

SINGIDUNUM UNIVERSITY
DEPARTMENT FOR POSTGRADUATE STUDIES

MASTER THESIS

**NEW APPROACH OF USING SMS FOR TESTING
QUALITY OF SERVICE IN WIRELESS NETWORKS**

Mentor:

Prof. dr Milan Milosavljević

Student:

Miloš Vukelić 410194/2015

Belgrade, 2016.

Table of Contents

1. Introduction	5
2. Theoretical Background	6
2.1. Mobile Networks	6
2.1.1. GSM	8
2.1.1.1. Channel Technology	9
2.1.1.2. SIM Card	9
2.1.2. GPRS	11
2.1.2.1. GPRS Benefits	11
2.1.2.2. GPRS Architecture	13
2.1.3. UMTS	14
2.1.3.1. Market Penetration and Evolution	14
2.1.3.2. Network Architecture	15
2.1.4. 4g	15
2.2. History of SMS	16
2.2.1. SMS Message Life Cycle in a GSM Network	17
2.2.2. Message Size	19
2.2.3. SMS Gateway	20
3. Quality of Service (QoS)	23
3.1. SMS QoS parameter evaluation	23
3.2. A reconfigurable QoS monitoring framework for professional SMS in GSM networks	26
3.3. A Quantitative Analysis of the Quality of Service of Short Message Service	27
3.4. Analysis of the Reliability of a Nationwide Short Message Service	27
4. Model for the new approach of using SMS for testing QoS in wireless networks	29
5. Architecture of the system	30
5.1. HTML5 & JavaScript	30
5.2. Java for Android	30
5.3. PHP	31
5.4. MySQL	31
6. Implementation	33
6.1. Registration (Android)	33
6.2. Login (Android)	39
6.3. Main Menu (Android)	40
6.3.1. Broadcast Receiver	41
6.3.2. Alarm Manager	42
6.4. Registration (Client Side)	50
6.5. Login (Client Side)	51
6.6. Testing menu (Client Side)	52
6.6.1. Test Results	54
6.6.2. Provider Dropdown list	57
6.6.3. Phone number status verification	58
7. Results	61
8. Conclusions	69
References	70

Figure 1 - Representation of cells in a cellular network (Miller, 2013).....	6
Figure 2 - Frequency Re-use (Frequency re-use pattern in cellular networks, n.d.)	7
Figure 3 - Simplified GSM architecture (Poole, n.d.)	10
Figure 4 - GPRS Network Architectur (Mousa, 2012).	13
Figure 5 - UMTS multi-radio access (Transition to 4G, 2010)	15
Figure 6 - Graphical representation of how SMS lifecycle procedure works (Hord, 2015)	17
Figure 7 - Diagrammatic representation of network elements in a GSM network (Gupta, 2015)	19
Figure 8 -GSM 7 bit alphabet (Harrington,2007)	20
Figure 9 - Direct-to-Mobile gateway (Developershome.com)	21
Figure 10 Direct-to-SMS gateway (Developershome.com)	22
Figure 11 - Quality of Service factors in a circuit switched network (Rahko, Herzberg and Rahko, 1972)	23
Figure 12- SMS Service Accessibility formula (Speech Processing, Transmission and Quality Aspects (STQ), 2008)	24
Figure 13 - SMS Access Delay Formula (Speech Processing, Transmission and Quality Aspects (STQ), 2008)	24
Figure 14 - SMS End-to-End Delivery Time formula (Speech Processing, Transmission and Quality Aspects (STQ), 2008)	25
Figure 15 - SMS Completion Failure Ratio formula (Speech Processing, Transmission and Quality Aspects (STQ), 2008)	25
Figure 16 - Qos Monitoring Framework (Waddt et al., 2005)	26
Figure 17 - Equation for the quantity analysis of QoS of SMS (Canlas et al., 2010)	27
Figure 18. New model - Using SMS for testing QoS in wireless networks.....	29
Figure 19. Three tier pattern [1]	30
Figure 20. Applied technologies.....	32
Figure 21. Registration screen	33
Figure 22. Registration Validation.....	34
Figure 23. Instantiation of AsyncTask	35
Figure 24. Task executed in background thread	36
Figure 25. HTTP Post request issued.....	36
Figure 26. Final step executed in UI thread	37
Figure 27. Validating result retrieved from the server	37
Figure 28. Parameters via HTTP request split into variables	38
Figure 29. MySql connection verified	38
Figure 30. Phone number is validated whether it exists in the database.....	38
Figure 31. Values inserted into the MobileUser and MobileUsersNew	39
Figure 32. Login screen	39
Figure 33. Current date and time obtained from the server	40
Figure 34. Specified SQL column is updated.....	40
Figure 35. Main screen of app	41
Figure 36. Broadcastreciever dynamically registered.....	42
Figure 37. PDU converted into SMS object.....	43
Figure 38. Parts from the constructed SMS object are accessed	43
Figure 39. Timestamp converted into date time format	44
Figure 40. Message body verified for the !reset! designator	44

Figure 41. Concatenated string passed into AsyncTask execute method	45
Figure 42. Concatenated string split into various variables.....	45
Figure 43. Further formatting	46
Figure 44. Current date and time from the server	46
Figure 45. Values inserted into the SMSTransaction table.....	46
Figure 46. Alarm Manager scheduled to fire an alarm at interval of 60 seconds.....	47
Figure 47. Current date is accessed via Calendar object and the extra phone number value is obtained	47
Figure 48. Content Provider query.....	48
Figure 49. First row is read.....	48
Figure 50. Message body check for !test! designator and placed as argument within execute() method.....	49
Figure 51. Cursor continues to loop until the last entry in the results set	49
Figure 52. Phone number value passed into execute method ().....	49
Figure 53. Current date and time.....	50
Figure 54. Lastconnecttoserver column updated	50
Figure 55. Registration of customer/client	50
Figure 56. Generation of unique test id.....	51
Figure 57. Login screen for customer/client	51
Figure 58. Main screen for testing results	52
Figure 59. Retrieving test id of corresponding user.....	53
Figure 60. Test id written in html section	53
Figure 61. Country drop-down menu process part 1.....	53
Figure 62. Country drop-down menu process part 2.....	54
Figure 63. Country drop down menu written to html section.....	54
Figure 64. Ajax call for requesting generation of test results.....	55
Figure 65. SQL query	55
Figure 66. Construction of test results table part 1.....	56
Figure 67. Construction of test results table part2.....	56
Figure 68. Test results table displayed on the client-side.....	57
Figure 69. Ajax call for requesting generation of provider drop down menu.	57
Figure 70. Provider drop-down menu displayed on client side.....	58
Figure 71. Ajax call for requesting generation of available phone numbers.....	58
Figure 72. Phone number list and status generated.....	59
Figure 73. Phone number indicated as available.....	59
Figure 74. Phone number indicated as unavailable.....	60

1. Introduction

Today's globalized and informatics societies mark a tremendous and subsequential increase in the demands of providing instant services at any point in place in any time. The emergence and sustained popularity of cellular or mobile networks has led many national and multi-national corporations to exploit the capabilities of mobile technologies present among such networks. Such a reason is because of a profound realization that there is a much more higher probability of achieving the necessary marketing goals. All of the specified goals are achieved through the assistance of various services or components available in a mobile communication system.

One of the most popular service component in a mobile phone is the Short Message Service (SMS). It is a service which enables the capability of exchanging short text messages between phone devices. Although over the past years SMS no longer remains a service strictly limited to sending messages between two phone devices but also between an application and mobile phone. This has given SMS a very important priority as it is applied in other various broad domains such as hotel booking, financial institutions, airlines etc. The very popularity and prominence of SMS services determines it being a valuable source of revenue within telecommunication industries.

By 2007 till 2014, the SMS messaging business has increased from the worth of 80 billion dollars [1] to 100 billion dollars [2]. These estimations indicate thoroughly the very competition taking place between multiple telecommunication companies which offer and provide wireless communication services. This has sparked a constant requirement of evaluating the quality of service (QoS) of their SMS traffic. Applying the various methodologies and techniques for the evaluation of QoS SMS traffic within a network, provides a major impact on the success of a telecommunication company and the prolongation of SMS as a reliable and sufficient service to meet end goals.

2. Theoretical Background

The functional abstract meaning of SMS diverges upon the very role or purpose it is utilized for. When taken from a very board perspective, it is either referred to as a service which enables the transmission and exchange of short messages between mobiles devices or a service which could offer customers personalized, sensitive or location based information [3]. The former designates a personalized use whilst the latter is more related towards the spectrum of marketing. Such an observation is very essential to take note of because the very fact that during its initial launch it was never conceived or indented to partake in such tasks.

A partial understanding of the history of SMS is very important as it runs analogous with an apprehensive description of the various technologies which underlie the very structure of the entire SMS life-cycle procedure.

2.1. Mobile Networks

Mobile phone networks represent the most common example of cellular networks. Cellular networks are utilized by wireless providers or mobile network operators for ensuring that services are offered to mobile subscribers at a higher coverage and capacity. Similar to other wireless networks or wireless technological standards, signals which carry text, voice and other kinds of data are transmitted via radio waves. The particularity of cellular networks is that the data transmitted goes through a global network of transmitters and receivers, and the underlying infrastructure is far more vast and immense when compared to the likes of other wireless technologies such as Wi-Fi and Bluetooth [4]. A very important aspect to highlight is the term "cellular" which designates the entire design or structure of the network. Generally cellular networks are conceptually represented as a number of hexagonal, circular or squared cells overlapping between each other and each cell including its own base station at the center. The diagram below represents a typical way how cellular networks are abstractly visualized.

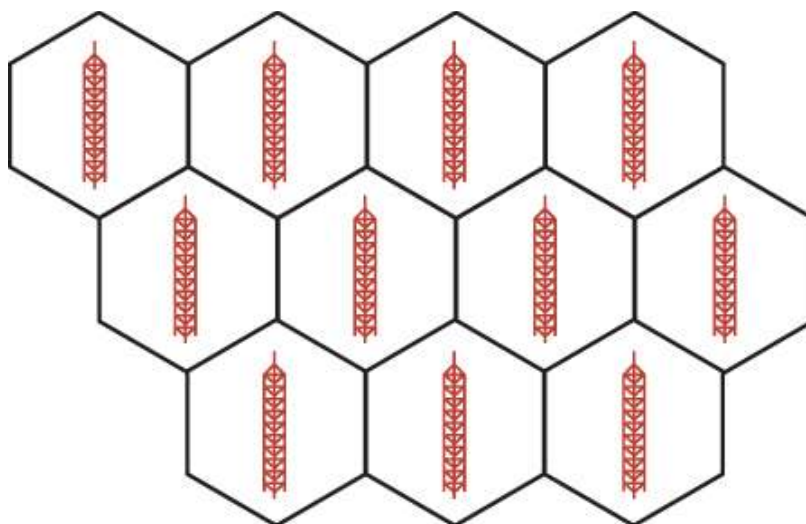


Figure 1 - Representation of cells in a cellular network [4]

Cells denote a specific geographic area where a cellular network is distributed, which is in fact the amount of area coverage provided by a single base station. Base stations serve as the central transceivers, relaying information or data from one device to the other. They also act as an interface allowing calls to be established between cell phones and wired phones. From a very concrete perspective, a base station is comprised of a number of antennas attached to a very high elevated structure known as a cell tower [5]. Other kinds of communications equipments are mounted on the tower such as the GPRS receivers [6]. The reason why the antennas must be mounted on very high structures such as cell towers is in order to overcome many of the obstacles present such as buildings, hills, trees etc.

Radio coverage over wide geographic areas is enabled by the various cells which overlap between each other. Whenever a caller would move beyond the coverage of a cell, a specific network element known as the Mobile Switching Center (MSC) informs the base station to transfer the control to another base station which is within the reach of the caller. Throughout this procedure it is taken into account that minimal interference is caused to the person's call and experiences the same frequency rate or in other words the same signal strength as before. The ability to re-use the same frequency rate within another cell in order to maintain a consistent signal strength with low interferences is the main key characteristic of cellular networks. This is achieved through a technique named frequency re-use. The technique is grounded on an evidential observation that the same frequency can be re-used in another cell as long as they are sufficiently far apart. Same frequency rates cannot be re-used on adjacent cells as it would cause co-channel interference. Co-channel interference is a cross-talk which is claimed to occur by two or more radio signals which are close to each other and of the same frequency. This often leads to undesired effects and major disruption. A common approach is that the frequency distribution goes according to a number of cell sets which cover a certain portion of a geographic area, these cells when grouped together form a cluster, then the same distribution is done to the other neighboring clutters. Certain cells in one clutter share the same frequency as a cell in the neighboring clutter. The figure below demonstrates such an approach.

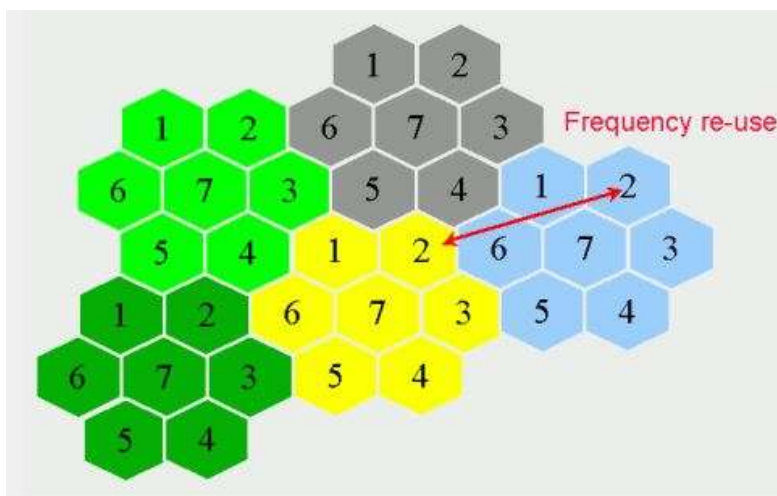


Figure 2 - Frequency Re-use [7]

Being that the entire infrastructure of cellular networks realistically differs from the theoretical conception it is difficult in finding the consistent theorem which would prevail over the many usual and reoccurring issues. Despite such an obstacle there has been plenty of innovation within the various technologies used in cellular networks which has lead to massive improvements and reliability.

Cellular networks in the very initial period of their deployment were regarded as the first generation of wireless mobile technology (1G). First generation cellular networks differed from the rest by being based on analog standards, the radio signals were analog rather than digital though an exception lies in the fact that digital signaling was used for connecting radio towers. Digital technology had a huge advantage when compared to analog because of the many capabilities it could offer which eventually led to the demise of 1G networks and replacement by 2g networks. Many of the services such as multimedia messaging and internet access would not have been actualized were it not for this decision to introduce digital radio signals in cellular networks. Third generation of wireless mobile technologies(3g) represent the culminating point so far that cellular technology has reached. In order to trace this evolution from 2g to 3g, it is necessary to provide an overview of three important cellular networks: GSM, GPRS and UMTS.

2.1.1. GSM

GSM refers to a set of standards and protocols which define the second generation of digital cellular networks developed by the European Telecommunication Standards Institute (ETSI) [8]. During the early 1980's the popularity of analog cellular networks was increasing as its use was expanding through most countries in Europe. However this only lasted for a short period as various telecommunication institutions were aware that limitations and issues present had to be overcome in a very delicate manner, which meant as soon as a possible alternative would be discovered this would include an entire replacement. Interoperability, security flaws and network congestion were the motivating factors in such a decision. In 1987, a memorandum of understanding was signed by several representatives from various European countries which involved the development and deployment of a common cellular system across Europe, and EU rules were passed to make GSM the mandatory standard [9]. Eventually this resulted in a unified, integrated, open standard network which according to some claims was more larger than the one in the United States [10].

2.1.1.1. Channel Technology

The chief technological backbone of GSM is a combination of time division multiple access (TDMA) and frequency division multiple access (FDMA). TDMA being the technology which succeeded FDMA. They belong to a set of channel methods which are known as multiple access schemes. As discussed earlier the importance of cellular networks is that multiple users would gain access to it and use it simultaneously but in accordance with a set of requirements which ensure that the spectrum efficiency is maximized and minimal interference is caused. For such requirements to be met, the significance lies in the approach

how each signal is allocated a frequency rate from the bandwidth spectrum. FDMA was applied within analog cellular networks, where each base stations send's the signals according to the frequency rate it was assigned with. Minimal interference and maximum efficiency is achieved by being able to reuse the same frequency rate as long as it is not within the same or adjacent neighboring cells. TDMA divides a radio frequency into multiple time slots which then each slot is allocated to a call. Though the difference with TDMA is that this radio frequency division occurs within one particular base station rather than being distributed into a wide array of base stations. Unlike FDMA, signal separations to not occur and this is the primary reason why TDMA must be used in conjunction with FDMA. When both are combined together this provides the use of multiple channels within the coverage of a single cell.

One major limitation was that if a handset would not respond within the allocated timeslot, the cell tower would discard it and switch to handling a call for another handset. Users would have to be given access to the shame channel by being differentiated through others mean except by the distributed timeslot. This limitation was overcome by differentiating each user according to a specific code they have been allocated with, the name of this multi access scheme came to be known as Code Division Multiple Access (CDMA). CDMA extracts the required data from a Direct Sequence Spread Spectrum(DSS) signal, which must consist of the correct spreading and chip code. Those DSS signals consisting of different chip codes would be rejected. Each of the codes allocated differ from each other, ensuring that the base station will be able to recognize from which handset was the signal transmitted.

