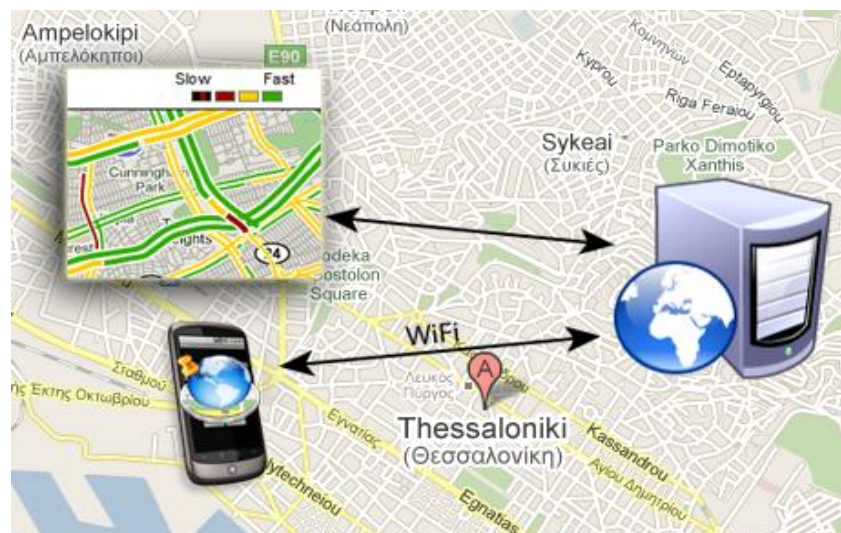




Department of Informatics
Alexander Technological Educational
Institute of Thessaloniki



**Design and implementation of a real-time cooperative system
for the estimation of traffic conditions on the road network
using mobile phone devices with enabled GPS receivers.**



By Pachatouridis Konstantinos

Supervisor: **Dr. Chatzimisios Periklis**

Thessaloniki, Greece

February 2011

Acknowledgments

My deepest and most sincere gratitude is to my supervisor, Dr. Chatzimisios Periklis, for his trust, continuous support and priceless guidelines during this thesis.

The theme of this thesis was an "idea" that arose during my practical training for the Department of Informatics / Alexander Technological Educational Institute of Thessaloniki, at the Hellenic Institute of Transportation (HIT) of the Center for Research and Technology Hellas (CERTH).

The current thesis participated in Contest for Applied Research and Innovation organized by Hellenic Federation of Enterprises.

I am also indebted to all the people of the administration of HIT and especially to Dr. Georgia Aifantopoulou, Head of Sector B: Infrastructure & Demand of HIT that provided me with the opportunity to fulfill my practical training. I would also like to thank all my colleagues at HIT for their support during the preparation of this thesis.

I am truly grateful to Mr. Evaggelos Mitsakis, research associate of H.I.T, for his scientific support and useful advice in all stages of this work and especially for his idea to deal with such issues in this paper.

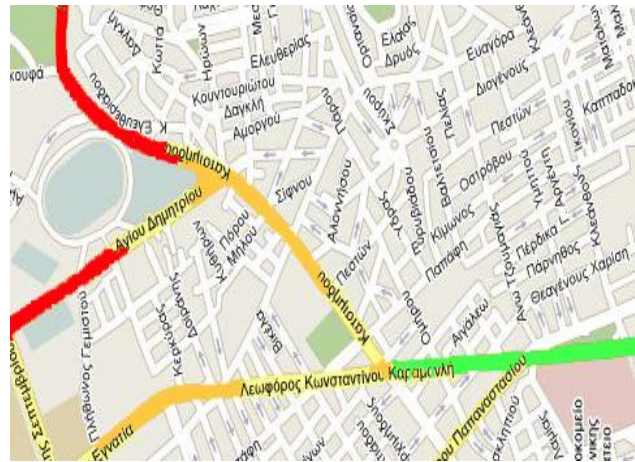
Finally, I would like to express my sincerest appreciation to my colleagues and friends for their precious support, help and discussions along with my deepest thanks and appreciations to my family.

Abstract

The proposed cooperative system provides the capability of informing users about the traffic situation on roads, at real time, based on information collected from users themselves and then return to them processed. Through an application for mobile devices, the user can be informed about traffic conditions, using data from all users in motion (Floating car data) that also utilize the system.

The user interface of the application shows to the user a map, according to each point he/she is on, on which the movement on the streets is reflected, with appropriate color depending on the actual traffic condition. For example, red roads are busy, orange ones have moderate traffic and green roads do not face any problem.

The application is compatible with mobile phones with integrated or external Global Positioning System (GPS) receiver and internet connectivity via General Packet Radio Service (GPRS) or Wireless Fidelity (Wi-Fi), so as to enable both the identification of the user's location (from GPS device) and the sending and receiving of the corresponding data of the service (via Internet connection).



Throughout the development of the application it was essential to develop such techniques to minimize the cost of information available, ensuring the privacy of users and to develop a user-friendly application. The major advantage of the system described is the use of virtual lines, the real-time function, the full anonymity of the data used along with the ability to provide service using widely available equipment which is already available to most users (mobile phones) in contrast to the expensive infrastructure required to measure the relevant information (traffic condition).

