characteristics. Nexus 7 Android tablet is used for this test. On this device screen size is larger and the camera resolution is worse than the HTC One Mini. Furthermore, it has more powerful processor. So the energy consumption parameters are observed to be much better in this device. Figure 4.11 shows the test results.

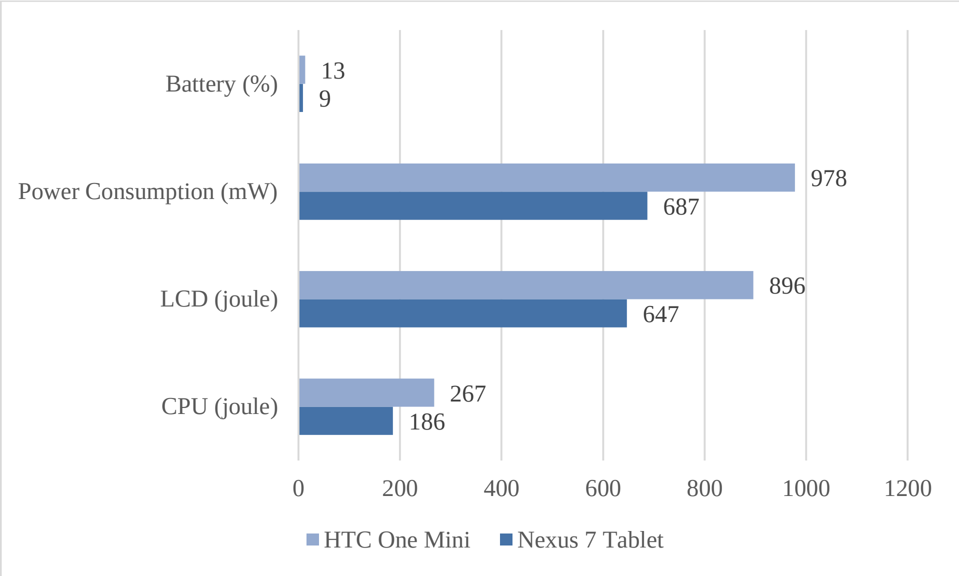

Figure 4.11 Testing with Android Tablet

4.3.6 Comparison with Other Applications

Other similar applications like LAYAR were also tested by the Power Tutor. Tests were done with same device and connection type. The augmented reality algorithm in Pro Android Augmented Reality book (Raghav, 2012) was applied to another main activity of SARAS for the test purposes. Another main activity was created to implement and run this algorithm. In this algorithm, screen replacement is provided by augmented reality named object. And in sensor changed event the objects' positions are calculated again. When looking at the results, given in Figure 4.12, this algorithm consumes more battery, but less CPU. Because more often but doing less work processes are existing. But, the results are very close to each other.

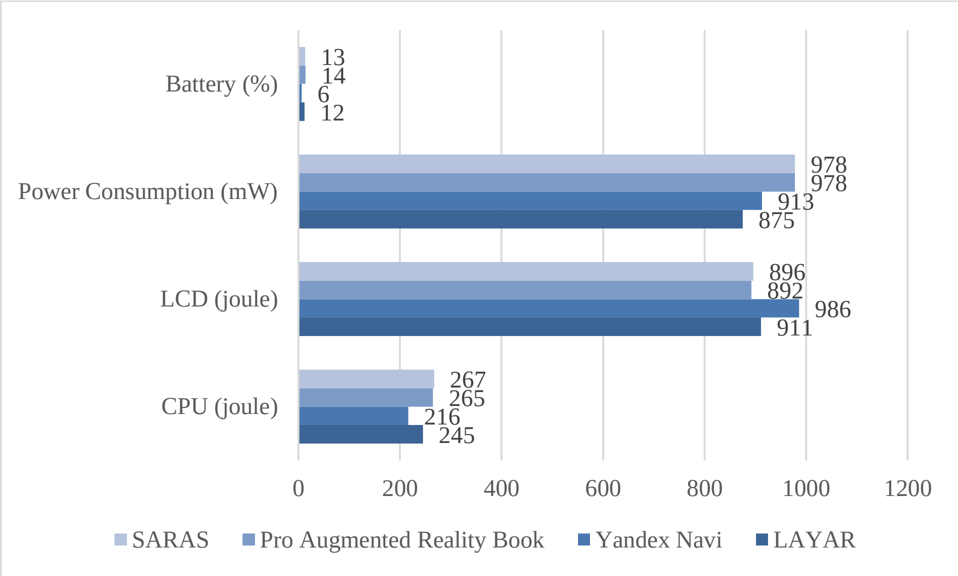

Figure 4.12 Comparison with Other Applications

This is similar for the results of other applications. They consume less battery. In the Yandex Navi application test any route calculations did not be made. So, this is the reason of the less battery consuming results. SARAS application is also tested in just Google Maps mode to compare with Yandex Navi. It consumed only 58 j of CPU and %9 of battery. Test results show that SARAS performs in close to commercial applications like LAYAR and the professional algorithms in terms of resource consumption.