2.1.1.2. SIM Card

Another aspect which differentiated GSM networks from its predecessors was the inclusion of an integrated memory circuit chips known as SIM cards within the mobile phones. SIM cards offer two key features: technological differentiation and data probability. Crucial information about the person's phone is kept inside the SIM cards memory, these are the telephone number, subscriber identity number and the carrier network the phone number is subscribed to. This enables the wireless network to identify whether the handsets posses the appropriate information to gain access to the network. Users would be able to easily swap handsets by inserting the SIM card previously used.

2.1.1.3. Network Architecture

GSM network architecture has remained mostly the same with only some additional changes. The changes are mostly related to the inclusion of several new interfaces which are demanded by specific network elements within other cellular systems. Most GSM network diagrams which depict the entire architecture in a very detailed manner tend to make it more incoherent and incomprehensible, for this reason a much a more simple diagram was given below.

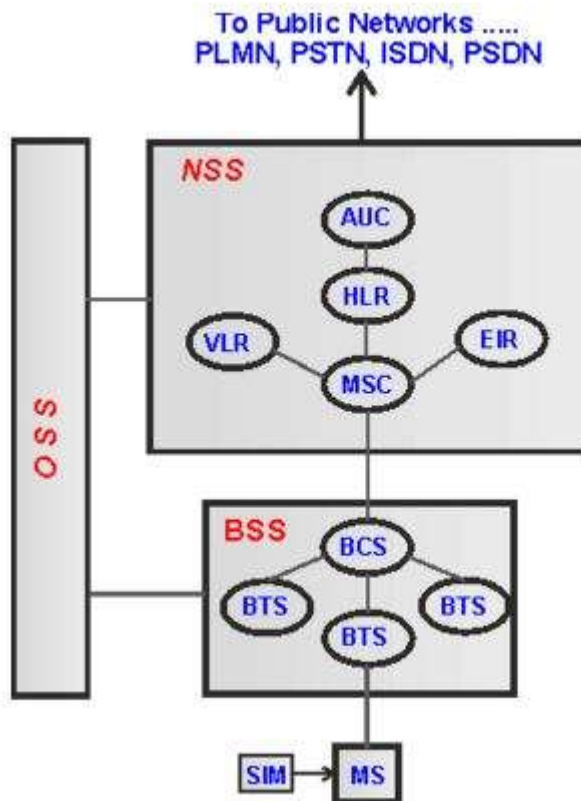


Figure 3 - Simplified GSM architecture [11]

Each interface forms a discrete section of the architecture which are:

Mobile Station (MS)

This section is related to the mobile equipment. Mobile equipments are comprised of two main important elements which are the hardware and SIM card.

Base Station Subsystem (BSS)

Base Station Subsystems are essentially associated with communicating with the devices that are on the network. This entire subsystem is formed out of two major components which are Base Transceiver Station (BTS) and Base Station Controller (BSC). BTS acts as interface between the mobile station and network. Radio signals are transmitted by the antennas, and the radio signals are received by the radio receivers. Each BTS is controlled by a BSC which handles radio resources, regulates handovers, allocates channels etc. All of its priorities are in one way or another related to ensuring the highest possible utilization of the radio sources.

Network Switching Subsystem (NSS)

NSS encompasses a number of elements each with its own specific functionalities in assisting and facilitating the GSM network for carrying out essential tasks such as call switching and mobility management. Often these entire elements when grouped together are regarded as constituting the "core network" of the entire GSM system.

Operation and Support Subsystem (OSS)

OSS represent computer systems utilized by telecommunication service providers for managing their cellular network. These management functions include network configuration, service provisioning, fault management etc.

2.1.2. GPRS

General packet radio service (GRPS) is a data service and protocol which is packet oriented. Packet switching involves the procedure of dividing the data which is to be transmitted into separate small blocks. Once the packets have been received to the corresponding address, they are reassembled in the proper order. GPRS is standardized by the ETSI and maintained by the 3rd Generation Partnership Project (3GPP) [12]. One of the very limitations present within GSM is that its core architecture is grounded upon the circuit switching methodology. A circuit switched network is ideal for the delivery of voice calls but limited with sending data. The data limitations imposed by the GSM required an extension which would overcome this issue, and that is the reason behind the invention and introduction of GPRS.

GSM cellular networks which include GPRS are categorized and described as 2.5G, this represents cellular technology between the second and third generation of mobile telephony. Such a network marks an evolution of GSM networks, which is why it is often claimed that GPRS was necessary migratory path towards 3G networks. Allowing most network operators to include an IP based core architecture within their cellular network, this resulted in being used and further expanded for 3G services [13].

2.1.2.1. GPRS Benefits

The numerous technological benefits brought by the GPRS has led to the development of various unique services provided to mobile subscribers. Each service is based on 3 specific characteristics which are mobility, immediacy and localization. Combining each characteristic has spawned a number of unpredicted and inconceivable applications which would emulate certain applications and services on the desktop computer. Most of the applications range between the consumer and corporate category. Typical examples would be e-mail, information services, banking, financial trading, traffic conditions, advertising etc. GPRS provides much more higher data rates than what would have been achieved within GSM systems. It is estimated that maximum data rates which are realistically achievable lie between the range of 40 - 114kpssecond. However some claims state that data rates up to 171kbps are possible [14], whether such claims are true or not in both ways surpasses the maximum data rate of a GSM system which is around 9.6 kbps [15]. Like any other technology whose data transmission mechanism is based on the IP core architecture, they offer the "always on" ability. Which implies when a connection is established, the user will

not experience any cuts being that charges are based only on the amount of data that is carried rather than on the time a circuit is used.

Two very important services existed during the period of GPRS which are Multimedia Messaging Service (MMS) and Wireless Application Protocol (WAP). MMS represents a service which extends the core concept of SMS, surpassing certain limitations such as the maximum length of 160 characters for each SMS message. Originally, MMS was developed on the intention of being a captivating technology which would enable service providers to collect fee any time someone would take a photograph [16]. Other media companies make use of MMS for delivering content such as news, entertainment on a commercial basis. Being that MMS had its own disadvantages such as no cross compatibility support, the features would eventually begin to be included in SMS. According to the surveys carried out during the several past years, SMS messages consisting of entertainment content account most of the traffic between 2010 and 2013 which has increased to 70% from 57 billion to 96 billion messages sent [17]. The ability for SMS to support such content led to its increasing prominence and persistence rather than being replaced by another service. WAP is a technical standard for information being accessed over a mobile wireless network. During the GPRS period it was considered the second most important technological innovation next to MMS. Before its release publicly, plenty of hype surrounded WAP that it would eventually lead users to believe that the performance would be equivalent to that of a fixed non-mobile Internet access [18]. However the reality fell far short of expectation when the first handsets which support WAP were released. The ability of delivering web content to the mobile which would be tailored to its limitations so far posed plenty of issues which are mostly delays and slow links. Most users who possessed a handset supporting the WAP protocol found the entire experience vivid enough. WAP was seen as another attempt to create a technology which would replace SMS by having the functionalities extended instead of being restricted to the reliable and robust 2 way messaging. Eventually including WAP in the subsequent mobile phones generation was quickly abandoned. Another important reason behind this, was that most apps running on mobile phones were not designed to access information over the internet. In fact this entire period can be viewed and understood as a tide whether mobile operators could charge their users on some other services which would emulate and replace SMS, but the very failure in the markets is another indication that SMS by itself is a service which gradually grew to become very important.

2.1.2.2. GPRS Architecture

As mentioned earlier that the GRPS is only partial data network which overlays or in other words extends the existing GSM network. Most of the elements from a GSM network still remain in a GPRS network with some slight modifications. GPRS network attempts to reuse most of the GSM elements but certain new elements, interfaces and protocols were included in order for an effective packet-based transmission.

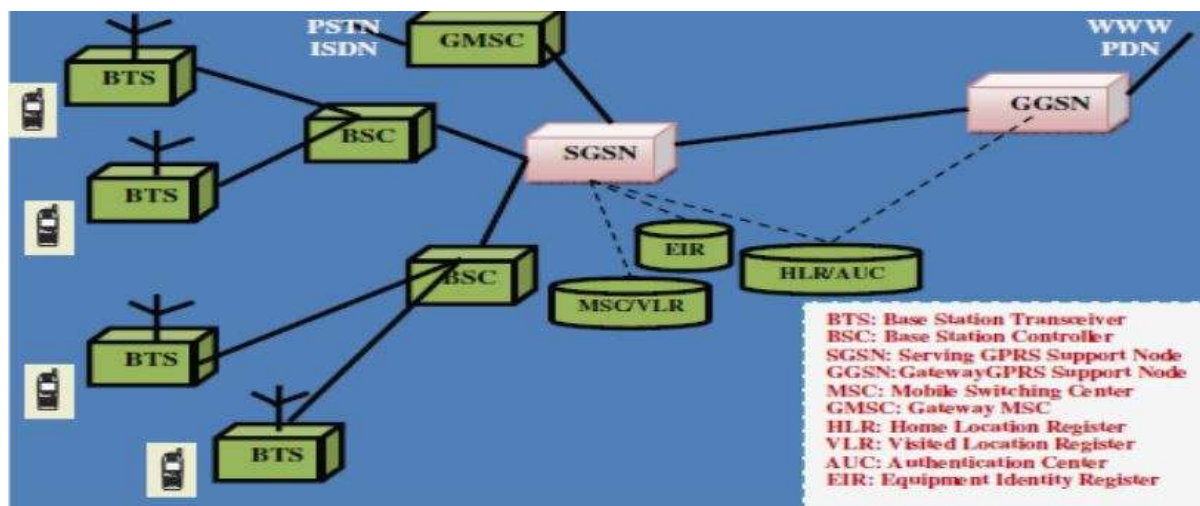


Figure 4 - GPRS Network [19].

GPRS Support Nodes

Two most fundamental and compulsory elements required for the installation and deployment of a GPRS network are the Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). Both of these elements form part of the GPRS support Nodes (GSNs) which handle the traffic and transmission. SGSN ensures that the data packets are delivered to proper mobile station or handset. Other tasks performed by the SGSN are mobility management, authentication, charging subscribers and preserving a profile of GPRS subscribers. GGSN acts as interface to any external IP network such as the public internet. Whenever the GGSN receives incoming data packets via the SGSN it forwards it to the relevant network. The reverse occurs when data packets need to be delivered to handset, where the GGSN transfers the entire data traffic to the corresponding SGSN.

GPRS Base Station Subsystem

Each BSS requires the installation and upgrade of a component known as the Packet Control Unit (PCU). By having this hardware implemented, the BSS will be provided with the physical and logical data interface for data packets to be transmitted. Data and voice traffic is as always transferred first to the BTS and then to the BSC, at this point each traffic will be sent to a separate entity; data packets are sent to SGSN via the PCU and voice traffic is sent to the NSS.

2.1.3. UMTS

The Universal Mobile Telecommunications System (UMTS) is the first cellular system successor to the GSM. Proposals to make UMTS the successor were stated explicitly at the International Mobile Telecommunications-2000 (IMT-2000) [20]. Besides this, UMTS is also based on a set of standards which comply with the specifications made by the International

Telecommunications Union which indicates UMTS is also the first third generation (3g) cellular network [21]. 3g technology provides a number of new advantages to network operators enabling them to offer more broadband services whilst maintaining a greater network capacity through improved spectral efficiency. Services included are voice calls, broadband wireless data, wide-area wireless voice telephony and low cost delivery all in a mobile environment with data rates up to 2mb/s.

Despite many of its initial setbacks UMTS preserves many of the global roaming capabilities of second generation GSM networks including those incorporated with GPRS and extends the core network with further components eventually becoming the established 3g technology.

2.1.3.1. Market Penetration and Evolution

One of the most interesting aspects of UMTS was the unpredicted market success and penetration it had attained. Japan and South Korea were the two first countries where UMTS branded as a 3g service had penetrated over 70% of the market. In Europe 3g became the dominant cellular telecommunication standard for marketing purposes and still remains to be for all these years since its deployment. The widespread adoption of 3g has been quickly followed into other countries whose technological progress was not as dynamic as those of the former but nevertheless the percentage increased voluminously.

This entire success was very critical to fulfilling one of the most important ideas behind 3g which was the unification of standards used in 2G wireless networks. Despite GSM being a global standard which would enable interoperability between networks it was never built with the intention of being the one single acceptable standard and for this reason different network types adopted in Europe, Japan and Americas differed. Through this standardization process a global family of 3g mobile communication systems would emerge having a long standing usage.

With the very spread of mobile phones containing a specialized operating system, marketers have finally found and seized the proper opportunity in being able to finally offer a vast number of different services to their subscribers. The reason is because with the very advent of mobile operating systems such as Android and iOS the average time most users keep their network connection on has increased tremendously. Many entrepreneurs foresaw this as the very beginning of the establishment and actualization of an "Internet Of Things" [22]. The very phrase "Internet Of Things" refers to a number of devices or physical objects such as vehicles or buildings embedded with specific type of electronics and software which would guarantee connection to a network and enable the collection and exchangeability of data [23].

2.1.3.2. Network Architecture

UMTS core network architecture is based on the GSM and GPRS network topology. The core network encompasses the necessary circuit switched elements and pack switched elements. New elements included in the UMTS are few and only extend and enhance the functionalities related to the preservation of data, authentication of mobile equipment and strengthening data security. There is only in two aspect that UMTS differs from GSM and GPRS which is the radio transmission standard and support for multiple radio access networks. Radio transmission standard used in an UTMS network is known as Wideband CDMA (WCDMA), as a result of some misconception the terms UTMS and WCMA have become used interchangeably to designate a 3g network. WCMA is often characterized by utilizing a wider band than that of its counterpart CDMA. Through such an approach WCMA is able to promise additional advantages of transfer rates, increased system capacity and communication quality by statistical multiplexing. According to the specifications, the UMTS core network supports multiple radio access methods which implies that mobile handsets designed specifically for a GSM network would be able to establish a connection with the core network. This offers much more flexibility to mobile operators when it comes to increasing their coverage area and a less expensive migration.

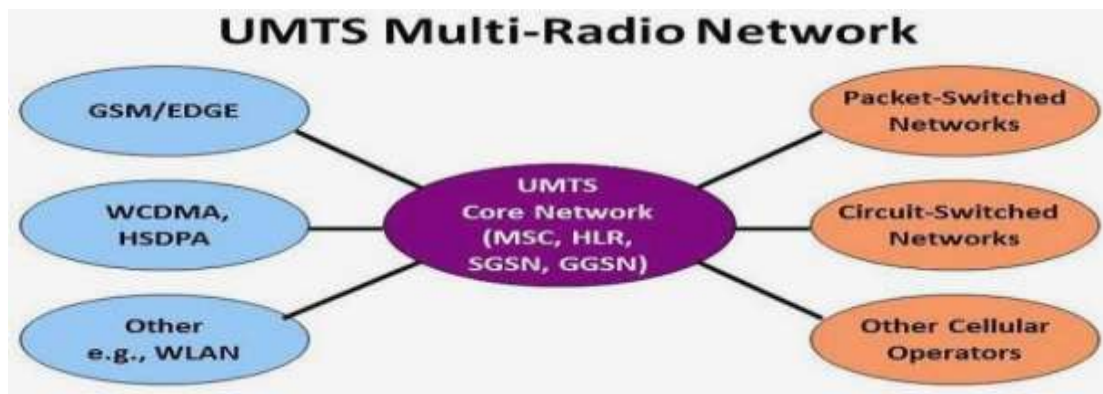


Figure 5 - UMTS multi-radio access [24]

2.1.4. 4g

Till now networks which fit in the category of 3g still remain to be the most deployable globally. However despite the amount of acclaim achieved, adaptation to newer emerging technologies, offering services at a higher capacity and new levels of user experience are constantly on the demand. As it can be known beforehand that third generation of mobile communications technology will be succeeded by the fourth generation (4g) the importance lies in knowing the necessary requirements for the actualization of a network which would fit in the 4g category. There are three main properties which outline the very objectives that any cellular telecommunication system must accomplish, these are the following:

Ubiquity

Ubiquity of a network refers to its availability for access in anytime and anywhere. It remains as the most demanding requirement but the most difficult to be achieved. In order for this objective to be accomplished it requires most of the services and technologies to be standardized at world wide scale.

Multi-service platform

4g networks are expected to provide a number of diverse services. The approximate number of services to be provided is still unknown but according to the specifications defined by ITU, these services are intended to replicate web experience of the computer on the phone through capabilities as video streaming and cloud computing. With the amount of new services included, mobile operators will be able to discover new levels of traffic.

Low bit cost

A technical aspect which will differentiate 4g cellular systems from its predecessors is its IP centricism. Mobile operators have decided that the traditional mechanism of circuit switching will be omitted. Apparently there are two reasons for this, one that voice will lose its value in the overall user bill and there is a higher demand for further implementation of data services. Further measurements must be applied which would ensure that charges are based on per bit transferred.

2.2. History of SMS

Before the concept of SMS could have been envisioned and developed, one of the necessary advancements was the inclusion of text-messaging in mobile devices. The development was approved by the first action plan of the CEPT Group GSM around the 1980's, which requested services and facilities present within public data and switched telephone networks to be available in a mobile system [25]. The purpose behind this was based on the inference that such services and facilities would provide the mechanism of exchanging text messages.