The development of the application was made in the programming language JAVA (Mobile Edition), the web server used for creating the service required is Glassfish 3

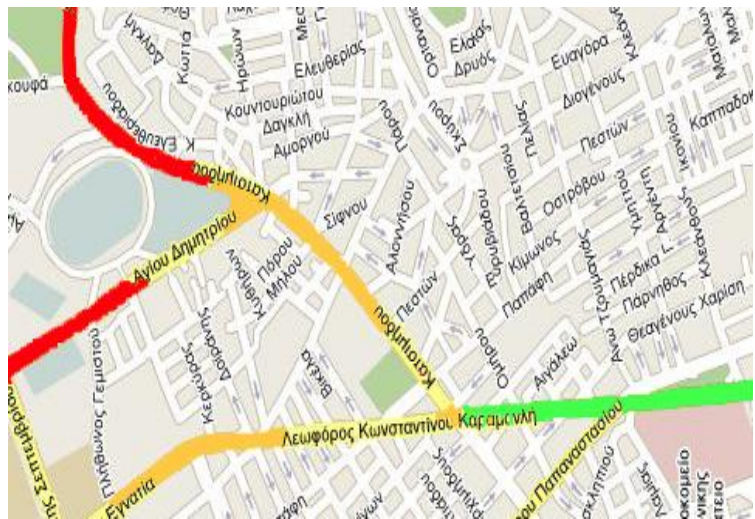
and the database used is Postgres Plus Advanced Server 8.4 with geographic data support (PostGIS). The full source code of the application is included in the cd attached to the current thesis.

Keywords: Real-time traffic updates, cooperative mobility systems, zero-infrastructure systems.

Abstract (Greek)

Το συνεργατικό σύστημα που περιγράφεται έχει ως σκοπό την πληροφόρηση των χρηστών για την κυκλοφοριακή κατάσταση στους δρόμους, σε επίπεδο πραγματικού χρόνου, βάσει των πληροφοριών που συλλέγονται από τους ίδιους τους χρήστες και επιστρέφουν επεξεργασμένες σε αυτούς. Ο χρήστης έχει τη δυνατότητα, μέσω μιας εφαρμογής για συσκευές κινητής τηλεφωνίας, να ενημερώνεται για την κίνηση στους δρόμους, χρησιμοποιώντας δεδομένα που προέρχονται από όλους τους χρήστες που κινούνται και ταυτόχρονα χρησιμοποιούν το σύστημα.

Η εφαρμογή απεικονίζει στον χρήστη ένα χάρτη, σύμφωνα με το εκάστοτε σημείο που βρίσκεται αυτός, όπου αποτυπώνεται η κίνηση πάνω στους δρόμους, με κατάλληλο χρωματισμό, ανάλογα με τη πραγματική κατάσταση της κυκλοφορίας. Για παράδειγμα, με κόκκινο οι δρόμοι με μεγάλη κίνηση, με πορτοκαλί αυτοί με μέτρια κίνηση και με πράσινο οι δρόμοι που δεν αντιμετωπίζουν κάποιο πρόβλημα.



Η εφαρμογή είναι συμβατή με συσκευές κινητής τηλεφωνίας που διαθέτουν ενσωματωμένο (ή και εξωτερικό) δέκτη GPS καθώς και δυνατότητα σύνδεσης στο διαδίκτυο με την χρήση υπηρεσιών General Packet Radio Service (GPRS) ή Wireless Fidelity (Wi-Fi), έτσι ώστε να καθίσταται δυνατός τόσο ο προσδιορισμός της θέσης του χρήστη (από την συσκευή GPS) όσο και η αποστολή και λήψη των αντίστοιχων δεδομένων της υπηρεσίας (μέσω της σύνδεσης στο διαδίκτυο).

Απαραίτητη κρίθηκε η ανάπτυξη τεχνικών για την ελαχιστοποίηση του κόστους της διακινούμενης πληροφορίας, τη διασφάλιση των προσωπικών δεδομένων των χρηστών καθώς και για την ανάπτυξη μιας φιλικής προς τον χρήστη εφαρμογής. Το σημαντικότερο πλεονέκτημα του συστήματος που περιγράφεται είναι η χρήση των

εικονικών γραμμών, η λειτουργία σε πραγματικό χρόνο, η πλήρης ανωνυμία των δεδομένων που χρησιμοποιούνται καθώς και η δυνατότητα παροχής μιας υπηρεσίας μέσω ευρέως διαδεδομένων συσκευών που ήδη διαθέτουν οι περισσότεροι χρήστες (κινητά τηλέφωνα) σε αντίθεση με την ακριβή υποδομή που απαιτείται σήμερα για την μέτρηση της αντίστοιχης πληροφορίας (κατάσταση κυκλοφορίας).