All these test results are shown as a summary in Table 4.1.

Application	GPS Type	Usage Mode	Conne ction	Power Cons.	CPU Cons.	LCD Total	Battery %
SARAS	G.P.S	Normal	WIFI	978 mW	267j	896j	13
SARAS	A.L	Normal	WIFI	1175 mW	299j	843j	13
SARAS	G.P. S	Drive	WIFI	1152 mW	309j	877j	13
SARAS	A.L	Drive	WIFI	1184 mW	315j	850j	13
SARAS	G.P. S	Normal	3G	1080 mW	290j	890j	15
SARAS (inc. Web service call)	G.P. S	Normal	WIFI	1084 mW	293j	902j	16
SARAS(web service)	G.P.S	Drive	3G	1105 mW	312j	880j	14
SARAS(Android Tablet)	G.P.S	Normal	WIFI	687 mW	186j	647j	9
LAYAR App			WIFI	875 mW	245j	911j	12
Yandex Navi App			WIFI	913 mW	216j	986j	6
Yandex Navi App		Route calc.	WIFI	962 mW	348j	980j	10
Pro Android Augmented Reality			WIFI	978 mW	265j	892j	14

Table 4.1 Performance Comparison in Different Modes and Applications

5. CONCLUSION

As a conclusion, in this work the main focus was on the energy consumption analysis for an augmented reality application. Some components, such as the server side, database and QR-code components, of a new GPS-based augmented reality application SARAS was developed and used for this purpose. This application works on the Android platforms. It shows the point of interest on the camera screen. It uses also motion sensors and the graphics API in addition to GPS and the camera and the web services for the content of bank data.

Firstly, power consumption analysis using the methods available in Android platforms were examined and improvements suggested by these methods are applied to the SARAS application. The functions and methods that consume much power were determined and improved. Basic resource closing controls are added to the application. The improvements reduced the resource consumptions up to %30. Afterwards, the detailed energy analysis was performed under different GPS and sensor processing settings. With the results of this detailed analysis the effects of the sensors and the GPS usage to the energy consumption has been clearly demonstrated. Also the impact of connection type and the different device factors on the energy consumption were observed.

Finally, the comparison with other applications were performed in terms of energy consumption. As a result, it has been observed that the SARAS application consumes energy close to the similar commercial applications like LAYAR.

5.1 Limitations and Future Work

In this thesis, the energy consumption analysis on different GPS and sensor processing methods were provided. The basic resource controls have been added for example battery and screen warnings if device has not enough battery and closing the resources on activity changes. Two GPS library options –Google Play Services and Android Location- and three different sensor process options were added to application for performance tests. For the future work, the performance analysis can be done by increasing the variety of these options. The effect camera, GPS usage and connection type effects can be considered in more detail. The screen positioning algorithm in the application can be developed to work more efficiently.

REFERENCES

Alem, L. and Huang, W. T. (2011). Recent trends of mobile collaborative augmented reality systems. *Springer*, 2011.

Anand, M., Nightingale, E. B., and Flinn, J. (2003). Self-tuning wireless network power management. *In Proceedings of ACM MobiCom '03*, San Diego, CA, USA, 2003.

ApacheFriends, (2011) ApacheFriends Home Page
URL: *https://www.apachefriends.org/index.html*

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. (2001). Recent advances in augmented reality. *Computer Graphics and Applications, IEEE, vol. 21, no. 6*, pp. 34– 47, 2001.

Ben Abdesslem, F., Phillips, A. and Henderson, T. (2009). Less is more: energy-efficient mobile sensing with senseless. *In Proceedings of ACM MobiHeld '09*, Barcelona, Spain, 2009.

Bimber, O., Raskar, R. and Inami, M. (2005). Spatial augmented reality. *AK Peters* Wellesley, 2005.