After a couple of years have passed since the initial development of text-messaging within mobile systems, the SMS concept was finally envisioned and developed by Friedhelm Hillebrand and Bernard Ghillebaert [25]. Their key idea was to take advantage of the optimized telephone system where SMS would be used. A differing aspect of SMS compared to the usual text messages was it being limited to a specified length. Such a decision was necessary in order for the messages to fit into the specific signaling formats which are transported via signaling paths involved in controlling the telephone traffic during periods the traffic did not take place. Through such an approach, unused resources within the system can be utilized for transmitting messages at a minimal cost.

The further development of SMS constituted the creation of various governing standards and protocols from numerous multinational collaborations which would pave the path for its

introduction in the entire world. Additional necessary requirements where a specific network element known as SMSC, and improvements among the radio capacity and network transmission architecture for adjustments of growing SMS traffic. Plausibly the last important milestone throughout the entire development phase, was an elaboration that the essence of SMS was not solely based on provisioning the transmission of messages between mobile users but also providing acknowledgement capabilities. According to some researchers this very clarification and elaboration had turned SMS into something much more efficacious than an ordinary paging mechanism [26]. By 1992 the introduction of SMS to the world began.

2.2.1. SMS Message Life Cycle in a GSM Network

Whenever an SMS text message is sent from a users mobile phone, it is delivered and retrieved by the Short Message Service Center(SMSC). As only conveyed in a very short manner previously in the history of SMS, the SMSC is a very prominent network element which handles the entire store and forward procedure of SMS messages within a mobile phone network. Once an SMSC has received the SMS it responds by sending the message to the corresponding phone number. If in certain cases the destination where the message is to be forwarded is not accessible at that particular moment, the message is placed by the SMSC in a que for later delivery.

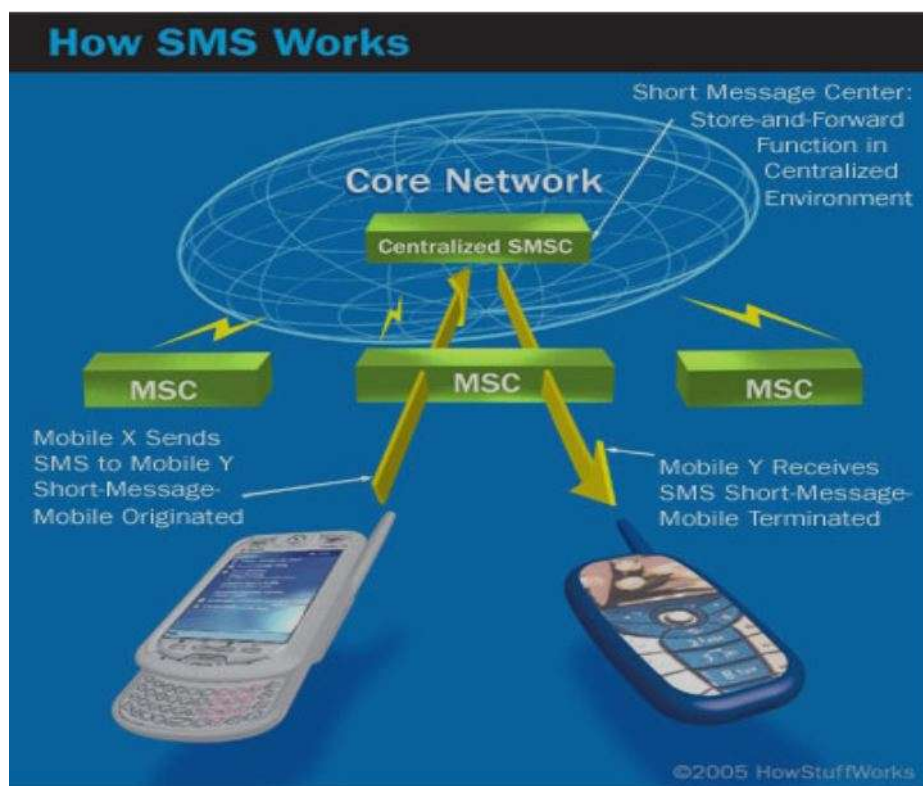


Figure 6 - Graphical representation of how SMS lifecycle procedure works [27]

In order for the SMSC to partake in its role and fulfill its purpose, it relies on the assistance of three other components which are Mobile Switching Center (MSC), Home Location Register(HLR) and Visitor Location Register(VLR).

MSC

A MSC is the central and core entity of a network switching subsystem (NSS). In other words it is the entity which carries out the switching of connections between mobile stations and networks. Switching involves the procedure that the proper destination route has been established where the SMS is to be delivered. A proper route signifies that the MSC has switched the route to a corresponding Base Station which would transfer the message to the mobile phone. The MSC works in conjunction with the HLR and VLR.

HLR

HLR is the central database in a mobile network. It stores and preserves entries for each mobile subscriber profile belonging to a specific Mobile Network Operator(MNO). The entries kept is bound only to subscribers who are authorized to use the GSM core network. Some of the important information maintained within an subscriber's entry are the international mobile subscriber identity (IMSI), mobile station international subscriber directory number (MSISDN) and current location of the subscriber. These 3 fields comprise a sufficient amount of details for the SMS message to be passed on to the proper MSC.

VLR

VLR is the second database in a mobile network. The information stored within it is similar to that of the HLR except that it only keeps entries of mobile subscribers which belong to a geographical location it is responsible for. In technical terms, each VLR is associated with at least one MSC that are allocated within a specific location area, and are to store information of subscribers which roam within a MSC it is associated with. Data kept in the VLR is changed dynamically as the subscriber moves into a MSC location area. With the dynamical modification of a subscriber's entry according to his or her position within a MSC location area, the MSC would be able to establish the route with the corresponding base station.

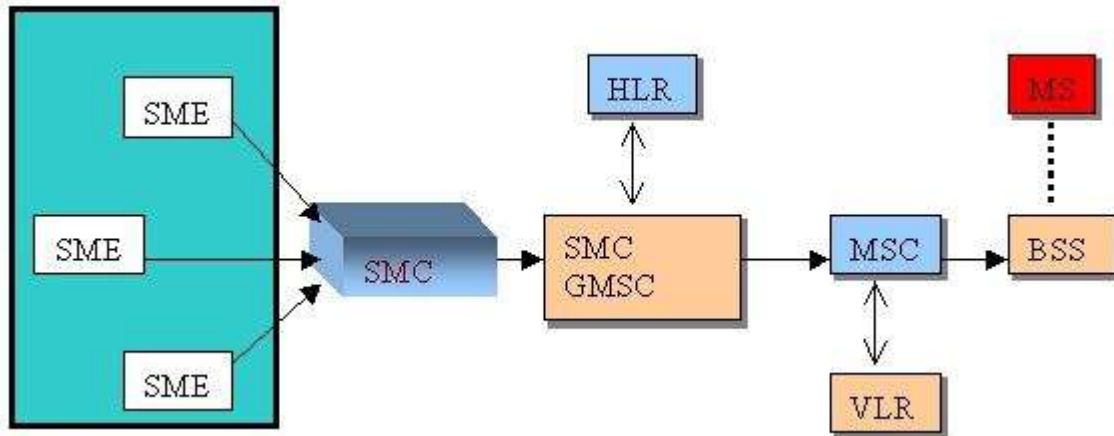


Figure 7 - Diagrammatic representation of network elements in a GSM network [28]

2.2.2. Message Size

Transmission of SMS messages between a mobile and an SMSC occurs through the employment of an SS7 protocol known as Mobile Application Part (MAP). Such protocols are specifically used in telecommunication networks. MAP provides an application layer or interface to various elements or nodes present within GSM, UMTS and GPRS networks in order to facilitate delivery of services and intercommunication. Messages are sent by MAP Mobile Originated (MO) and mobile terminated (MT) operations whose length is generally restricted to 140 8-bit characters because of the constraints present within SS7 signaling protocols [29]. Once the period of SMS popularity began to grow it was necessary to provide encoding support for other varieties of alphabets. Length of SMS messages now vary on the alphabet encoding option selected, for the very reason that the alphabet denotes the number of possible characters [30]. The maximum length that an SMS message can be composed of is 160 characters. The figure below represents the default GSM 7 bit alphabet.

	0	1	2	3	4	5	6	7
0	@	Δ	SP	0	;	P	ı	p
1	£	_	!	1	A	Q	a	q
2	\$	Φ	"	2	B	R	b	r
3	¥	Γ	#	3	C	S	c	s
4	è	Λ	α	4	D	T	d	t
5	é	Ω	%	5	E	U	e	u
6	ù	Π	&	6	F	V	f	v
7	ı	Ψ	*	7	G	W	g	w
8	ò	Σ	(	8	H	X	h	x
9	Ç	Θ	)	9	I	Y	i	y
10	LF	Ξ	*	:	J	Z	j	z
11	Ø	Note 1	+	;	K	Ä	k	ä
12	ø	Æ	,	<	L	Ö	l	ö
13	CR	æ	-	=	M	Ñ	m	ñ
14	Å	ß	.	>	N	Ü	n	ü
15	å	É	/	?	O	Š	o	š
Note 1: This code is an escape character used to access the GSM 7-bit extension alphabet.								

Figure 8 -GSM 7 bit alphabet (Harrington.2007) [1]

The 160 character limitation imposed upon SMS messages can be overcome by the use of concatenated SMS messages. Conceptually, a message which surpasses the maximum limit will be split into smaller pieces and assembled together once delivered to the appropriate handset. Concatenated SMS messages are technically known as PDU mode SMS. The adoption of such a feature began in the late 2000s to culminating in the early 2010s.

2.2.3. SMS Gateway

Each wireless carrier or mobile network operator makes use of an SMSC from a specific SMSC vendor. Transmission of SMS messages between two mobile subscribers belonging to different wireless operator networks takes place between two SMS centers. In order for the SMS message to be delivered it requires both SMS centers to be based on the similar communication technologies. For such an issue to be overcome, an SMS gateway was invented. An SMS gateway stands as a relay entity between the two SMSCs translating one protocol to other. In such a way an interconnection can be established between two different wireless networks.

The advent and prominence of A2P(Application 2 Person) SMS messaging as being the main driver behind SMS revenue and a growing traffic which gradually exceeds that of P2P (Person 2 Person) would not have occurred without the very technical and marketing benefits brought by the SMS gateway. A very popular SMS marketing campaign targeted by vendors is known as Bulk messaging. Bulk messaging is the dissemination of a vast amount of SMS messages to be delivered to mobile phones spread around the globe. Besides being used for marketing agendas, it is also utilized in mission-critical messaging areas such as determining whether a credit card transaction was fraudulent or not. For messages to be routed into the specified handsets, a connection must be established between the various wireless or cellular mobile networks and the SMS gateway. Business entities offering Bulk messaging services negotiate agreements with mobile carriers for connections to be established. Such Business entities are often classified as SMS aggregators or SMS gateway providers.

SMS gateways tend to differentiate according to the way and specific paths being followed as to how SMS messages are transmitted. The two most common SMS gateways are Direct-To-Mobile and Direct-to-SMS gateway.

Direct-to-Mobile gateway

This kind of gateway differs from an SMS aggregator for being in an organizations own network which connects to a local mobile network. SMS messages can be sent or received by email, Web site/Web app and by other desktop apps. Connections are established by acquiring a SIM card belonging to the corresponding mobile network and installing it inside the gateway. Being that these gateways have built in wireless GSM connectivity, connections could either be directed at GSM/GPRS modems or mobile phones.

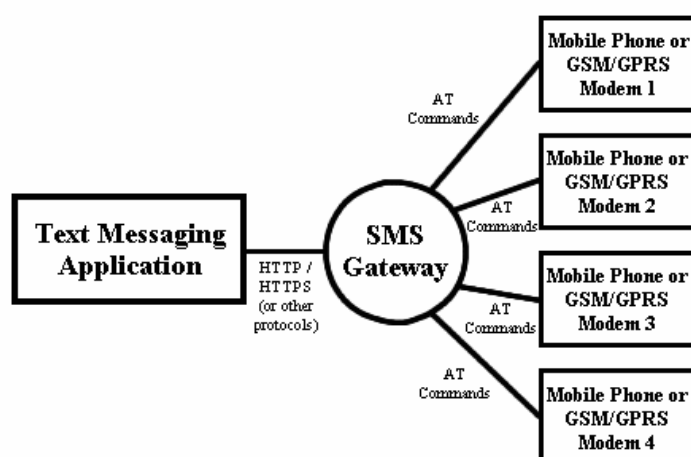


Figure 9 - Direct-to-Mobile gateway [31]

Direct-to-SMS gateway

This type of SMS gateway is mostly used by SMS aggregators for carrying out Bulk services. On account of that such gateways can handle a huge amount of messages, and feasible connections maintained with other mobile networks through direct contracts.

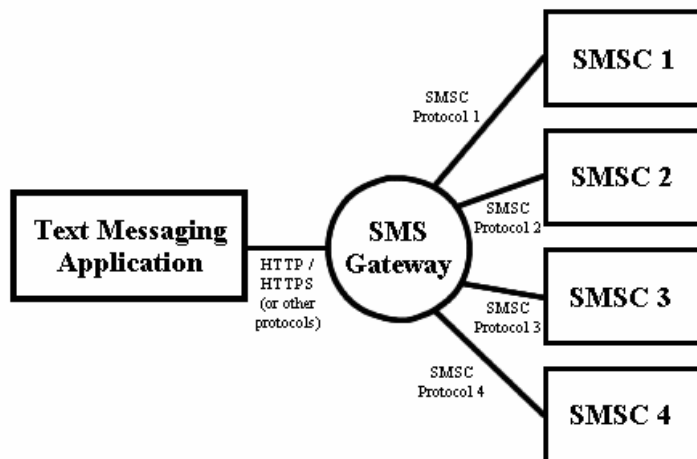


Figure 10 Direct-to-SMS gateway [31]

3. Quality of Service (QoS)

Quality of Service (QoS) in its simple terms refers to the overall performance of a network as seen by the user. Within the field of telephony, quality of service was contextually defined by ITU in 1994 [32]. The outstanding prominence given to QoS in telephony was only with the emergence of IP centric cellular networks. Practically QoS was not important as now because most mobile operators would charge a subscriber based on the duration of his or her connectivity. IP revolution has resulted in a tremendous popularity of Voice over IP (VoIP) technologies where subscribers are able to make use of voice and multimedia services over IP networks and be charged only for the bits of data transferred rather than the connectivity duration, which amounts to low costs. Mobile operators have come up with the alternative on charging their subscribers based on the content and utilities the service provides. In order for this charging to take place it is necessary to understand the requirements desired from the users point of view. Putting it firmly QoS stands for a set of specific requirements provided by a network to users, which are necessary to achieve the required functionality of an application [33]. According to ITU properly conducted QoS evaluation settles a degree of user satisfaction [34].

3.1. SMS QoS parameter evaluation

Each service is expressed through a number of parameters. Parameters represent various factors which determine the quality of the service from the perspective of the sender and receiver. For most services involving packet transmission there are set of explicit universal parameters that are directly or indirectly relevant to the service itself. SMS remains still to be one of the widely used services where the transmission occurs over circuit switched network rather than a IP based network. There appears to be only few number of abstractly defined parameters which comprise the entire QoS for a circuit switched network which can be seen in the figure below [35]. Two main categories are exhibited where each parameter is classified accordingly.

Human factors	Technical factors
stability of service quality	reliability
availability of subscriber lines	expandability
waiting times	effectiveness
fault clearance times	maintainability of the system
subscriber information	congestion waiting
stability of operation of the system	transmission quality
...	...

Figure 11 - Quality of Service factors in a circuit switched network [35]

One of the problems faced when these parameters are used for carrying out an QoS evaluation of the SMS service is that they do not express or convey factors which are intrinsic to SMS itself. Without any parameters which correlate with any specific property of an SMS service, the evaluations conducted will appear meaningless. The meaningless eventually leads to no insights into further decision making for the improvement of the SMS service. In order for such an issue to be tackled the European Telecommunication Standards Institute (ETSI) has defined a set of parameters for the evaluating the quality provided by an SMS service [36]:

SMS Service Non-Accessibility Mobile Originated (MO) [%]

Probability indicating the failure of the subscriber to use the service even though it is accessible. This value is obtained by first using the formula below to calculate SMS service accessibility rate.

$$\text{SMS Service Accessibility [\%]} = \frac{\text{No. of Successfully Sent SMS}}{\text{Total No. of SMS Sending Attempts}} \times 100\%$$

Figure 12- SMS Service Accessibility formula [36]

After the first step is done, the SMS Service Non-Accessibility formula is used.

$$\text{SMS Service Non-Accessibility[\%]} = 100 - \text{Service Accessibility[\%]}$$

SMS Access Delay MO [seconds]

The time elapsing between submitting a message to the SMSC and receiving an acknowledgement reception from the SMSC. SMS Access Delay is sometimes referred to as SMS sending time.

$$\text{SMS Sending Time [s]} = t_{\text{notification}} - t_{\text{sending}}$$

t_{sending} – moment when the user starts sending the SMS.

$t_{\text{notification}}$ – moment when the mobile equipment originating the SMS receives the confirmation that the SMS was delivered to the SMSC.