Η υλοποίηση της εφαρμογής έγινε σε γλώσσα προγραμματισμού JAVA (Mobile Edition), ο web server που χρησιμοποιήθηκε για την δημιουργία των απαραίτητων service είναι ο GlassFish 3 και η βάση δεδομένων που χρησιμοποιήθηκε είναι η Postgres Plus Advanced Server 8.4 με υποστήριξη γεωγραφικών δεδομένων (PostGIS). Ο κώδικας της υλοποίησης του συστήματος διατίθεται αυτούσιος σε cd που επισυνάπτεται στην παρούσα πτυχιακή.

Λέξεις-κλειδιά: Ενημέρωση κυκλοφορίας σε πραγματικό χρόνο (real time traffic information), συνεργατικά συστήματα κινητικότητας (cooperative mobility systems), συστήματα μηδενικής υποδομής (zero-infrastructure systems).

Glossary

Term or Abbreviation	Meaning
J2ME™	Java™ Platform Micro Edition
Java EE	Java Platform, Enterprise Edition is a platform for server programming in the Java programming language
OTA	Over The Air
API	Application Program Interface
GPRS	General Packet Radio Service
GUI	Graphical User Interface
Bluetooth	Open wireless technology standard for exchanging data over short distances
SWOT	Acronym of Strengths, Weaknesses, Opportunities, Threats
VTL	Virtual Trip Line
DTL	Driver Trip Line
MIDlets	Java application written for J2ME / MIDP platform
IDE	Integrated Development Environment
PostgreSQL	Object-relational database management system
PostGIS	Open source software program that adds support for geographic objects to the PostgreSQL
GlassFish	Open source application server project led by Sun

	Microsystems for the Java EE platform
GPS	Global Positioning System
WLAN	Wireless local network
Wi-Fi	Networking technology for access to WLAN
SDK	Software development kit
JRE	Java Runtime Enviroment
RMS	Record Management System
MIDP	Mobile Information Device Profile
CLDC	Connected Limited Device Configuration
CDC	Connected Device Configuration

Table of Contents

Abstract (Greek)	5
Glossary	7
Table of Contents	9
List of Figures	12
1. Traffic monitoring system-“The Concept”	13
1.1 Introduction	13
1.2 The concept of the virtual line and its implementation in the network.	14
1.3 Operation of application	15
1.4 How to provide service through the application to server	16
1.5 How to provide service through the server to the application	17
1.6 The practical application of the system and its contribution to society	17
1.7 Major elements taken into account during design and development of the system	18
2. Mobile development platforms	19
2.1 Introduction	19
2.1.1 Java ME	19
2.1.2 .NET Compact Framework	20
2.1.3 Adobe Flash Lite	22
2.1.4 Android	23
2.1.5 iOS SDK	24
2.2 Comparison of Mobile Application Platforms	25
2.3 Current Trends	27
2.4 Summary	28
3. System Architecture and Design	30
3.1 Introduction	30
3.2 System design	31
3.2.1 Presentation tier	31
3.2.2 Application server	33
3.2.3 Database Server	33
3.3 System implementation	34

3.3.1	Presentation tier	34
3.3.2	Application server	37
3.3.3	Database Server (Entity-Relationship diagram and database table description)	39
3.4	Issues arising from the implementation of the system.....	41
3.4.1	Virtual Trip Line concept	41
3.4.2	Creation of VTL and transfer of geographic information in the database... ..	42
3.4.3	Spatial results from sql query	43
3.4.4	Application core process, how to detect a trigger	43
3.4.5	How to set the direction	45
3.4.6	Need of clustering VTL in areas.	46
3.4.7	Display of the results of the information system	47
4.	Commercial Exploitation	49
4.1	Overview	49
4.2	Benefits	49
4.2.1	Originality compared with international benchmarks	49
4.2.2	Description of the maturity of innovation.....	50
4.2.3	Feasibility of practical application	50
4.2.4	Evaluation of commercial exploitation	50
4.2.5	Evaluation of the contribution to the extroversion of the country	50
4.2.6	Evaluation of the contribution to viable development.....	51
4.2.7	Evaluation of the contribution to the improvement of everyday life.....	51
4.3	SWOT Analysis	52
4.3.1	Strengths	52
4.3.2	Weaknesses	53
4.3.3	Opportunities	53
4.3.4	Threads	53
4.4	Summary.....	54
5.	Conclusion and recommendations for Further Work.....	55
5.1	Conclusions derived from the current thesis	55
5.2	Recommendations for Further Work	56

6. References	58
7. Appendix.....	60

List of Figures

Figure 1.1: Virtual Trip Lines (VTL)	14
Figure 1.2: Traffic Representation using colors	16
Figure 2.1: .J2ME Architecture	20
Figure 2.2: .Net Compact Framework Stack	21
Figure 2.3: Flash Lite Framework Stack.....	22
Figure 3.1: Three-tier Arcitecture	30
Figure 3.2: J2ME Application Activity Diagram.....	33
Figure 3.3: VTL Creation	42
Figure 3.4.:VTL Placement.....	44
Figure 3.6 Intersection point inside of VTL	44
Figure 3.5 Intersection point outside of VTL.....	44
Figure 3.7: VTL Table Structure	45
Figure 3.8: Geographical Axis System	46
Figure 3.9: Separation of Thessaloniki, Greece.	47
Figure 3.10: PNG Picture as a Google service reply	48

1. Traffic monitoring system-“The Concept”

1.1 Introduction

The continuing development of mobile telephony and its adoption by the public create the appropriate conditions and prospects for the development of services designed to improve everyday life. New trends in the field of technology lead to the increasing production of devices that provide geographic data services (location based services) and Internet connectivity. The use of such data can help to estimate real-time traffic situation on the road network to more direct and reliable information of the moving driver.