Chen, M., Chen, L., Zhang W. (2011). Analysis of Augmented Reality Application based on Cloud Computing. *Image and Signal Processing (CISP), 2011 4th International Congress on (Volume: 2)* pp. 569-572

Constandache, I., Gaonkar, S., Sayler, M., Choudhury, R. R. and Cox, L. (2009). Enloc: Energy-efficient localization for mobile phones. *In Proceedings of IEEE INFOCOM Mini Conference '09*, Rio de Janeiro, Brazil, 2009.

Erışık, D. (2016). *Screen Positioning Algorithm for Augmented Reality Application on Android Devices*. Master's thesis, Galatasaray University.

Janin, A. L., Mizell, D. W., and Caudell, T. P. (1993). Calibration of head-mounted displays for augmented reality applications. *Virtual Reality Annual International Symposium, 1993. 1993 IEEE. IEEE, 1993*, pp. 246–255.

LAYAR (2009). Layar Home Page.

URL: *https://www.layar.com/*

LAYAR Pricing (2009), LAYAR Pricing Detail Page.

URL: *http://www.layar.com/pricing/*

Mehdi, M., Lemieux, A. (2010). Augmented Reality: Applications, Challenges and Future Trends.

Orientation Vector, (2015) Orientation Vector

URL: *http://www.mathworks.com/hardware-support/android-sensor.html?s_tid=srchtitle*

PowerTutor (2011). PowerTutor Project.

URL: *http://ziyang.eecs.umich.edu/projects/powertutor/*

Raghav, S. (2012) Pro Android Augmented Reality. 2012.

Sarmiento, A. L., Margarita, A., Emilio J. P., Carlos V. R., Raquel C. and Pablo Q. (2012). Evaluating Performance of Android Systems as a Platform for Augmented Reality Applications. *International Journal on Advances in Software, vol 5 no 3 & 4* pp. 335-344

Shatte, A., Holdsworth, J. and Lee, I. (2014). Mobile augmented reality based context-aware library management system. *Expert Systems with Applications, vol. 41, no. 5,* pp. 2174–2185, 2014.

Shih, E., Bahl, P. and Sinclair, M. J. (2002). Wake on wireless: an event driven energy saving strategy for battery operated devices. *In Proceedings of ACM MobiCom '02*, Atlanta, Georgia, USA, 2002.

TURKCELL (2012). TURKCELL Pusula Application Page.

URL: *http://www.turkcell.com.tr/servisler/turkcell-pusula*

Van Krevelen, D.W.F. and Poelman, R. (2010). A Survey of Augmented Reality Technologies, Applications and Limitations. 2010

Viredaz, M. A., Brakmo, L. S. and Hamburgen, W. R. (2003). Energy management on handheld devices. *ACM Queue, 1(7):44–52*, 2003.

Wagner, D. and Schmalstieg, D. (2009). Making Augmented Reality Practical on Mobile Phones. 2009. *IEEE Computer Soceity* pp. 11-14

Wikipedia (2015). Global Positioning System

URL: *https://en.wikipedia.org/wiki/Global_Positioning_System*

WIKITUDE (2008). Wikitude Application Home Page.

URL: *http://www.wikitude.com/*

Zhanpeng, H., Pan, H., Christof, P. and Dimitris C. (2013). Mobile Augmented Reality Survey: A Bottom-up Approach. *ArXiv: 1309.4413v2* pp. A1-A35

Zhenyun, Z., Kyu-Han, K. and Jatinder, P. S. (2011). Improving Energy Efficiency of Location Sensing on Smartphones.

Zhou, F., Duh, H. B. L. and Billinghurst, M. (2008). Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. *IEEE Computer Society*, *2008*, pp. 193–202.

ZXING, (2015) Zxing Github Project Page.
URL: *https://github.com/zxing/zxing*

BIOGRAPHICAL SKETCH

The author of this thesis was born in 1986 in Izmir, Turkey. He has studied in Atakent Anatolian High School between 2000 and 2004, and started his undergraduate education in the Computer Science and Engineering Department of the Engineering Faculty of Ege University in 2004-2008 terms. Consequent to the graduation from the undergraduate degree, in 2011 he has enrolled to the Computer Engineering Master’s Degree in Galatasaray University Institute of Sciences. Since 2008, he has been working at banking sector in software departments.

Paper titled "Design Of Sensor-Based Augmented Realıty Software (SARAS)" of this thesis has been presented in 23rd IEEE Signal Processing and Communications Applications (SIU 2015).