Figure 13 - SMS Access Delay Formula [36]

SMS End-to-End Delivery Time [seconds]

The time interval between submission of a message and receiving a message at another mobile equipment. SMS End-to-End Delivery is sometimes known as SMS Delivery Rate. Messages not delivered within the specific timeframe are identified as failed messages and excluded from the percent rate.

$$\text{SMS Delivery Rate [\%]} = \frac{\text{No. of Successfully Received SMS}}{\text{Total No. of SMS Sending Attempts}} \times 100\%$$

Figure 14 - SMS End-to-End Delivery Time formula [36]

SMS Completion Failure Ratio [%]

A ratio between the number of messages that were not delivered and total number of SMS messages sent. First step in obtaining the ratio, the SMS delivery rate must be calculated.

$$\text{SMS Delivery Rate [\%]} = \frac{\text{No. of Successfully Received SMS}}{\text{Total No. of SMS Sending Attempts}} \times 100\%$$

Figure 15 - SMS Completion Failure Ratio formula [36]

After the first step, the following formula is used:

$$\text{SMS Completion Failure Ratio [\%]} = 100 - \text{SMS Delivery}[\%]$$

3.2. A reconfigurable QoS monitoring framework for professional short message services in GSM networks

Waddt [37] is proposed a framework for QoS monitoring, alerting and reconfiguring the SMSC. Through this framework they attempt to elucidate the issue of SMS packet loss caused by forced packet removal at SMSC due to timeouts (i.e ., timeout of the message has expired before the message has been delivered). Their proposed system architecture is presented below.

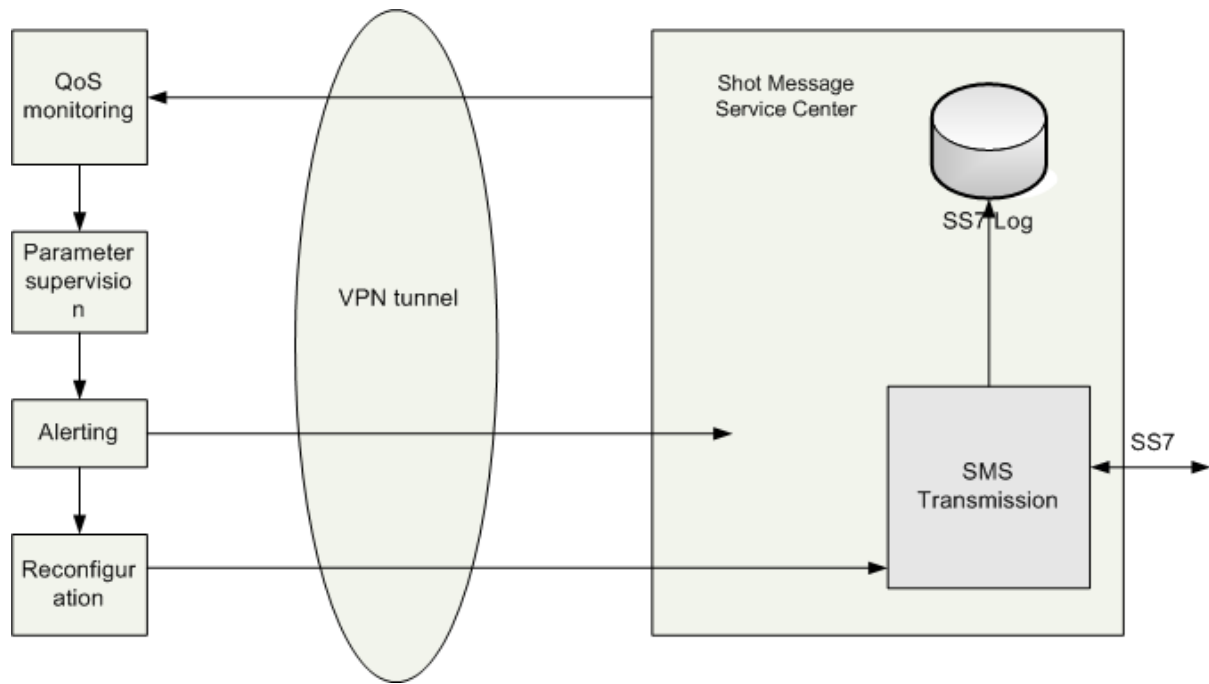


Figure 16 - QoS Monitoring Framework [37]

For the evaluation of the signaling domain, besides the 4 QoS parameters (Service Accessibility SMS MO, Access Delay SMS MO, End-to-end Delivery Time SMS, and SMS Completion Failure Ratio), three additional QoS parameters were introduced:

Success Ratio SMS (SR SMS)

Ratio of the amount of successfully executed Mobile Application Part (MAP) commands and the total number of MAP commands.

Error # Ratios SMS (E#R SMS)

Ratio of the number of MAP commands returning a particular error code of the SS7 protocol and the total number of MAP commands.

Effort SMS (E SMS)

Average number of protocol commands made when attempting to transmit a message.

The QoS parameters values are measured and compared to the values stored in a Parameter Supervision block. If the observed values are out of range, then an alert is sent to the SMSC.

3.3. A Quantitative Analysis of the Quality of Service of Short Message Service

A quantitative analysis of the QoS of SMS requires the formulation of an enhanced QoS equation being based on a number of properly selected parameters. Canlas [38] suggested packet-loss, inaccurate prepaid load and end-to-end delay are measured in order for an QoS

equation to be formulated. This QoS equation would be able to describe billing performance, network behavior and overall quality service of an SMS system. Three mobile operators were chosen to create nine sending combinations. The following parameters were most strictly observed: Time Delay per message, Number of Messages sent, Number of Messages received and Load Consumption. Grounded upon these chosen parameters the final equation was developed:

$$QoS = \left(\frac{M_R}{M_{S,ideal}}\right)^3 * \left(\frac{M_{S,actual}}{M_{S,ideal}}\right) * \left(\frac{TI}{T_D}\right) * \left(\frac{1.00 \frac{Php^2}{messages^2}}{L_C}\right)$$

where:

QoS = overall QoS [unitless]

M_R = number of received message [messages]

$M_{S,ideal}$ = number of messages attempted to be sent [messages]

$M_{S,actual}$ = actual number of messages sent [messages]

TI = Timeliness Index [seconds]

T_D = average time delay [seconds]

L_C = load consumption

Figure 17 - Equation for the quantity analysis of QoS of SMS [38]

3.4. Analysis of the Reliability of a Nationwide Short Message Service

Whilst most analysis carried out through the use of formulas and enhanced equations to detect certain flaws in a service and how such flows could be adjusted. Certain misconceptions emerged after the tests regarding the overall quality of an SMS service. This is because some parameters such as SMS access delay and SMS end-to-end delivery which are classified as Key Point Indicators (KPI) only indicate the quality and performance based on certain conditions. A very interesting researched conducted by Meng [39] focuses on the intrinsic robustness and reliability of the SMS service. Without performing any test themselves, they trace their study of SMS reliability through data captured by a national cellular operator which contains information related to 59 million messages. Throughout this research the two primary parameters considered to be KPIs for service reliability were message completion failure ratio and end-to-end latency. According to their research the delivery failure ratio is as high as 5.1%, whilst the latency experienced by at least 91% of subscribers is less than 5 minutes.

Overall these results demonstrate the very efficiency inherent in the SMS service without the need of adding further improvements. SMS packets which have been failed to be delivered is obviously because the SMSC has not received any response from the MSC that a destination route has been established. And after a certain period the SMSC is bound to handle the transmission of another SMS packet. The experience of latency does not reflect whether the SMS service is not capable of fulfilling its role, but rather is a way for mobile operators to

increase their demand on routes offered to bulk SMS services. The authors indicated that bulk SMS services have overall effect on the increasing latency and failure of delivery. Traffic ratio within bulk-sms services tend to exceed to that of person-to-person traffic by around 53.7 %. Person-to-person traffic ratio remains to be below 3.7%.

The QoS evaluation of SMS has always been addressed from a topical point of view. Throughout this research a variety of formulas and equations based on an underlying framework induced by the ETSI standard have determined very accurate results regarding the overall reliability and performance of an SMS. Despite this one of the issues with these evaluations is that they are not addressed from the perspective of a business entity such as the SMS aggregators. SMS aggregators need to constantly verify the traffic of the routes they have established with various mobile carriers, the quality of each route is determined by whether a big portion of SMS messages is delivered or not. One of the main problems such business entities face is there being no way to carry out a proper verification whether the end results are legitimate. This issue moves further to other business entities as mobile marketers who excessively rely on using the services provided by an SMS aggregator which have escalated into confrontations where SMS aggregators have been accused of providing fake delivery reports.

As a result of these issues, some telecommunication companies have begun to offer integrated testing environments. These testing environments allow SMS aggregators and mobile operators to perform QoS tests through an efficient means by being able to examine the end results. Nevertheless there are still problems posed by such environments being that there are two primary characteristics they lack, which are coverage and real time tests. Coverage is a characteristic indicating the amount of possible mobile terminals where a QoS evaluation test could be carried out. Basically these are handsets installed with specific hardware and software that would redirect the message upon delivery to the testing environment server. An apparent and persistent problem is that there are only a few number of mobile terminals available for testing leaving mobile operators and SMS aggregators with little choice in furthering their QoS tests. Real time tests implies that the tests could be issued on ordinary handsets rather than mobile terminals embedded with specific hardware, culminating in a global scale support.

4. Model for the new approach of using SMS for testing Quality of Service in wireless networks

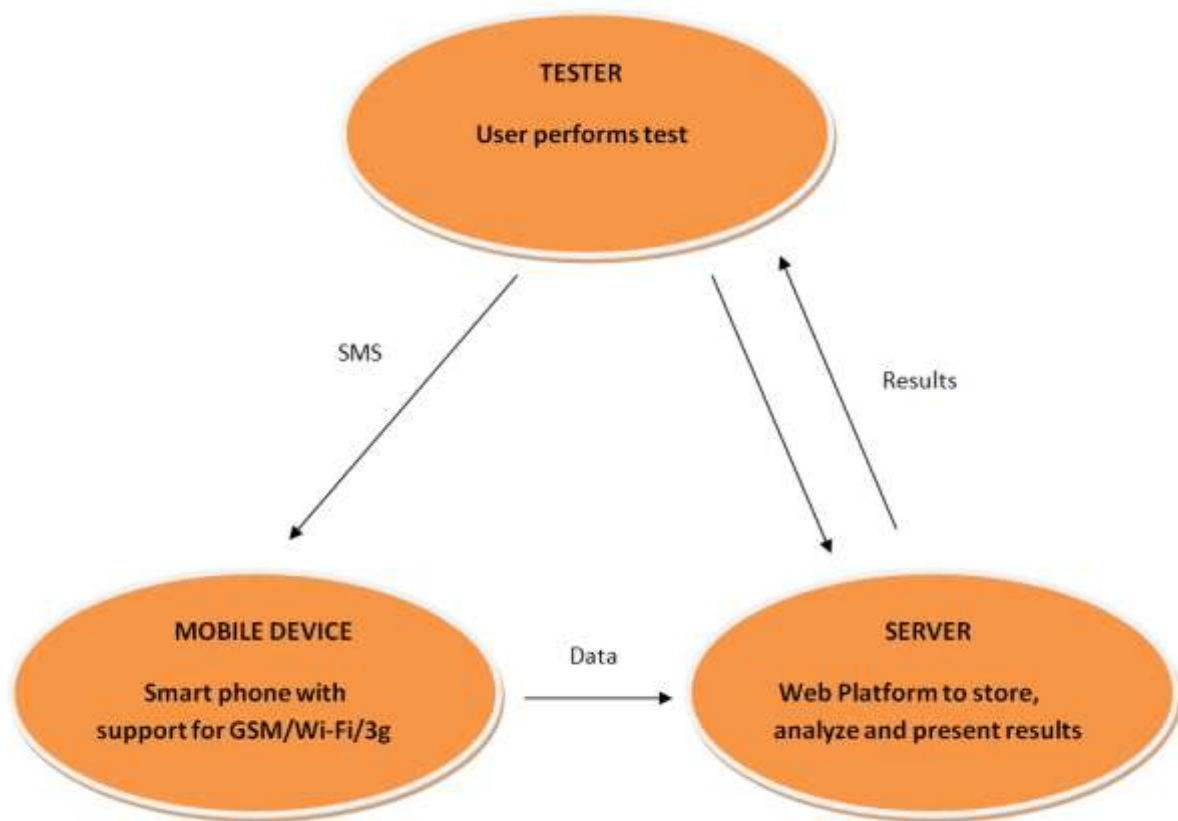


Figure 18. New model - Using SMS for testing QoS in wireless networks

The presented model for the new approach of using SMS for Quality Of Service testing in wireless networks possesses several characteristics which brings it at an advantage when compared to existing solutions:

- Combines Wireless Networks/Cellular Networks and the Internet.
- Provides the capability of global coverage for testing.
- Offers real-time validation for SMS messages.
- Cloud solution which enables instant testing.

5. Architecture of the system

The architecture of the system is based on three-tier pattern.

The advantages of three-tier applications:

- Reusability
- Reliability
- Flexibility

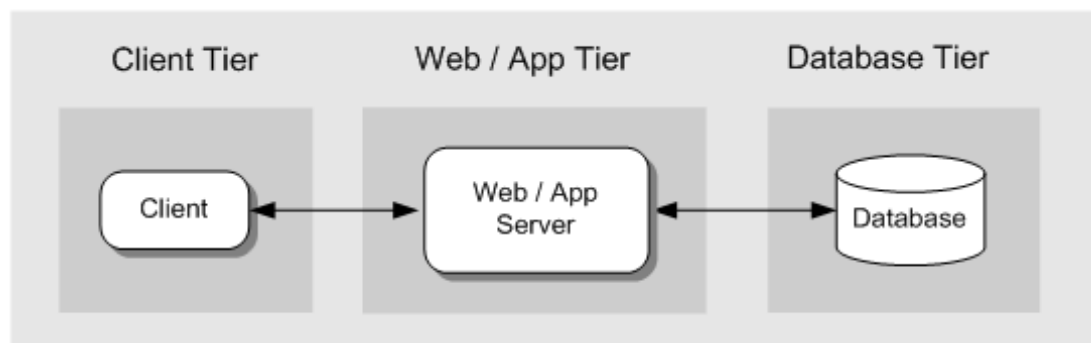


Figure 19. Three tier pattern [40]

The key technologies to be used:

- HTML5 with JavaScript (Client implementation)
- Java for Android (Client implementation)
- PHP (Service implementation)
- MySQL (Database implementation)

5.1. HTML5 & JavaScript

According to Taivalsaari, A. and Mikkonen HTML5 with JavaScript enables development across several platforms and devices [41].

5.2. Java for Android

Comparing to Apple iPhones, Android-based smartphones offer more possibility for customization. The platform for application development for Android is Java.

There are many reasons why Java was chosen as main language for Android. Java is a commonly used programming language, it can run on VM, no recompile for different phones, better security etc [42].

5.3. PHP

According to Aydin the development of the services is natural evolution of current development models [43].

PHP is hypertext preprocessor programming language. It provides web developers with a full suite of tools for building services [44].

5.4. MySQL

The relational databases are still the most popular way of organizing the collection of data in a way that supports processing, analyzing and maintaining them.

MySQL Community Server is the world's most popular open source database [45].

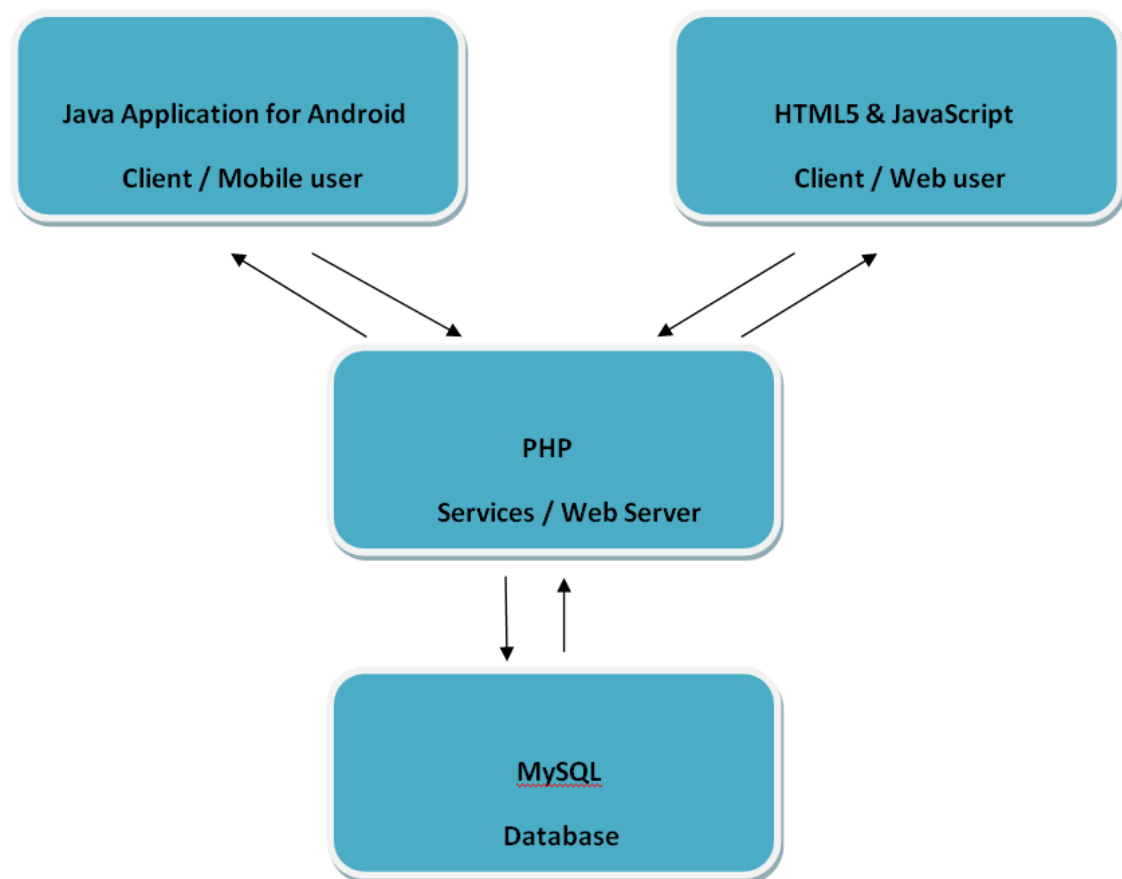


Figure 20. Applied technologies

6. Implementation

The implementation will be divided into two primary sections: mobile app (android) and testing environment (HTML 5 + JavaScript + PHP). Besides detailing the steps involved in each segment of the code snippets presented, further explanations will be provided on why particular methods, class objects etc have been chosen to perform a particular task.