On a mobile phone connected to a Global Positioning System (GPS) receiver, we can receive data concerning our motion on the road such as speed, latitude, longitude, etc. This data information can then be sent to a server via Internet connection over General Packet Radio Service (GPRS) or Wireless Fidelity (Wi-Fi) who will undertake to process it.

Currently, the existing infrastructure supports the recording of the traffic situation using static measurement points (i.e. cameras, sensors) provide that the appropriate equipment is installed. However, this infrastructure has several disadvantages. Due to the high installation and maintenance costs, it is not possible to cover the whole road network or a large part of it. Additionally, the static meters disable the dynamic management depending on the conditions formed on the road network. Finally, security issues of personal data for the citizens emerge since what is essentially recorded is the location of the citizen and many other personal information as it is recorded by means of multimedia (video, image).

The aim of this thesis is to create a system that assesses and calculates the traffic condition of the road network in real time, increasing the percentage of coverage of the network and the accuracy of results while giving great emphasis on security and privacy of the citizen-user system.

In order to properly implement the system the concept of the virtual line is employed. This line basically consists of a sequence of data in which the location and speed of the driver in motion are recorded.

The system is concerned about the design and development of an application for mobile phones with Internet connectivity and the support of geographic data services (location based services). Final product of the system is an application that provides the situation of traffic on roads in real time to the moving driver.

The novelty of the idea is the most feasible implementation because of the low economic cost (since everyone has a mobile phone), the protection of driver's personal data (since the structure of the system ensures it) as well as the provision of information in real time.

1.2 The concept of the virtual line and its implementation in the network.

The virtual line is essentially a line on a road that is part of the network involved in the calculation of the motion (Figure 1.1) [16]. When the user passes this point, the application sends the speed and position of the user to the server.

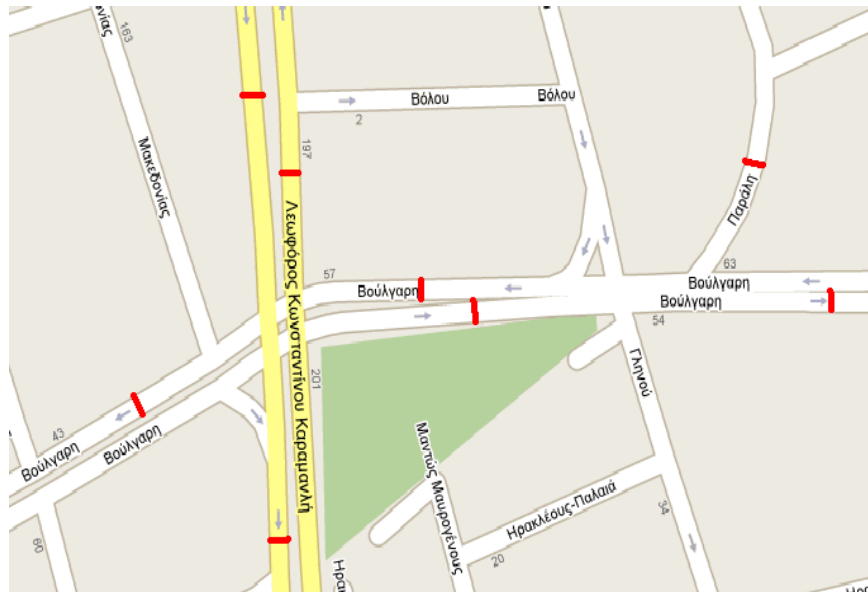


Figure 1.1: Virtual Trip Lines (VTL)

This implementation has the following advantages:

Data of space rather than of time. Basically, the data we collect to calculate the traffic is data of space and not of time. The data is concerned with some specific points which have been defined and are more of interest than others. These points can be changed dynamically according to the requirements.

Personal data Protection. These points are stored on the device of the user and automatically updated from the Internet. The selection of these, during the update, is random among a whole, ensuring the non-continuous (spatial) record of the path followed by the driver.

Coverage of a large part of the network. With the proper placement of the virtual lines on the road makes possible the reception of information about almost the whole road network.

Dynamic entrustment of points. Through the dynamic nature of this implementation, it is possible to change the location of the collection of data according to the needs created (e.g. insufficient information on some points).

1.3 Operation of application

The operation of this application is divided into two different levels:

- The application is involved in the update about the traffic situation on the streets by sending information to a remote server who takes on to assess, applying appropriate algorithms, and the traffic on the roads. This information, obtained through GPS, are related to the location of the device and the speed of the driver, while ensuring the security of personal user data.
 - In this mode the user is not actively involved, they are simply informed and accept how they will participate, at the starting of the application.
- The application receives data for traffic and depicts it in embedded map by using appropriate colors to represent the movement (Figure 1.1).

- The user can see the traffic on all available roads, and can choose himself/herself the route suitable for him/her.

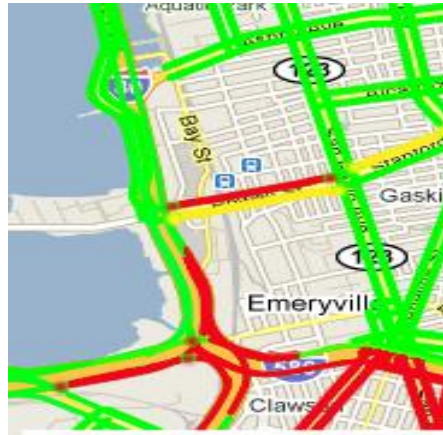


Figure 1.2: Traffic Representation using colors

1.4 How to provide service through the application to server

The application, having top priority to ensure the privacy of the user (path followed, etc.) along with the minimization of the cost of using mobile internet services (reducing the volume of data exchanged) works as follows:

- The application receives from the GPS receiver the position of the moving driver at regular intervals. A check is performed whether this point belongs to a virtual line or not and the line participates in the network for the calculation of the traffic situation. These lines are defined based on criteria that serve the procedures for the assessment of motion. If this check is true then the application sends the necessary information (position, speed, direction) to the server. Thus, there is no continuous recording of the driver's location (personal data protection) and the cost of using the service is minimized due to the non-continuous transmission of data.
- When the server receives, the information it takes on dividing the data related to user authentication and those relating to information on the traffic situation. The first data are processed only for statistical study on the performance of the system depending on the number of users that have enabled

the application and certification purposes. The geographic data along with speed are diverted to another server that is responsible for applying the algorithms that estimate traffic conditions.

Due to above techniques of non-"horizontal" representation of the location and identity of the user, security and privacy are ensured to all participating users.

1.5 How to provide service through the server to the application

The server that provides the calculations for the state of roads undertakes to send information to users who have activated the application. The data is obtained by the application and is represented on the embedded map.

Moreover there is another service that is responsible for sending updates on the various areas where the collection of data from users will be done. These areas may be changed either dynamically (eg after updating for false information depending on the needs), or statically after specific transportation analysis / calculations.

1.6 The practical application of the system and its contribution to society

One of the main characteristics of the proposed system is the creation of an informatory system for traffic situation using the users and therefore their mobile phones, as meters.

So with the minimum cost (everyone has a mobile phone, and increasingly with integrated GPS receiver) we have a large number of dynamic meters involved in the process of gathering information on the export of results (Zero-infrastructure systems), as opposed to static meters (sensors, etc.) that and can be installed on roads where the cost of installation and maintenance is much higher. In addition to, the continuous development of telecommunications renders the conditions in market appropriate for using the Internet and its services on mobile devices at low cost.

The bigger the number of users is, the more secure and reliable information on traffic conditions we will have (in the concept of cooperative mobility systems - Cooperative

mobility systems). Finally, the update will be made directly to the device and through the application, providing real time information (real-time) at every point of the route.

1.7 Major elements taken into account during design and development of the system

During the design and development of the system several issues were taken into account, such as the privacy of the user, the minimization of the cost as well as the issues concerning the best use of computing power of the mobile phone device. Below is a general description of those issues.

1. User's Personal Data Protection
 - Structure of data that is sent from the application to the server, separation of user authentication information from the data used for calculations
 - Transmission of data from the application, by dynamically points randomly selected from a range of areas.
2. Minimisation of cost of the dispatched / received information when using mobile internet for data communication
 - Discontinuous transmission of information from application to server.
 - Codification, classification of areas in an appropriate way to reduce the amount of information.
3. Minimization of required computational power of mobile telephony devices.
 - Development of techniques for the least possible algorithmic requirements, that the application is consistent with the highest percentage of mobile telephony devices.

2. Mobile development platforms

2.1 Introduction

Mobile devices have been gaining acceptance because of their physical characteristics (mobility, location-awareness) and services offered by their manufacturers. Current application programming platforms and development tools provide developers to use various technologies like, Open C, Python, Flash Lite or Web technologies (such as XHTML/CSS, JavaScript, Mobile Ajax and mobile widgets) to create highly functional mobile applications. This section briefly discusses the most important characteristics of runtime environments for handheld devices (mainly mobile phones and PDAs), which currently enjoy the largest developers and deployment base: Java ME, .NET Compact Framework, Flash Lite and Android, iOS SDK.

2.1.1 Java ME

J2ME, also known as Java ME, is a platform that is specifically designed for development of applications that run on mobile phones, PDAs and other embedded systems. J2ME development features an adaptable user interface, which integrates with network protocols and extends support for mobile application development. The mobile applications that are developed on J2ME can be also migrated across various different mobile devices. J2ME application developers create a variety of different mobile applications based on the customized requirements of the clients [8].