6.1. Registration (Android)

At the registration screen, the user is required to enter the information for several important fields. The information entered for fields such as phone number, country and provider is very important as it will serve for many other purposes.



The screenshot shows an Android registration screen. At the top, the status bar displays '3G', signal strength, battery level, and the time '5:46 PM'. The main content area has a title 'Please add phone number in international format and without +'. Below this is a text input field containing '356455678899'. The next field is labeled 'Email' and contains 'user@gmail.com'. Below that is a 'Country List' dropdown menu showing 'Malta'. The next field is labeled 'Network List' and contains a button 'Click to select networks'. Below that is a 'GO' button. The final field is labeled 'Password' and contains masked characters '.....'.

Figure 21. Registration screen

Once the "Register New Account" button is clicked the following OnClick event will be triggered. To ensure that none of the fields have been left empty and the appropriate information has been entered a number of if..else statements have been included each validating a specific field. In the code snippet below the value stored in the phone number field is checked whether is empty or contains the "+" sign. If one of these is true this will avert the process of registration.

```

public void onClick(View view) {

    String phonenumber = inputPhoneNumber.getText().toString();
    String email = inputEmail.getText().toString();
    // String paypal = inputPaypal.getText().toString();
    // String country = inputCountry.getText().toString();
    // String provider = inputProvider.getText().toString();
    String country = spinner1.getSelectedItem().toString();
    //String provider = spinner2.getSelectedItem().toString();
    String password = inputPassword.getText().toString();

    if(phonenumber.compareTo("")== 0)
    {
        Toast.makeText(getApplicationContext(),
            "Please enter your phonenumber", Toast.LENGTH_LONG)
            .show();

        return;
    }

    if(phonenumber.contains("+")== true)
    {
        Toast.makeText(getApplicationContext(),
            "Please remove the + sign", Toast.LENGTH_LONG)
            .show();

        return;
    }
    // end cheking phone
}

```

Figure 22. Registration Validation

After each field is validated that it consists of the appropriate information, the values are grabbed and joined together in one string. This string is passed as an argument to the execute function of an instantiated AsyncTask. Before the execute method is invoked it is first checked whether there is any kind of internet connection (Wi-Fi or 3g). If yes then the execute method will be invoked, if not the process will end. This is done because the data collected will be sent via a post request through one of the AsyncTask methods. There are certain regulations which need to be taken into consideration when a task requiring internet connection is to be performed, one of them is to ensure the presence of an internet connection. Not applying this step can result in resource wastage and the automatic crash of the app on some Android OS versions. The second important regulation is that these processes will take place within a background thread, which is achieved through the use of AsyncTask.

```

if (phonenummer.trim().length() > 0 && password.trim().length() > 0 && email.trim().length() > 0 && provider != null) {

    String all = phonenummer + "|" + password + "|" + email + "|" + country + "|" + provider;

    ConnectivityManager cm =
        (ConnectivityManager)getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);

    NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
    boolean isConnected = activeNetwork != null &&
        activeNetwork.isConnectedOrConnecting();

    if (isConnected)
    {
        new MyAsyncTask().execute(all);
    }
    else {
        Toast.makeText(getApplicationContext(),
            "No network!", Toast.LENGTH_LONG)
            .show();
    }
}

else {
    // Prompt user to enter credentials
    Toast.makeText(getApplicationContext(),
        "Please make sure all the necessary fields have been entered", Toast.LENGTH_LONG)
        .show();
}
}

```

Figure 23. Instantiation of AsyncTask

Every component of an android app is by default instantiated on the same thread, which is referred to as the main thread or UI thread. The main thread is distinguished from other threads by constantly being run in a loop, awaiting for any event to be triggered from the GUI. As soon as the main thread receives a request to perform a task it will concentrate on dispatching it before shifting to another task. Excessive use of the main thread to carry out operations can lead to major slowdowns which often result in the app crashing. This tends to occur mostly when the main thread is required to carry out a long durational operation such as reading a large amount of files or waiting for a response from a http post request to the server before the task could be completed. To prevent this from occurring it is necessary to run the operations on separate threads known as worker threads or background threads. There are number of approaches that can be applied for executing specific operations on a separate thread and the most solvent has proven to be the AsyncTask class.

AsyncTask stands in short for asynchronous task, it is designed to be a helper class around Threads and Handlers. In other words it simplifies the process of creating asynchronous operations by performing durational operations on a background thread and displaying the results on the UI thread without having to manually manipulate Threads and Handlers.

Every task instance must be created and executed within the UI thread. Once a task instance execute method is called, the first method to be invoked is the `doInBackground()`. This method is run in a background thread and performs the requested operation. The results of the operation are passed on to the `OnPostExecute` method. In this particular case the `doInBackground()` method calls the `PostData` method and returns the computed result.

```

private class MyAsyncTask extends AsyncTask<String, Integer, String>{

    @Override
    //protected Double doInBackground(String... params) {
    // TODO Auto-generated method stub
    /// postData(params[0]);
    // return null;
    ///}

    protected String doInBackground(String... params) {
        // TODO Auto-generated method stub
        String res = "d";
        res = postData(params[0]); // 0 for 1
        //return null;
        return res;
    }
}

```

Figure 24. Task executed in background thread

Within the postData() method the Apache HTTP client instance is used for sending the entered data to the designated php file hosted on the server. After a post request is issued, the response from the server will be obtained and sent to the doInBackground() method.

```

public String postData(String all) {
    // Create a new HttpClient and Post Header
    String res = "0";
    HttpParams params = new BasicHttpParams();
    params.setParameter(CoreProtocolNames.PROTOCOL_VERSION, HttpVersion.HTTP_1_1);
    HttpClient httpclient = new DefaultHttpClient(params);
    HttpPost httppost = new HttpPost("http://62.210.116.25/registration.php");

    try {
        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(1);
        nameValuePairs.add(new BasicNameValuePair("all", all));
        httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
        HttpResponse response = httpclient.execute(httppost);
        res = EntityUtils.toString(response.getEntity());

    } catch (ClientProtocolException e) {

    } catch (IOException e) {

    } catch (Exception e) {}

    }

    return res;
}
}

```

Figure 25. HTTP Post request issued

The onPostExecute method represents the final step in the entire AsyncTask procedure. Results returned in the doInBackground are passed to it as a parameter. Within this method updates in the UI thread are permitted. An if and else statement is applied to validate the response, for the purpose of identifying whether the registration was successful or not.

```

protected void onPostExecute(String result) {

    //pb.setVisibility(View.GONE);
    //Toast.makeText(getApplicationContext(), "command sent", Toast.LENGTH_LONG).show();
    Toast.makeText(getApplicationContext(), result, Toast.LENGTH_LONG).show();

    String res = "OK";
    if (result.toLowerCase().contains("error")==true)
    {
        res = "NOTOK";
    }
    else
    {
        res = "OK";
    }

    if (result.toLowerCase().contains("phone number in use")==true)
    {
        res = "USED";
    }
}

```

Figure 26. Final step executed in UI thread

A successful registration would fire up an intent to start the MainActivity activity, this would transfer the user to the main menu. If in case the response received indicated that the registration for some reason was unsuccessful this would display a toast indicating a relevant reason for the cause.

```

if (res.compareTo("OK")==0)
{
    String phonenumber = inputPhoneNumber.getText().toString();
    Intent i = new Intent(getApplicationContext(), MainActivity.class);
    i.putExtra("phone", phonenumber);
    startActivity(i);

    finish();
}
if (res.compareTo("NOTOK")==0)
{
    Toast.makeText(getApplicationContext(),
        "Error in saving data, try again", Toast.LENGTH_LONG)
        .show();
}
if (res.compareTo("USED")==0)
{
    Toast.makeText(getApplicationContext(),
        "Phone number in use, you must insert a different number!", Toast.LENGTH_LONG)
        .show();
}
}

```

Figure 27. Validating result retrieved from the server

All the values sent via the HTTP post request method, will be split into a corresponding variable wherever a vertical bar is located within the concatenated string.

```
<?php

$all = $_POST['all'];

//list($user, $password, $email,$paypal,$country,$provider) = explode("|", $all);

list($user, $password, $email,$country,$provider) = explode("|", $all);
```

Figure 28. Parameters via HTTP request split into variables

Before any MySQL operations may be performed a proper connection must be established which includes the proper parameters. After defining a connection through a Mysqli instance, it must be verified whether any errors have occurred preventing a proper connection.

```
$conn = new mysqli('localhost', 'mrovoleg_wrdp1', 'ZhLiXN60YLZevb', 'mrovoleg_wrdp1');

// check connection
if (mysqli_connect_errno()) {
    exit('Connect failed: '. mysqli_connect_error());
}
```

Figure 29. MySql connection verified

Every phone number stored in the database must be unique. If at least one row is found with a phone number which is equal to the value sent, the registration process will stop.

```
$sqlQ = "SELECT * From MobileUsers WHERE phonenumber='" . $user . "'";

// perform the query and store the result
$result = $conn->query($sqlQ);

// if the $result contains at least one row
if ($result->num_rows > 0) {

    $existingUser=true;
    $response = "phone number in use " . $user;
    echo $response;
}
```

Figure 30. Phone number is validated whether it exists in the database

The last part of the registration involves inserting the values sent inside the MobileUser and MobileUsersNew table.

```
// only if user is not there
if ($existingUser==false)
{
    // INSERT IF NO EXISTING USER
    // sql query for INSERT INTO users
    // $sql = "INSERT INTO MobileUsers (phonenumber, password, email, paypaladdress,
    country, provider) VALUES ('$user',
    '$password', '$email', '$paypal', '$country', '$provider')";
    $sql = "INSERT INTO MobileUsers (phonenumber, password, email, country, provider) VALUES
    ('$user', '$password', '$email', '$country', '$provider')";
    $sql2 = "INSERT INTO MobileUsersNew (phonenumber, password, email, country, provider)
    VALUES ('$user', '$password', '$email', '$country', '$provider')";
}
```

Figure 31. Values inserted into the MobileUser and MobileUsersNew

6.2. Login (Android)

The underlying code structure facilitating the entire logging in process is similar to that of the registration menu. When the login button is pressed, the phone number and password field are validated whether being empty or not. If the correct conditions are met, which means both fields are not empty, an AsyncTask is initialized where the values entered are sent to the server. Once the server completes the process, it will return a response to the app, where based on the response received either the main menu will be activated or a toast message will appear stating that the phone number or password entered are incorrect.

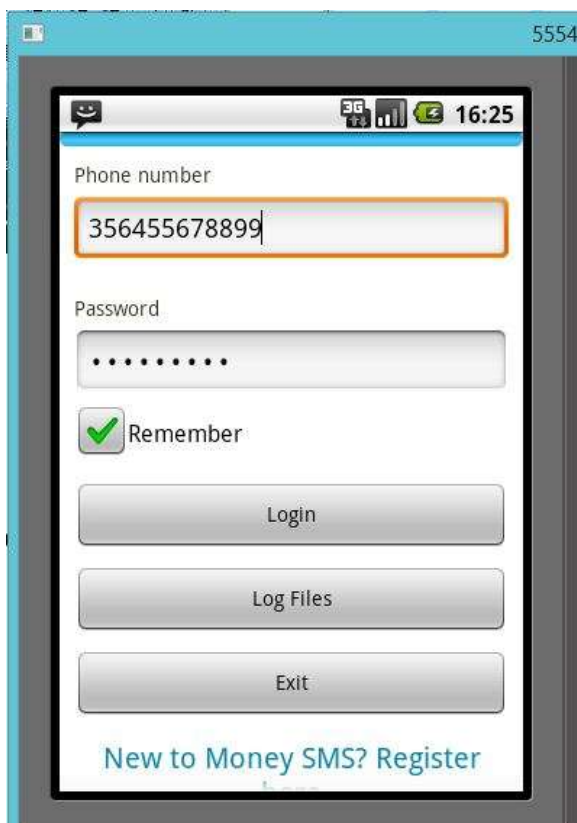


Figure 32. Login screen

A very important aspect to highlight, is the SQL command executed to update the lastconnecttoserver field in the MobileUsers table.

```
$datetimeow = date("Y-m-d H:i:s");
```

Figure 33. Current date and time obtained from the server

This field will be mainly used for a number of key factors which indicate whether the app has a properly established connection the server.

```
$sqlUpdate = "UPDATE MobileUsers SET active='1', lastconnecttoserver='" . $datetimeow . "'  
            WHERE phonenumber='" . $user . "' and password='" . $pwd . "'";  
  
$resultUpdate = $conn->query($sqlUpdate);
```

Figure 34. Specified SQL column is updated

6.3. Main Menu (Android)

The core functionalities of the app such as seizing incoming SMS messages or obtaining a delivered SMS message by going through the entire inbox are invoked at the main menu. When an SMS message is obtained, the necessary data is extracted and sent to the server. These processes are carried out without the need of user intervention or interaction, and are executed at the very instance by the detecting the presence of a incoming or delivered SMS. Two major components have been utilized in order fulfill tasks which are the Broadcast Receiver and Alarm Manager.

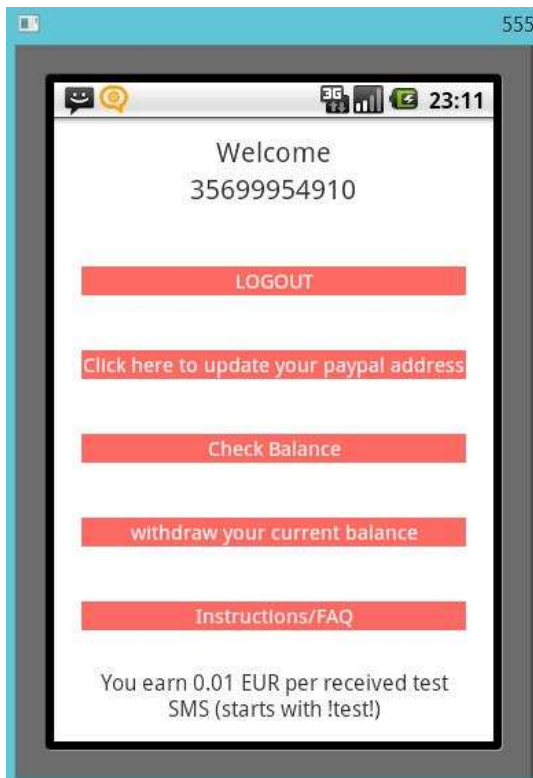


Figure 35. Main screen of app

6.3.1. Broadcast Receiver

Broadcast receivers represent components specifically implemented to receive broadcasts for various system events. In this particular case the broadcast receiver dynamically implemented in the MainActivity will listen to broadcasts that an incoming SMS message has been received by the device. Dynamic broadcast receivers will only be executed whilst the application is running in the memory.

Before a broadcast receiver can be registered it is necessary to define the intent filter. Intent filters specify the type of broadcast a component is to receive. In the code snippet below an intent filter object is instantiated with a broadcast intent type that reports the delivery of an SMS message. The filters priority value is also set to a high integer value. This type of broadcast intent falls in the category of ordered broadcasts. Ordered broadcasts are delivered to one receiver at a time depending on the priority value of the intent filter. Receivers with a higher priority value would receive the broadcast first and then be passed on to receivers with a lower priority value. A disadvantage of ordered broadcasts is that if specified the receiver that receives the broadcast first may abort the broadcast, preventing other receivers from being triggered or activated. For such a reason the priority value of the intent filter was set to the highest possible acceptable integer value which is 2147483647.