J2ME comprises of configurations and profiles that allow a developer to customize it for the Java Runtime Environment (JRE). The configuration defines the JVM used and the profile adds domain-specific classes to define the application.

- **Configurations.** The configuration uses a set of core classes and a specific JVM to define the basic run-time environment. The configurations are of two types. The first is called CLDC for handheld devices and the second is CDC for plug-in devices. CLDC has been developed for 16-bit or 32-bit small computing devices that have

limited memory. CDC requires a 32-bit architecture and has at least 2 MB of memory and implements a functional JVM.

Profiles. A profile comprises of classes that allow J2ME application developers to execute features that are typically available on a group of small computing devices. The profiles used with CLDC are Mobile Information Device Profile (MIDP) and PDA profile (PDAP). The profiles used with CDC include Foundation Profile, Game Profile, Personal Profile, Personal Basis Profile and RMI Profile.

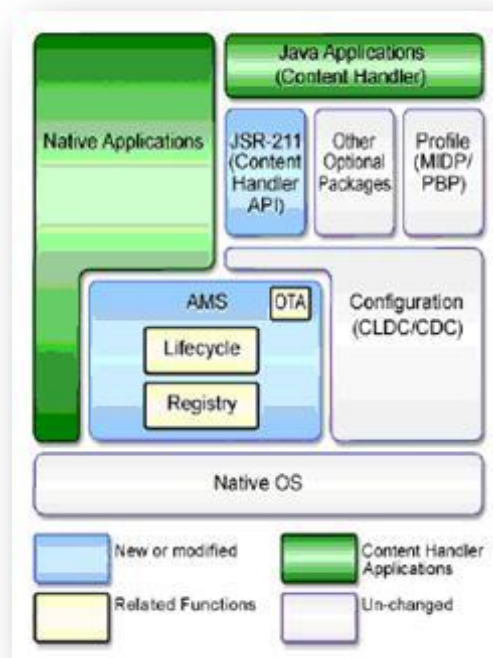


Figure 2.1: J2ME Architecture

2.1.2 .NET Compact Framework

The Microsoft .NET Compact Framework (.NET CF) [9] is a version of the .NET Framework that is designed to run on resource constrained mobile/embedded devices such as personal digital assistants (PDAs), mobile phones, factory controllers, set-top boxes, etc. The .NET Compact Framework uses some of the same class libraries as the full .NET Framework and also a few libraries designed specifically for mobile devices such as Windows CE Input Panel. However, the libraries are not exact copies of the

.NET Framework. The ones in the .NET Compact Framework are scaled down to use less space. To be able to run applications powered by the .NET Compact Framework, the platform must support the Microsoft .NET Compact Framework runtime (Figure 2.2).

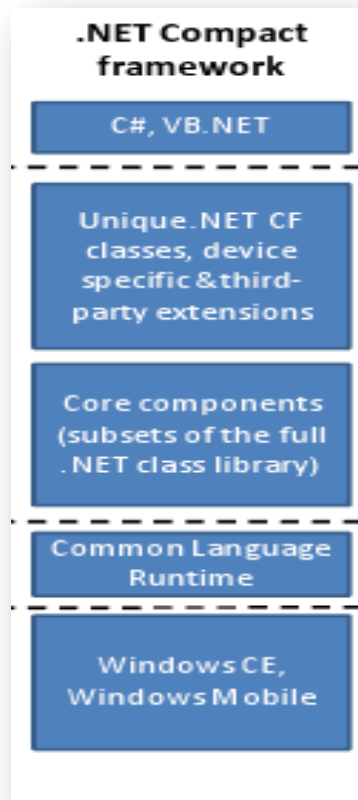


Figure 2.2: .Net Compact Framework Stack

Some operating systems that include .NET CF are Windows CE 4.1, Microsoft Pocket PC, Microsoft Pocket PC 2002, Smartphone 2003, and Symbian v3. The .NET Compact Framework applications can be run on desktop computers with the full .NET Framework as long as they only access the shared parts of both frameworks, though their user interface cannot be upgraded to look like that of an application developed for desktop PCs.

2.1.3 Adobe Flash Lite

Adobe® Flash® Lite™ software is a powerful runtime engine for mobile and consumer electronics devices that benefits both device manufacturers and content developers. By enabling manufacturers to provide customers with improved web browsing, video content, and compelling interactive experiences. Flash Lite 3 allows developers to quickly create engaging mobile applications, speed time to market, and increase customer adoption.

Currently, there is a quickly growing adoption of Flash Lite by Original Equipment Manufacturers (OEMs), operators and developers worldwide. Flash Lite 1.1 supports Flash 4 ActionScript 1.0. Flash Lite 2.0 and 3.1, based on Flash Player 7 and 9 respectively; support ActionScript 2.0 (no support provided for Flash Player 9 compatible content based on ActionScript 3.0). All versions support the W3C standard Scalable Vector Graphics (SVG) Tiny (a mobile profile of W3C's SVG recommendation). Figure 2.3, represents Flash Lite software stack.