Receivers are registered by invoking the registerReceiver method. Two important arguments which are passed into the method are the name of the broadcast receiver and the intent filter. The arguments passed indicate which broadcast receiver is to handle the broadcast intent specified by the filter. The return value is a Boolean which indicates whether there is an

intent filter that matches the intent about to be broadcasted. If yes the specified receiver will be executed in the main application thread.

```
IntentFilter filter = new IntentFilter("android.provider.Telephony.SMS_RECEIVED");
filter.setPriority(2147483647);

///
//  registerReceiver(smsreceiver, new IntentFilter("android.provider.Telephony.SMS_RECEIVED"));

    registerReceiver(smsreceiver, filter);
}

//

// here the new class task
```

Figure 36. Broadcastreceiver dynamically registered

The first method that is called when a receiver is triggered is the OnRecieve method. Two arguments are passed to it, which is the context of the current activity the receiver is running in and the intent received. Additional information stored in the extras of an intent is assigned to a bundle object. Extras represent key value pairs that contain the additional information required to fulfill the requested action. Within this particular scenario the extra that is retrieved from the intent is the PDU(Protocol Data Unit). Each PDU delivered to the receiver from an SMS_RECIEVED_ACTION intent must be converted into an SmsMessage object using the createFromPdu method. After a proper SmsMessage object is constructed, the important parts within the message can be accessed. First parts to be obtained are the originating address and the entire message body.

```

private BroadcastReceiver smsreceiver = new BroadcastReceiver()
{
    @Override

    public void onReceive(Context context, Intent intent)
    {
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;

        String phonenumber = " ";
        String msg = " ";

        if (null != bundle)
        {
            String info = "Text SMS from ";
            Object[] pdus = (Object[]) bundle.get("pdus");
            msgs = new SmsMessage[pdus.length];

            String info1 = "";
            for (int i=0; i<msgs.length; i++){
                msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
                info += msgs[i].getOriginatingAddress();
                phonenumber = msgs[i].getOriginatingAddress();
                info += "\n*****TEXT MESSAGE*****\n";
                info += msgs[i].getMessageBody().toString();
                //msg = msgs[i].getMessageBody().toString();

                msg += msgs[i].getMessageBody().toString();
            }
        }
    }
}

```

Figure 37. PDU converted into SMS object

Then the service center, protocol identifier and the timestamp of when the message was received in milliseconds format are obtained.

```

info1 = "";
if (msgs[i].getServiceCenterAddress() != null)
{
    info1 = msgs[i].getServiceCenterAddress().toString();
}
info1 = info1 + "=" + msgs[i].getProtocolIdentifier();
info1 = info1 + "=" + msgs[i].getTimestampMillis();

long milliSeconds = msgs[i].getTimestampMillis();

```

Figure 38. Parts from the constructed SMS object are accessed

Since the timestamp can only be obtained in the milliseconds format it is required to have it converted to an actual date time format. The final result of the conversion will entirely depend on the local configuration settings set on the phone. There are certain cases where the number notation set on the phone are symbols rather than digits, in such situations the conversion would always be based on a US locale.

```

    long milliSeconds = msgs[i].getTimestampMillis();
    // milliSeconds is the time of real message it is datestamp in milliseconds

    // String realMilliseconds = getDate(milliSeconds, "dd/MM/yyyy hh:mm:ss.SSS");

    String realMilliseconds = getDate(milliSeconds, "yyyy-MM-dd HH:mm:ss");

    if (containsDigit(realMilliseconds)==false)
    {
        realMilliseconds = getDateDefUS(milliSeconds, "yyyy-MM-dd HH:mm:ss");
    }

    String totalDateTime = ts; // this should be the time from the phone when SMS is received

    // totalDateTime test to see the real time form mobile
    //totalDateTime = realMilliseconds;
    // info1 = info1 + "=" + totalDateTime;

    info1 = info1 + "=" + realMilliseconds;

```

Figure 39. Timestamp converted into date time format

Messages must be verified whether they possess "!test!" in their body, which is used as a designator to distinguish between messages that are to be sent to the server and not.

```

    info = phonenumber + "_" + msg;
    // msg = msg.toLowerCase();
    if (msg.contains("!test!")==true)
    // {
    // Toast.makeText(context, "location in msg", Toast.LENGTH_SHORT).show();
    // }
    // else
    {

        // here to o http to web php to save the data
        Toast.makeText(context, info+"=C", Toast.LENGTH_SHORT).show();

        abortBroadcast();
        txtName = (TextView) findViewById(R.id.name);
        String phonenumberRec = txtName.getText().toString();

        String codeofsender = msg.substring(6, 11);

        String all = phonenumberRec + "|" + msg + "|" + phonenumber + "|" + codeofsender + "|" + info1;

        ConnectivityManager cm =
            (ConnectivityManager) getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);

        NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
        boolean isConnected = activeNetwork != null &&
            activeNetwork.isConnectedOrConnecting();

```

Figure 40. Message body verified for the !test! designator

As always before an AsyncTask is executed it must be verified whether there is an available internet connection.

```
String _data = GetDateTimeWithProcedure("In SMS before network connected ");
WriteToLogTextFile(_data);

if (isConnected)
{
    String _data1 = GetDateTimeWithProcedure("In SMS network connected ");

    if (activeNetwork != null && activeNetwork.isConnected() && activeNetwork.getType() == ConnectivityManager.TYPE_WIFI)
    {
        _data1 = "WIFI " + _data1;
    }

    if (activeNetwork != null && activeNetwork.isConnected() && activeNetwork.getType() == ConnectivityManager.TYPE_MOBILE)
    {
        _data1 = "Mobile Network " + _data1;
    }

    WriteToLogTextFile2(_data1);

    // try to delete message

    new MyAsyncTask().execute(all);
}
```

Figure 41. Concatenated string passed into AsyncTask execute method

Values sent via the HTTP post request method are split into corresponding variables where a vertical bar is located within the concatenated string. The variable where the SMS message value is stored in is trimmed of any whitespaces.

```
$all = $_POST['all'];

//list($phonenumber, $smsmessage, $phonenumberofsender,$codeofsender, $info1) =
explode("|", $all);

list($phonenumber, $smsmessageTmp, $phonenumberofsender,$codeofsender, $info1) = explode("|", $all);

$smsmessage = trim($smsmessageTmp);
```

Figure 42. Concatenated string split into various variables

As mentioned before only SMS messages with the designator !test! are sent to the server. Besides this, a test id is also included. Each user registered on the testing platform possesses a test id that represents a unique identifier. Properly obtaining the test id by exploding the entire string of the SMS message between an exclamation mark and reconstructing it is a very important step. The test id will serve as the identifier to which user should results stored in the database be shown. Additional information of the SMS message stored in the info1 variable are split into the corresponding variables where an equals sign is located.

```
list($idtest1, $idtest2, $idtest3) = explode("!", $smsmessage);
$idtest = '!' . $idtest2 . '!' . $idtest3 . '!';
list($servicecenter, $protocolidentifier, $datetimestamp, $datetimestamp_mobile) = explode(
    "=", $info1);
```

Figure 43. Further formatting

The local date and time from the server is obtained in a formatted date string.

```
$datemessagerecieved = date("Y-m-d H:i:s");
echo $datemessagerecieved;
```

Figure 44. Current date and time from the server

The test id placed inside the message must be verified whether there is an account inside the database that contains a test id that precisely matches the one in the message. If the outcome of the condition is equals to true, all the of the values obtained are inserted into the SMSTransaction table.

```
if ($existingTestID == true) {
    // INSERT new transaction
    // sql query for INSERT INTO users
    $sql = "INSERT INTO SMSTransaction (phonenumber, smsmessage, phonenumberofsender,
        codeofsender, qty, info1, servicecenter, protocolidentifier, datetimesms,
        datemessagerecieved, idtest) VALUES ('$phonenumber', '$smsmessage', '$
        phonenumberofsender', '$codeofsender', 1, '$info1', '$servicecenter', '$
        protocolidentifier', '$datetimestamp_mobileresult', '$datemessagerecieved', '$idtest' )"
    ;
}
```

Figure 45. Values inserted into the SMSTransaction table

6.3.2. Alarm Manager

In order for an Android application to properly carry out the requested tasks, the environment in which the app is running needs to meet a set of conditions. Two of the most prominent conditions is that the app can fulfill certain critical tasks only when the device is not on sleep mode and the app is not prevented by the Android operating system itself. The Android operating system needs to constantly monitor the amount of memory and cpu usage each app is utilizing, in doing so it would be able to determine whether to change the state of the device to sleep mode. By some sort of criteria, the Android OS is able to decide which app may be terminable. This disadvantage can be countered by ensuring that such tasks or operations occur outside the lifetime of the application. One way of achieving this is through an alarm manager.

An alarm manager is an Android system service which allows the repeatedly triggering of alarms that execute an operation at a regular interval. Each alarm manager is instantiated indirectly by retrieving it through the getSystemService method. The operations performed

by the alarm manager are broadcasting intents to the component specified within the intent. Before the alarm manager would be able to trigger alarms which would broadcast intents it must be granted permission to do so. Granting permissions to a foreign service to execute certain components within or outside the app is done by placing the intent in a pending intent. In the code snippet below a pending intent is instantiated with the `getBroadcast()` method. This provides an indication that the operation to be performed is a broadcast. Final important step is to specify the schedule at what interval will the alarms be set off. For this application a repeating alarm was scheduled at an interval of 60 seconds.

```
int interval = 60000;

AlarmManager manager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);

Intent broadcast_intent = new Intent(this, AlarmReceiver.class);
broadcast_intent.putExtra("phoneNum", phoneNum); //data to pass

PendingIntent pendingIntent = PendingIntent.getBroadcast(this, 0, broadcast_intent, PendingIntent.FLAG_UPDATE_CURRENT);

manager.setInexactRepeating(AlarmManager.RTC_WAKEUP, System.currentTimeMillis(), interval, pendingIntent);
```

Figure 46. Alarm Manager scheduled to fire an alarm at interval of 60 seconds

Two important procedures are carried out within the stand alone receiver, which are accessing the SMS input box in order to check whether are any new messages and sending a request to the server to update the phone number's `lasconnectoserver` field inside the `MobileUsers` table.

```
public class AlarmReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        // TODO Auto-generated method stub

        // HERE method to read phone number from settings
        //Toast.makeText(context, "Alarm Triggered", Toast.LENGTH_LONG).show();

        String mem = getMemoryInfo(context);

        // Toast.makeText(context, mem, Toast.LENGTH_LONG).show();

        Calendar c = Calendar.getInstance();

        c.set(Calendar.HOUR_OF_DAY, 0);
        c.set(Calendar.MINUTE, 0);
        c.set(Calendar.SECOND, 0);

        String dateValue = String.valueOf(c);

        long dateinMilliseconds = c.getTimeInMillis();

        String datestring = String.valueOf(dateinMilliseconds);

        String phoneNum = intent.getStringExtra("phoneNum");
```

Figure 47. Current date is accessed via Calendar object and the extra phone number value is obtained

Some data of an Android application is shared among different applications and is maintained and managed by a specific interface known as the `ContentProvider`. Data managed by the content provider can be accessed via a `ContentResolver` object. The `ContentResolver` object communicates with the `ContentProvider` as a client. Whenever the `ContentProvider` receives a request from the `ContentResolver` it performs the requested operation and returns the results. Proper abstraction between the layer of data and the visual appearance of data as tables is offered by the `ContentProvider` enabling the structuring of SQL like queries.

In the code snippet below a query is constructed by using `ContentResolver.query()`. Two arguments have been supplied to the query, which is the content URI that identifies the path to the inbox table and a selection argument to retrieve SMS message rows whose date column is equal or bigger to the current date. The rows retrieved through the query are placed inside the cursor object. A cursor provides read access methods and the ability to navigate through the rows of the result.

```
Cursor cursor = context.getContentResolver().query(Uri.parse("content://sms/inbox"), null, "date>=" + datestring, null, null);
```

Figure 48. Content Provider query

The cursor must be checked whether it has moved to the first row of the results. In case a false value is returned from the if statement, it is a sign that the zero rows were retrieved in the query. Column values of a row are retrieved using the `getString()` or `getLong()` method.

```
if (cursor.moveToFirst()==true)
{
    // take first

    String address1 = cursor.getString(cursor.getColumnIndex("address"));
    String body1 = cursor.getString(cursor.getColumnIndex("body"));

    //String dateTime1=cursor.getString(cursor.getColumnIndex("date"));
    long _dtimel=cursor.getLong(cursor.getColumnIndex("date"));
    String dateTime1 = String.valueOf(_dtimel);

    String realMilliseconds = getDate(_dtimel, "yyyy-MM-dd HH:mm:ss");

    if (containsDigit(realMilliseconds)==false)
    {
        realMilliseconds = getDateDefUS(_dtimel, "yyyy-MM-dd HH:mm:ss");
    }

    String scenter1 =cursor.getString(cursor.getColumnIndex("service_center"));
    long _id1 =cursor.getLong(cursor.getColumnIndex("_id"));
```

Figure 49. First row is read

As always, the body of the message is checked whether it possesses the designator "!test!". All the necessary values are concatenated in one string and passed as an argument within the AsyncTask().execute() method. The entire procedure at the php script for parsing the entire string and inserting the obtained values inside the SMSTransaction table is identical to that discussed in the broadcast receiver section.

```

if (body1.contains("!test!")==true)
{
    String info1 = scenter1;

    info1 = info1 + "=" + "0";
    info1 = info1 + "=" + dateTime1;

    info1 = info1 + "=" + realMilliseconds;

    String codeofsender = body1.substring(6, 11);
    String all = phoneNum + "|" + body1 + "|" + address1 + "|" + codeofsender + "|" + info1;

    // send to PHP

    ConnectivityManager cm =
        (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY_SERVICE);

    NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
    boolean isConnected = activeNetwork != null &&
        activeNetwork.isConnectedOrConnecting();

    //
    if (isConnected)
    {
        new MyAsyncTask().execute(all);
    }
    else
    {

    }

}
}

```

Figure 50. Message body check for !test! designator and placed as argument within execute() method

The column values for the rest of the rows are retrieved by placing a while loop, which will continue to loop until the cursor has passed the last entry in the results set.

```

while (cursor.moveToNext())

```

Figure 51. Cursor continues to loop until the last entry in the results set

As mentioned earlier the second task that takes place is the phone number value passed to the receiver as an extra being sent to a php script.

```

new MyAsyncTaskConnectToServer().execute(phoneNum);

```

Figure 52. Phone number value passed into execute method ()

The current date and time from the server is obtained.

```
$datetimeow = date("Y-m-d H:i:s");
```

Figure 53. Current date and time

This php script executes a SQL query command that updates the lastconnecttoserver column of the row whose phonenumber column value matches the phone number value sent to the php script.

```
$sqlUpdate = "UPDATE MobileUsers SET active='1', lastconnecttoserver='". $datetimeow .  
            "' WHERE phonenumber='". $user . "'";  
  
$resultUpdate = $conn->query($sqlUpdate);
```

Figure 54. Lastconnecttoserver column updated

6.4. Registration (Client Side)

Registration form of the testing platform. The client must fill in each required field according to pattern requested. In order to ensure that all the details have been entered in the acceptable standard both client-side (HTML5 + JavaScript) and server-side (PHP) validation have been included.

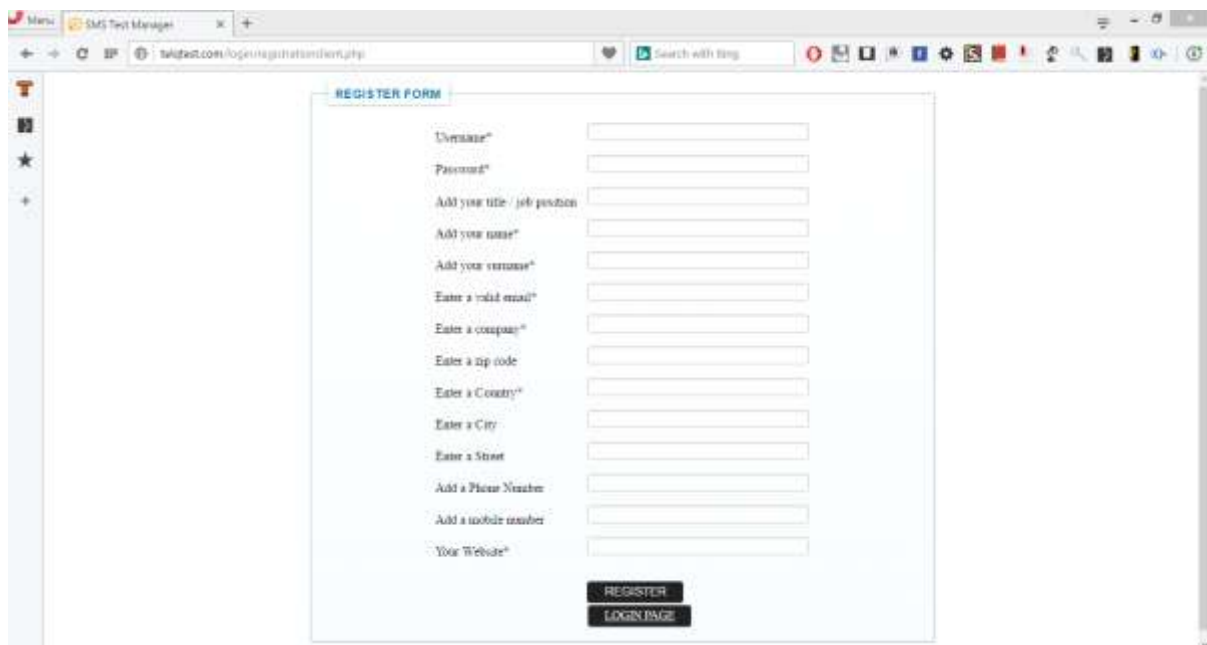
A screenshot of a web browser displaying a registration form. The browser's address bar shows the URL 'http://localhost:8080/login/registration/index.php'. The form is titled 'REGISTER FORM' and contains the following fields: Username*, Password*, Add your title / job position, Add your name*, Add your surname*, Enter a valid email*, Enter a company*, Enter a zip code, Enter a Country*, Enter a City, Enter a Street, Add a Phone Number, Add a mobile number, and Your Website*. At the bottom of the form are two buttons: 'REGISTER' and 'LOGIN PAGE'.