Figure 2.3: Flash Lite Framework Stack

2.1.4 Android

Google's Android has been launched in 2007, devoted to advancing open standards for mobile devices. Android is an Apache free-software platform with open source license for mobile devices based on Linux. It essentially comprises a software stack for mobile devices that includes an OS, middleware and key applications (Figure 2.4).

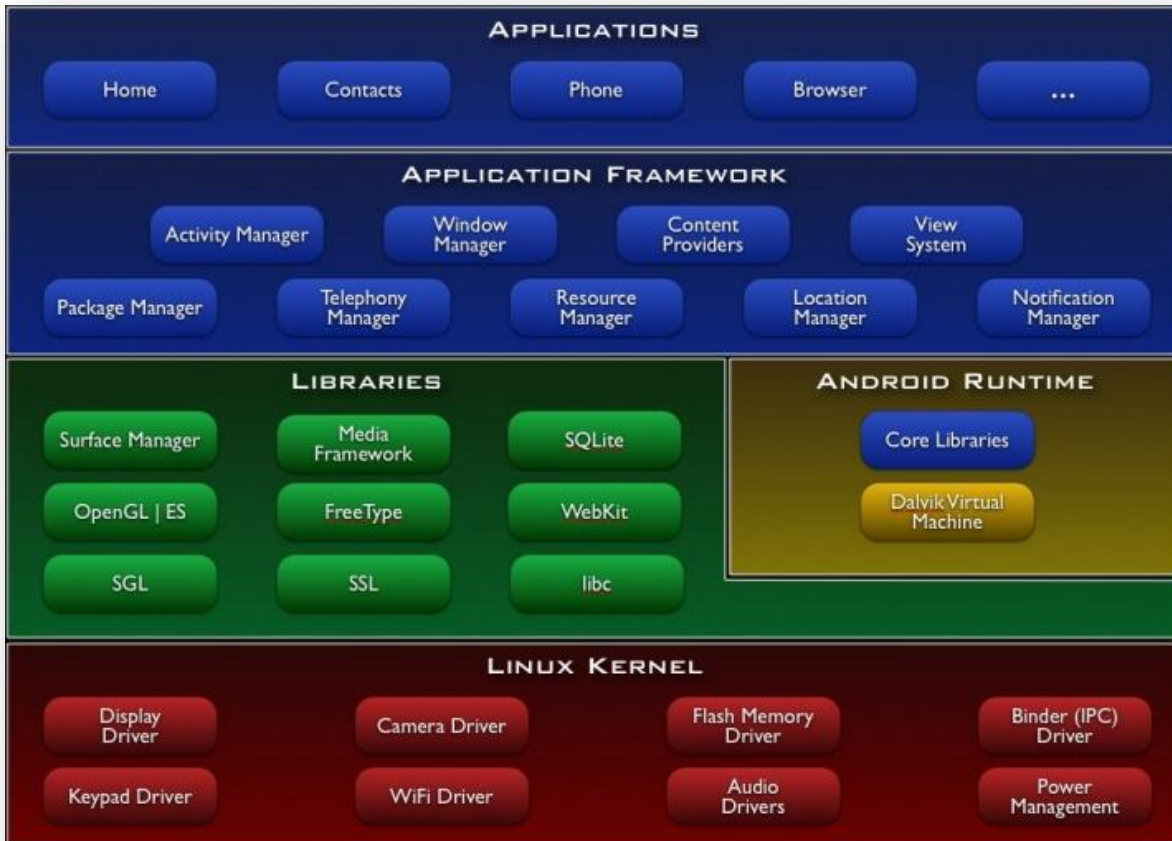


Figure 2.4: Android Framework Overview

The Android software development kit (SDK) includes a comprehensive set of development tools [10]. These tools include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.4.9 or later, Windows XP or later. The officially supported Integrated Development Environment (IDE) is Eclipse (currently 3.4, 3.5 or 3.6) using the Android Development Tools (ADT) Plugin. However, developers may use any text

editor to edit Java and XML files. They then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely) [11].

2.1.5 iOS SDK

iOS (known as iPhone OS prior to June 2010) is Apple's mobile operating system.[7] Originally developed for the iPhone, it has since been extended to support other Apple devices such as the iPod Touch, iPad and Apple TV. Apple does not license iOS for installation on third-party hardware. The user interface of iOS is based on the concept of direct manipulation, using multi-touch gestures. Interface control elements consist of sliders, switches, and buttons. The response to user input is immediate and provides a fluid interface. Interaction with the OS includes gestures such as swiping, tapping, pinching, and reverse pinching. Internal accelerometers are used by some applications to respond to shaking the device (one common result is the undo command) or rotating it in three dimensions (one common result is switching from portrait to landscape mode).

iOS is derived from Mac OS X, with which it shares the Darwin foundation, and is therefore a Unix-like operating system by nature. In iOS, there are four abstraction layers: the Core OS layer, the Core Services layer, the Media layer, and the Cocoa Touch layer. The operating system uses roughly 500 megabytes of the device's storage, varying for each model.