Figure 55. Registration of customer/client

Throughout the entire server-side registration procedure, the most prominent and specific part is the generation of a unique test id. Since each test id must be unique for every client registered in the system, it is of the outmost importance that the chances of producing an

identical result is minimized. The pattern works in the following way: the username entered in the form is reversed and then conjoined into one string with the hour, minutes and seconds value.

```
$string = $_POST['username'];
$string3 = "";

$last = strlen($string)-1;

$e = date("h:i:s");

list($u1, $u2, $u3) = explode(":", $e);

for ($i=$last; $i > -1; $i--)
{
    $string3 = $string3 . $string[$i];
}

//$milliseconds = round(microtime(true) * 1000);
//$testidA = strval($milliseconds);
$testid = "!test!";
//$testid = $testid . $testidA . $string3 . "!";

$testid = $testid . $u2.$u1 . $string3 . $u3 . "!";
```

Figure 56. Generation of unique test id

6.5. Login (Client Side)

Login form of the testing platform.

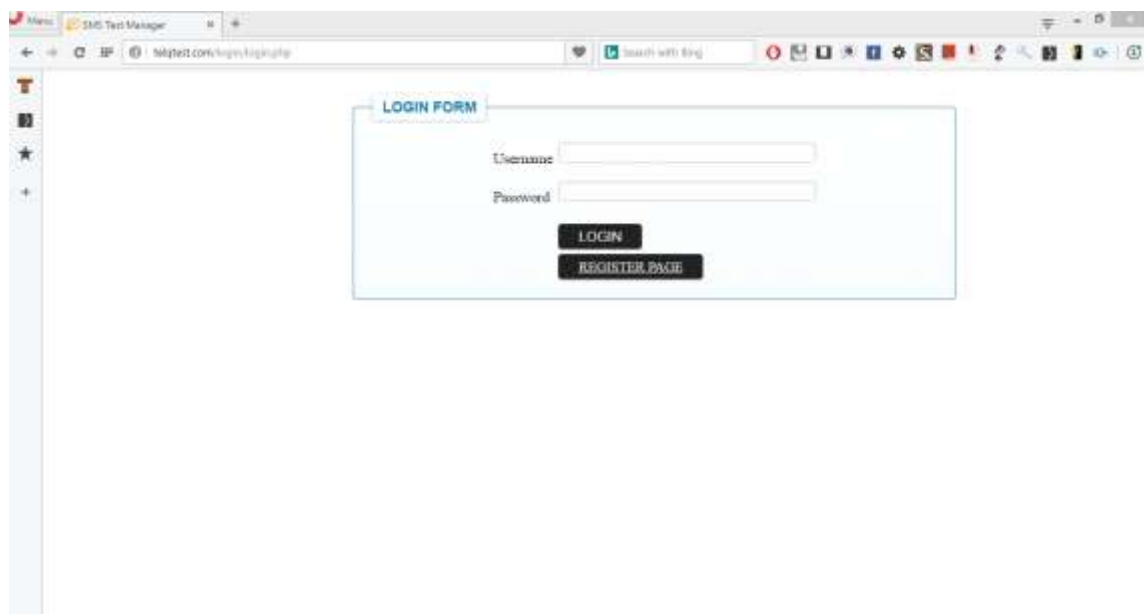
A screenshot of a web browser window showing a login form. The browser's address bar displays 'Mytest.com/login.php'. The login form is titled 'LOGIN FORM' and contains two input fields: 'Username' and 'Password'. Below these fields are two buttons: 'LOGIN' and 'REGISTER PAGE'. The browser's taskbar at the bottom shows various application icons.

Figure 57. Login screen for customer/client

6.6. Testing menu (Client Side)

Testing menu of the testing platform. It is in this menu where message results stored in the database will be displayed. Phone numbers shown by selecting the corresponding country and provider, reflect only those that have been registered in the database via the android application. In addition to this, the availability status of the phone number is also given.

Your testId is 'test171lmakun35'.

Select Country:
Malta (278)

Select Provider:
Vodafone (01)

Request to get phone numbers

Test Number: Status:
35699954910 Phone number is available

Test Results

Time received:	Country(MCC)/Network(MNC):	Phonenumber:	Sender ID:	SMS Message/Text:	SMSC:	Timestamp:	Protocol ID:
2016-04-19 16:19:01	Malta(278)/Vodafone(01)	35699954910	434343	'test171lmakun35' test711		2016-04-19 16:18:57	0
2016-04-19 16:18:43	Malta(278)/Vodafone(01)	35699954910	434345	'test171lmakun35' test710		2016-04-19 16:18:39	0

Figure 58. Main screen for testing results

Most of the data is loaded from separate PHP scripts which is done using the jQuery Ajax post() call method. Only the testid and country drop-down menu are loaded directly from the current PHP file "smsmanager2.php".

An SQL query command is executed which retrieves a row from the ClientUsers table corresponding to the username value and password value passed via the HTTP Post method. If the number of rows retrieved is bigger than 1, the test id of the result row is fetched.

```

$sql = "SELECT * From ClientUsers WHERE username='" . $username . "' and password='" .
$password . "'";

$test_id = '0';
$result = $conn->query($sql);

// if the $result contains at least one row
if ($result->num_rows > 0) {

    // $response = "<p><h1>Login Successful</h1></p><p></p>";
    // $response = "UserName in use " . $user;
    $existingUser = true;
    while($row = $result->fetch_assoc()) {

        $testid = $row["testid"];
        $test_id = $row["testid"];

    }
}

```

Figure 59. Retrieving test id of corresponding user

The variable the test id was assigned with, is placed within the HTML section.

```

Your testId is

<?php echo $testid; ?>

```

Figure 60. Test id written in html section

Similar process is repeated for the drop down menu displaying a list of countries.

```

$sql = "SELECT DISTINCT country as Country,MCC FROM CountryWithActiveMob where
lastconnecttoserver>NOW() - INTERVAL 180 MINUTE order by country";

// $sql = "SELECT 'username', 'password' FROM 'users'";
// perform the query and store the result
$result = $conn->query($sql);

// if the $result contains at least one row
if ($result->num_rows > 0) {
    // output data of each row from $result

    $i = 0;

    $dyn_box = "<select name='countrybox' id='countrybox' onchange='changeHiddenInput(this)'\>";

    if ($pcountry=="0")
        $dyn_box .= "<option selected='selected' value='0' --- </option>";
    else
        // $dyn_box .= "<option selected='selected' value='" . $pcountry . "' . $pcountry . " + " .
        $countrycode . " </option>";
        $dyn_box .= "<option selected='selected' value='" . $pcountry . "' . $countrycode . " </option>";
}

```

Figure 61. Country drop-down menu process part 1

An option tag is appended to the variable \$dyn_box. This mechanism continues until the last row of the retrieved results has been reached.

```

while($row = $result->fetch_assoc()) {
    // echo '<br /> phonenumber: '. $row['phonenumber']. ' smsmessage: '. $row['smsmessage']. '
    sender '. $row['phonenumberofsender']. ' info '. $row['info1'];

    // $phonenumber = $row["phonenumber"];
    // $smsmessage = $row["smsmessage"];
    // $sender = $row["sender"];

    //$strcountry = $row["Country"];
    $strcountry = $row["Country"];
    // $strcode = $row["CountryCode"];
    $strcode = $row["MCC"];

    $dyn_box .= "<option value='$strcountry'>".$strcountry . " (" . $strcode . ")" . "</option>";

}

$dyn_box .= "</select>";

```

Figure 62. Country drop-down menu process part 2

Generated drop down menu is written to HTML section.

```

<?php echo 'Select Country: ' ; ?>
<br>
<?php echo $dyn_box; ?>

```

Figure 63. Country drop down menu written to html section.

6.6.1. Test Results

The CheckTestIdFunctionOnce() method is invoked right after the document is loaded. Within this method an AJAX HTTP POST request is issued to the specified script and supplies the test id value to the request. Data returned by the server from the request is placed within the html element containing the id attribute "msg". In case new results are stored in the database, there is another JavaScript function whose main priority is to issue a request to the server to check whether the quantity of rows for a testID has increased.

```

<script type="text/javascript">
    function CheckTestIdFunctionOnce() {
        var testid = $("#testid").val();

        // alert(testid);

        $.post(
            "resultsPost.php",
            { testid: testid },
            function(data) {

                $('#msg').html(data);
                $('#TableResults').paging({limit:50, limitPaging: 2});

            }

        );
        // end post

        // $('#TableResults').paging({limit:5});
        // $('#msg').paging({limit:5});
    }
</script>

```

Figure 64. Ajax call for requesting generation of test results.

Results retrieved from the outer query and inner query are left joined. The inner query retrieves the last 100 rows for the specified columns from the SMSTransaction table. Rows are matched by the phonenummer column.

```

$sql="SELECT MobileUsersNew.country, MobileUsersNew.provider, MobileUsersNew.MNC,
MobileUsersNew.mcc, SMSTransaction2.* FROM ( SELECT SMSTransaction.phonenumber,
SMSTransaction.smsmessage,SMSTransaction.datetimesms,SMSTransaction.datemessagerecieved
,SMSTransaction.phonenumberofsender,SMSTransaction.servicecenter,SMSTransaction.
protocolidentifier from SMSTransaction where SMSTransaction.idtest="" $testid "" ORDER
BY datemessagerecieved DESC LIMIT 100) SMSTransaction2 LEFT JOIN MobileUsersNew ON
SMSTransaction2.phonenumber = MobileUsersNew.phonenumber";

```

Figure 65. SQL query

If the number of rows in the result set is determined to be bigger than zero, this will commence the generation of a table. Construction of the table begins by defining the table tag, headings tag, rows and data tag which are appended to the variable \$dyn_table.

```

$result = $conn->query($sql);

// if the $result contains at least one row
if ($result->num_rows > 0) {
    // output data of each row from $result

    $i = 0;
    $dyn_table = '<table id="TableResults" border="1" cellpadding="4">';

    $dyn_table .= '<caption style="text-align:center">Test Results</caption>';

    $dyn_table .= '<thead id="resultsHeader">' . '<tr><th>' . 'Time recieved:' . '</th><th>' .
        'Country(MCC)/Network(MNC):' . '</th><th>' . 'Phonenumber:' . '</th><th>' . 'Sender ID:' . '</th><th>' .
        'SMS Message/Text:' . '</th><th>' . 'SMSC: ' . '</th><th>' . 'Timestamp:' . '</th><th>' .
        'Protocol ID:' . '</th></tr>' . '</thead>';

```

Figure 66. Construction of test results table part 1

The `fetch_assoc()` function places all the results from the query into an associative array which can be looped through. Values retrieved from the specified columns are set as cells in the table. After the while loop has finished looping through the result set, the constructed table is displayed.

```

while($row = $result->fetch_assoc()) {
    $countryR = $row["country"];
    $providerR = $row["provider"];

    $mcc = $row["mcc"];
    $mnc = $row["MNC"];

    $phonenumberR = $row["phonenumber"];
    $smsmessage = $row["smsmessage"];
    //utf8_encode(string)
    $sender = $row["phonenumberofsender"];
    $info1 = $row["info1"];

    $servicecenter = $row["servicecenter"];
    $protocolidentifier = $row["protocolidentifier"];
    $originaldatetimesms = $row["datemessagerecieved"];
    $datetimesms = $row["datetimesms"];

    $dyn_table .= '<tbody>' . '<tr><td>' . $originaldatetimesms . '</td><td>' . $countryR . "(" . $mcc . ") / " . $providerR . "(" . $mnc .
        ") - " . '</td><td>' . $phonenumberR . '</td><td>' . $sender . '</td><td>' . $smsmessage . '</td><td>' . $
        servicecenter . '</td><td>' . $datetimesms . '</td><td>' . $protocolidentifier . '</td></tr>' . '</tbody>';

    }

    $dyn_table .= '</table>';
    echo $dyn_table;

```

Figure 67. Construction of test results table part2

Resulting table from the request is displayed on the client-side.

Test Results							
Time recieved:	Country(MCC)/Network(MNC):	Phonenumber:	Sender ID:	SMS Message/Text:	SMSC:	Timestamp:	Protocol ID:
2016-04-19 16:19:01	Malta(278)/Vodafone(01)	35699954910	434345	!test!171Inakim35! test711		2016-04-19 16:18:57	0
2016-04-19 16:18:43	Malta(278)/Vodafone(01)	35699954910	434345	!test!171Inakim35! test710		2016-04-19 16:18:39	0
2016-04-19 16:17:37	Malta(278)/Vodafone(01)	35699954910	434345	!test!171Inakim35! test709	null	2016-04-19 16:17:09	0
2016-04-19 16:17:37	Malta(278)/Vodafone(01)	35699954910	434345	!test!171Inakim35! test708	null	2016-04-19 16:17:06	0
2016-04-19 16:10:37	Malta(278)/Vodafone(01)	35699954910	434345	!test!171Inakim35! test707		2016-04-19 16:10:33	0
2016-04-18 13:09:11	Malta(278)/Vodafone(01)	35699954910	34345	!test!171Inakim35! test706		2016-04-18 13:09:08	0
2016-04-15 21:22:22	Malta(278)/Vodafone(01)	35699954910	343456	!test!171Inakim35! test705	null	2016-04-15 21:20:40	0
2016-04-15 21:15:31	Malta(278)/Vodafone(01)	35699954910	343456	!test!171Inakim35! test704		2016-04-15 21:15:30	0
2016-04-15 20:55:14	Malta(278)/Vodafone(01)	35699954910	343456	!test!171Inakim35! test703		2016-04-15 20:55:13	0
2016-04-15 18:17:25	Malta(278)/Vodafone(01)	35699954910	343456	!test!171Inakim35! test702DCFGF		2016-04-15 18:17:24	0

Figure 68. Test results table displayed on the client-side.

6.6.2. Provider Dropdown list

Whenever a new option is selected in the country drop down menu this triggers the `changeHiddenInput()` method. The value of the selected option is passed as an argument, which is placed as a parameter to the request issued to the server. Data retrieved from the request is a drop-down menu which displays a list of network operators corresponding to the country selected. At the server side, the process for generating the drop down menu is identical to the country drop down menu.

```
<script type="text/javascript">
function changeHiddenInput (objDropDown)
{
    var v_pcountry = objDropDown.value;

    $.post(
        "resultsProviderPost.php",
        { pcountry: v_pcountry },
        function(data) {
            $('#msgproviders').html(data);
            //$('#fileField').html(data);
        }
    );
}
```

Figure 69. Ajax call for requesting generation of provider drop down menu.

Resulting drop down menu from the request is displayed on the client-side.

Your testId is !test!1711nakim35!

Select Country:

Select Provider:

hone numbers

Figure 70. Provider drop-down menu displayed on client side.

6.6.3. Phone number status verification

Once a country and network provider have been selected, clicking on the "request to get phone numbers" button will generate a list of phone numbers and their current status. The method that is invoked is the CheckPhoneNumbers(). Three values are placed as parameters to the request URL which are the country, provider and MNC. Being that the status of a phone number is inclined to change, request calls will be issued to server at interval of 20 seconds.

```
<script type="text/javascript">
    function CheckPhoneNumbers() {

        var autocheckphone = document.getElementById("autocheckphonenumber");
        autocheckphone.value = "1";

        var v_pcountry = $("#country").val();
        var v_provider = $("#provider").val();
        var v_mnc = $("#textmnccode").val();
        // alert(v_pcountry);
        // alert(v_provider);
        $.post(
            "resultsPhonesPost.php",
            { pcountry: v_pcountry, provider: v_provider, mnc: v_mnc },
            function(data) {

                $('#msgphones').html(data);
                //$('#fileField').html(data);

            }

        );
    }
</script>
```

Figure 71. Ajax call for requesting generation of available phone numbers

Whilst the while loop is still looping through the array, the status of each phone number is determined by, whether the time from the lastconnecttoserver is ahead from the server time. The server time is always subtracted 5 minutes from it before a comparison takes place. As long as the time from the lastconnecttoserver is larger than the server time, the phone number status will be placed as available. If the lastconnectserverfield is not updated in the next 5 minutes, this will place the status of the phone number as not available.

```
$datetimeow = date("Y-m-d H:i:s", strtotime("-5 minutes"));
$datetimeFromUser = $row["lastconnecttoserver"];

//if ($active == 1) {
    if ($datetimeFromUser > $datetimeow) {

        // $dyn_table1 .= '<tr><td>' . $phonenumber . '</td></tr>';
        // $dyn_table1 .= '<tr><td>' . "Phone number is available" . '</td></tr>';
        $dyn_table1 .= '<tr><td>' . $phonenumber . '</td>' . '<td style="color:green;">' . "Phone number is available" . '</td></tr>';

    }
    // else if ($active == 0)
    else
    {

        // $dyn_table1 .= '<tr><td>' . $phonenumber . '</td></tr>';
        // $dyn_table1 .= '<tr><td>' . "Phone number is not available" . '</td></tr>';
        $dyn_table1 .= '<tr><td>' . $phonenumber . '</td>' . '<td style="color:red;">' . "Phone number is not available" . '</td></tr>';

    }
}
```

Figure 72. Phone number list and status generated

Phone number result corresponding to the selected country and provider. The phone number status is shown as active.

Your testId is !test!1711nakim35!

Select Country:

Malta (278) ▾

Select Provider:

Vodafone [01] ▾

Request to get phone numbers

Test Numbers: Status:

35699954910 Phone number is available

Figure 73. Phone number indicated as available

Phone number status shown as inactive.

Your testId is !test!1711nakim35!

Select Country:

Malta (278) ▾

Select Provider:

Vodafone [01] ▾

Request to get phone numbers

Test Numbers: Status:

35699954910 **Phone number is not available**

Figure 74. Phone number indicated as unavailable

7. Results

The pilot project which included the complete software mentioned-above began in the 1st of January 2016. The results are the following:

Countries

Afghanistan
Algeria
Angola
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Bosnia and Herzegovina
Botswana
Brazil
Brunei
Bulgaria
Cambodia
Cameroon
Canada
China
Colombia
Costa Rica
Curacao
Czech Republic
Denmark
Dominica
Dominican Republic
Ecuador
Egypt
Fiji
Finland
France
Georgia
Germany
Ghana
Greece
Guyana

Hong Kong
India
Indonesia
Iran
Iraq
Ireland
Israel
Italy
Ivory Coast
Jamaica
Japan
Jordan
Kazakhstan
Kenya
Kuwait
Kyrgyzstan
Laos
Latvia
Lebanon
Libya
Lithuania
Macao
Macedonia (FYROM)
Malawi
Malaysia
Malta
Mauritius
Mexico
Moldova
Monaco
Mongolia
Morocco
Myanmar
Nepal
Netherlands
New Zealand
Nigeria
Norway
Oman
Pakistan
Philippines
Poland
Portugal
Puerto Rico
Romania
Russian Federation
Rwanda
Saudi Arabia
Serbia
Singapore

Slovakia
Slovenia
South Africa
South Korea
Spain
Sri Lanka
Sudan
Suriname
Sweden
Syria
Taiwan
Tajikistan
Tanzania
Thailand
Trinidad and Tobago
Tunisia
Turkey
Turkmenistan
Uganda
Ukraine
United Arab Emirates
United Kingdom
United States of America
Uzbekistan
Venezuela
Vietnam
Yemen

The total number of the **countries** is **118**

Mobile Operators

Inc (T-Mobile USA)
Inc.
LLC
3Mob
A-Mobile
Abkhazia
Aditya Birla Telecom Limited (IDEA BEHAR)
Advantage Cellular Systems Inc (Advantage)
AINA
Aircel
Aircel Limited (AIRCEL)
Airtel
Airtel Wireless LLC
AIS
Alaska Wireless Communications
Alltel

Altel (CDMA)
Amarillo License L.P.
American Cellular Corporation
Asia Cell
AT&T Mobility
AT&T Mobility (AT&T)
AXIS
Azercell
Babilon-M
Bakcell
Banglalink
Barablu (MVNO)
BASE
BaTelCo
Baykalwestcom
Beeline
Bell Mobility
Bharat Sanchar Nigam Limited (CellOne Andhra Pradesh)
Bharat Sanchar Nigam Limited (CellOne Bihar)
Bharat Sanchar Nigam Limited (CellOne Chennai)
Bharat Sanchar Nigam Limited (CellOne Haryana)
Bharat Sanchar Nigam Limited (CellOne Karnataka)
Bharat Sanchar Nigam Limited (CellOne Punjab)
Bharat Sanchar Nigam Limited (CellOne Tamil Nadu)
Bharat Sanchar Nigam Limited (CellOne Uttar Pradesh (East))
Bharti Airtel Limited (AirTel)
Bharti Hexacom Ltd (AIRTEL)
Bite
Blue Sky Communications
Bouygues
BTC Mobile
Celcom
Cell C
Cellcom
China Mobile
Cingular Wireless
Claro
Cosmote CDMA
Cricket Communications
Dialog
DiGi
Digicel
DigiCell
DigiMobil (MVNO)
Digitel
Djezzy
DSTCom
dtac
dtac 3G
du

E-Plus
Elisa
Emtel
Endless Mountains Wireless, LLC
Eronet
Etisalat
Everything Everywhere
FarEasTone
Free Mobile
Geocell
Glo
Globe
GO
Golan Telecom
GrameenPhone
GT&T Cellink Plus
HiTS-UNITEL
Hotlink
IAM
IDEA Cellular Limited
IDEA Cellular Limited (Idea (Escotel) Haryana)
IDEA Cellular Limited (Idea (Escotel) UP West)
IDEA Cellular Limited (IDEA Cellular - Andhra Pradesh)
IDEA Cellular Limited (IDEA Cellular - Madhya Pradesh)
IDEA Cellular Limited (IDEA Cellular - Maharashtra)
IDEA Cellular Limited (IDEA)
IDEA Cellular Ltd
India
Indigo Wireless
INDOSAT
INWI
IR-MCI
Iusacell / Unefon
Jazztel (MVNO)
Kcell
Kolbi ICE
KPN
KT
Kyivstar
Libyana
life:)
LIME
Loop Cellular - Kolkata
Lycamobile
Lycamobile (MVNO)
Lycamobile (MVNO)
M-tel
M1
MagtiCom
Mahanagar Telephone Nigam Ltd (Mahanagar Telephone Nigam)

Maxis
Meditel
MegaCom
MegaFon
MEO
Metfone
Metro PCS
MobiCom
MobiFone
Mobile Tel Inc.
Mobilink
Mobilis
Mobily
Mobinil
Mobitel
Moldcell
Monaco Telecom
MOTIV
MOVICEL
Movilnet
Movistar
MPT
mtc touch
MTN
MTPCS
MTS
Nar Mobile
Nepal Telecommunications
NetCom
Nextel
NTT DoCoMo, Inc (DOCOMO)
O!
O2
Oi
Oman Mobile
Omnitel
One
Ooredoo
Optus
Orange
PCCW Mobile
Pelephone
Personal
Play
Plus
Proximus
Reliance Communications Limited (Reliance Communications Limited)
Reliance Telecom Private Ltd (RELIANCE TELECOM LIMITED)
Reliance Telecom Pvt Ltd (CDMA)
Rightel

Robi
Roshan
Rostelecom
Safaricom
Simyo (MVNO)
SingTel
SK Telecom
Smart
SMART (CDMA)
Smart Mobile
Smartfren (CDMA)
Sonera
Spice Communications Limited (Spice Telecom)
Sprint
StarHub
STC / Al Jawal
Sun
Syriatel
T-Mobile
T-Mobile USA
Taiwan Mobile
Tata Teleservices / Andhra Pradesh
TATA TELESERVICES LTD
Tata TeleServices(Maharashtra) Limited
Tcell
Telcel
Tele2
Telekom
Telenor
Telesur
Telkom Mobile
Telkomsel
Telstra
Tigo
TIM
TM-Cell
True Move
True Move-H
Tune Talk (MVNO)
Turkcell
U Mobile
Ucell
Ufone
Umniah
uninor
Unite (GSM)
UNITEL
US Cellular
Velcom
Verizon

Verizon Wireless
Videocon Telecom
Vietnamobile
Viettel
Vinaphone
VIP
Viva
Viva Cell (MTS)
Vivo
Vodacom
Vodafone
Vodafone Essar Cellular Limite
Vodafone Essar Cellular Limited (Vodafone - Kerala)
Vodafone Essar Cellular Limited (Vodafone - Maharashtra & Goa)
Vodafone Essar Digilink Limited (Vodafone - Haryana)
Vodafone Essar Gujarat Limited (Vodafone - Gujarat)
Vodafone Essar Mobile Services Limited (Vodafone - Delhi)
Vodafone Essar South Limited (Vodafone - Andhra Pradesh)
Vodafone Essar South Limited (Vodafone - Punjab)
Vodafone Essar Spacetel Limited (Vodafone NORTHEAST)
Vodafone Essar Spacetel Limited (Vodafone ORISSA)
Warid
Wind
Wireless Communications Venture
XL
Yoigo
Zain
Zong

The total number of **mobile operators** is **244**.

Mobile users (different mobile number)

The total number of **mobile users** (different mobile number) is **1862**.

SMS

The total number of **SMS** is **21260**.

The software solution based on the new framework proved to be applicable to the world wide areas with the high level of reliability.

8. Conclusions

As the world is shifting from the industrial to the informatics era, the Internet remains the most dominant global phenomena being further integrated with various telecommunication networks (GSM, GPRS, Wi-Fi). This integration is often considered to be one of the key features that has led to the foundation of the informatics society. With such a rapid and structured advancement of the communication networks, substantial changes have taken place in the goods and services sector. Demands are no longer just goods and services of a higher caliber but rather their **realization in real time** e.g. nonstop.

Short Message Service (SMS) remains one of the most reliable and robust services in the mobile telecommunication networks. By such standards, it represents an ideal component for testing availability, bandwidth and territory coverage, in other words the overall quality of mobile telecommunication networks. On the other hand, SMS has become very important in the following types of communication:

- Messages exchanged between two mobile devices (without human intervention)
- Application-to-person(A2P), where messages are sent from an application to a mobile device and vice versa.
- Marketing campaigns based upon the transmission of a large number of messages (bulk SMS)

Contrary to the previous solutions in the domain of SMS quality of service which are primarily based on the statistical processing of data, the new framework possesses primary benefits in:

- ✓ Real time confirmation for sent and received SMS messages
- ✓ Using realistic samples from mobile users
- ✓ Unlimited territorial coverage

The Software solution developed on the new framework is based on the following modern technologies:

- Java Android application for the mobile phones
- Web cloud system based and developed using HTML5, JavaScript, PHP, MySQL

Pilot project indicated positive results in the usage of the new approach for SMS QoS. New inquiries specifically in the possibility applying this concept in the area of Voice over IP are the directives for further development within this area.

References

- [1] Harrington, M. (2007). Understanding SMS-Practitioner's Basics.
- [2] Analysis and forecasts for Mobile Messaging worldwide. (2014). Portio Research.
- [3] Leppäniemi, M. (2008). Mobile marketing communications in consumer markets. (33).
- [4] Miller, M. (2013). *Theoretical conception of a cells in a cellular network*. [image] Available at: <http://www.quepublishing.com/articles/article.aspx?p=2021961> [Accessed 11 Dec. 2015].
- [5] Ethw.org, (2015). *Cellular Base Stations - Engineering and Technology History Wiki*. [online] Available at: http://ethw.org/Cellular_Base_Stations [Accessed 14 Dec. 2015].
- [6] Withoutthecat.com, (2010). *Learn about what is on a cell tower*. [online] Available at: <http://www.withoutthecat.com/Antenna1900.html> [Accessed 14 Dec. 2015].
- [7] Frequency re-use pattern in cellular networks. (n.d.). [image] Available at: <http://www.telecomabc.com/c/cellular.html> [Accessed 11 Dec. 2015].
- [8] Huurdeman, A. (2003). *The worldwide history of telecommunications*. New York: J. Wiley.
- [9] Jolis, A. and Jones, D. (2007). *EU Seeks To End Mandatory GSM for 900Mhz* - Source. [online] cellular-news. Available at: <http://www.cellular-news.com/story/25102.php> [Accessed 15 Dec. 2015].
- [10] ZDNet, (2007). *Happy 20th birthday, GSM* | ZDNet. [online] Available at: <http://www.zdnet.com/article/happy-20th-birthday-gsm/> [Accessed 15 Dec. 2015].
- [11] Poole, I. (n.d.). *Simplified GSM architecture*. [image] Available at: http://www.radio-electronics.com/info/cellulartelecomms/gprs/gprs_tutorial.php [Accessed 12 Dec. 2015].
- [12] 3gpp.org, (n.d.). *GPRS & EDGE*. [online] Available at: <http://www.3gpp.org/technologies/keywords-acronyms/102-gprs-edge> [Accessed 25 Dec. 2015].
- [13] Docstore.mik.ua, (n.d.). *Overview of GSM, GPRS, and UMTS*. [online] Available at: http://docstore.mik.ua/univercd/cc/td/doc/product/wireless/moblwrls/cmx/mmg_sg/cmxgsm.htm#wp1033806 [Accessed 6 Jan. 2016].
- [14] Radio-electronics.com, (n.d.). *What is GPRS | General Packet Radio Service | Radio-Electronics.com*. [online] Available at: http://www.radio-electronics.com/info/cellulartelecomms/gprs/gprs_tutorial.php [Accessed 8 Jan. 2016].
- [15] Pereira, V. and Sousa, T. (n.d.). Evolution of Mobile Communications: from 1G to 4G.

- [16] Parks, B. (2008). *The Big Picture*. [online] WIRED. Available at: <http://www.wired.com/2000/10/kahn/> [Accessed 11 Jan. 2016].
- [17] Ctia.org, (2014). *Archive*. [online] Available at: <http://www.ctia.org/resource-library/press-releases/archive/ctia-annual-survey-2013> [Accessed 12 Jan. 2016].
- [18] Vununet.com, (2000). *Will Wap's Call Go unanswered ?*. [online] Available at: <http://www.v3.co.uk/v3-uk/analysis/2131787/wap-call-unanswered> [Accessed 12 Jan. 2016].
- [19] Mousa, A. (2012). Prospective of Fifth Generation Mobile Communications. *International Journal of Next-Generation Networks*, 4(3), pp.11-30.
- [20] Itu.int, (2007). *Press Release: ITU defines the future of mobile communications*. [online] Available at: http://www.itu.int/newsroom/press_releases/2007/30.html [Accessed 20 Jan. 2016].
- [21] Smith, C. and Collins, D. (2001). *3G Wireless Networks*. McGraw-Hill Telecom Professional.
- [22] Chakraborty, A. (2013). A Study on Third Generation Mobile Technology (3G) and Comparison among All Generations of Mobile Communication. *International Journal of Innovative Technology & Adaptive Management*, 1(2).
- [23] Internet Of Things: Science Fiction Or Business Fact?. (n.d.). *Harvard Business Review*.
- [24] Transition to 4G. (2010). Rysavy Research/3G Americas.
- [25] Trosby, F., Harris, I., Hillebrand, F. and Holley, K. (2010). *Short message service (SMS)*. Chichester, West Sussex, U.K.: Wiley.
- [26] GSM History: History of GSM, Mobile Networks, Vintage Mobiles, (2013). SMS. [online] Available at: <http://www.gsmhistory.com/sms/> [Accessed 17 Nov. 2015].
- [27] Hord, J. (2015). *How SMS works*. [image] Available at: <http://computer.howstuffworks.com/e-mail-messaging/sms2.htm> [Accessed 23 Nov. 2015].
- [28] Gupta, P. (2015). *Diagrammatic representation of network elements in a GSM network*. [image] Available at: <http://www.wirelessdevnet.com/channels/sms/features/sms.html> [Accessed 23 Nov. 2015].
- [29] Guthery, S. and Cronin, M. (2002). *Mobile application development with SMS and the SIM toolkit*. New York: McGraw-Hill.
- [30] Le Bodic, G. (2005). *Mobile messaging technologies & services*. New York: Wiley.
- [31] Developershome,(n.d.). [image] Available at: http://www.developershome.com/sms/sms_tutorial.asp?page=sms Gateway [Accessed 20 Dec. 2015].
- [32] Definitions of terms related to quality of service. (2008). *E Series*.

- [33] Carvalho de Gouveia, F. and Magedanz, T. (n.d.). Quality of Service in Telecommunication Networks. *Telecommunication Systems And Technologies*, 2.
- [34] Quality of Service Parameters for Supporting Service Aspects Models for Telecommunication Services. (2011). *ITU Recommendation*, ITU-T E.803.
- [35] Rahko, K., Herzberg, S. and Rahko, T. (1972). Grade of service and human factors. *In 6th International Symposium on Human Factors in Telecommunications*.
- [36] Speech Processing, Transmission and Quality Aspects (STQ). (2008). *ETSI TS*, V1.6.2.
- [37] Waddt, A., Bruck, G., Jung, P., Kowalzik, M., Trapp, T. and Begall, C. (2005). A reconfigurable QoS monitoring framework for professional short message services in GSM networks. *Services Computing*, 2005 IEEE International Conference, 2, pp.22-29.
- [38] M. Canlas, K.P. Cruz, M.K. Dimaricut, P. Uyengco, G. Tangonan, M.L. Guico, N. Libatique, and C. Pineda. (2010). A Quantitative Analysis of the Quality of Service of Short Message Service in the Philippines. *IEEE International Conference*, pp.710 –714.
- [39] Xiaoqiao Meng, P. Zerfos, V. Samanta, S.H.Y. Wong, and Songwu Lu .(2007). Analysis of the Reliability of a Nationwide Short Message Service. *In INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, IEEE, pp.1811 –1819.
- [40] Guidanceshare.com. (2009). *Application Architecture Guide - Chapter 9 - Layers and Tiers - Guidance Share*. [online] Available at: http://www.guidanceshare.com/wiki/Application_Architecture_Guide_-_Chapter_9_-_Layers_and_Tiers [Accessed: 11 Jan 2016].
- [41] Taivalsaari, A. and Mikkonen, T. (2011). The web as an application platform: The saga continues. In: *Proc. of the 37th Euromicro Conference on Software Engineering and Advanced Applications (SEAA'2011, Oulu, Finland, August 30 - September 2, 2011)*, pp. 170-174.
- [42] Shweta, S. (2014). *Introduction to Java for Android Application Development* [online] Available at <https://blog.udemy.com/java-for-android/> [Accessed: 11 Mar 2016].
- [43] Aydin, G. (2007). *Service Oriented Architecture for Geographical Information Systems Supporting Real Time Data Grids*. Ph.D. Indiana University.
- [44] Php.net. (2016). [online] Available at: <http://php.net/> [Accessed: 10 Feb 2016].
- [45] MySQL.com (2016) [online] Available at: <http://dev.mysql.com/downloads/> [Accessed: 15 Feb 2016].