The SDK [6] was released on March 6, 2008, and allows developers to develop applications for the iPhone and iPod Touch, as well as test them utilizing a "iPhone simulator". However, loading an application into the devices is only possible after paying an iPhone Developer Program fee. Since the release of Xcode 3.1, Xcode is the development environment for the iOS SDK. iPhone applications, like iOS and Mac OS X, are written in Objective-C.

Developers are able to set any price above a set minimum for their applications to be distributed through the App Store, of which they will receive a 70% share. Alternately,

they may opt to release the application for free and need not pay any costs to release or distribute the application except for the membership fee.

2.2 Comparison of Mobile Application Platforms

In this subsection we provide the comparison of different tools and devices for each of the platforms to make the best of your mobile applications [15].

Java ME is the dominant mobile software platform with respect to its installation and developers base. However, the 'write once, run anywhere' axiom of the Java language does not apply to the world of Java ME Device capabilities such as screen size, color depth, keyboard values and memory vary considerably. Given the wide range of Java ME-compatible phones along with the choice of profiles, configurations and Java ME APIs (specified in JSRs), developers are required to provide slightly different application versions (not all phones support the same set of JSRs, while JSR implementations vary across devices), which often results in dozens of executables for a given title. This is commonly referred to as device fragmentation, which considerably increases operational costs as it affects the development-testing-signing-delivery-maintenance cycle. Fragmentation restricts Java applications device reachability and suggests Java ME only for vertical applications, i.e. for limited targeted devices with similar capabilities and Java APIs support.

By targeting individual OS platforms, mobile application developers may use a large set of well-defined and mature JSRs (e.g. Java applications targeting the Symbian platform can reach ~70% of the world's smartphones). There are currently more than 80 JSRs providing the MIDlet developers with a rich and diverse set of additional technologies; however, MIDlets programming is not straightforward as it requires serious Java development skills. Examples of commonly available JSRs that extend MIDP 2.0 on the Symbian platform include the Bluetooth API (JSR 82), the Wireless Messaging API (JSR 205) and the Mobile 3D Graphics API (JSR 184).

.NET CF is comparable to Java ME with respect to providing: (a) a managed runtime environment to manage security, memory usage and runtime optimization, (b) rich

libraries and components to allow developers to reuse software modules (advanced UI components, network connectivity, data management, XML web services, etc), and (c) familiar APIs from the full .NET framework (e.g. the Windows Forms controls) which allow as smooth transition of desktop developers to mobile development.

The use of a runtime system for intermediate (managed) code (similarly to Java ME) implies relatively low execution performance. Unlike Java though, it is designed to be language-agnostic and simply specifies the Common Intermediate Language (CIL) instructions. Therefore the .NET supported languages (i.e. C# and VB.NET) compile to the same CIL and can be executed by the .NET CF runtime.

Flash Lite platform is a reasonable choice for graphics-intensive phone and PDA applications. It has enjoyed increasing adoption by mobile technologies industry players, since the developers already skilled in working with Flash for desktop applications can easily switch into using Flash Lite to design mobile applications. The main assets of Flash Lite include rapid development (easy to learn, fast migration from Flash applications, rich designer/developer tools), rich media support (images, video, sound, animation), a relatively broad runtime installation base and small deployment files (more animation and graphics packed into the same file size when using SVG). As of Flash Lite 2.x compressed SWF may be used, while Flash Lite 3.0 added support for the popular native Flash video (flv). Currently, Flash Lite is mostly suitable for creating animations, casual games, mobile web-based Flash applications, front-end interfaces, device-specific content (e.g. wallpapers, screen savers, etc), however not for developing full-fledged standalone applications mainly due to the lack of powerful mobiles-oriented APIs (e.g. compared to the Java ME platform).

On the other hand, Flash Lite exhibits relatively poor graphics performance (partially due to the complex processing required for vector graphics), while its extensive toolset (Adobe CS4, Adobe Device Central) is shipped with a license fee. Its low-level device integration can be perceived as limited; 'outside the box' low-level device APIs are offered by third parties (e.g. Kunerilite toolkit extends Flash Lite capabilities in the Symbian platform) allowing the development of powerful, innovative applications at the expense of raising fragmentation issues and increasing memory footprint.

Android supports a relatively large subset of the JSE 5.0 library (implying reduced migration cost from Java desktop applications) along with several third-party libraries; similarly to Java ME, application development is powered by popular Java IDEs (e.g. NetBeans, Eclipse). An important asset of Android is its inherent support for modular service-oriented applications and inter-application communication (MIDP 3.0 of Java ME also supports inter-MIDlet communication).

2.3 Current Trends

This section presents the trends in the development platforms of mobile applications with the latest conditions in the market as follows [12] [13]:

Micropayments: Micropayments can be defined as mobile transactions that involve a small sum of money. Micropayments within mobile applications can be used to upgrade basic app to a premium version, purchase game items, digital content or even small gifts for friends.

Enhanced Security: Better security for mobile application platforms is expected. This is especially important when more users are conducting financial transactions and life streaming using their phones.

Business App Store: With all the mobile platforms targeting their application store towards average consumers, the introduction of a business app store is imminent.

Location-Based Technology: Location-based technology or GPS technology received the most buzz in 2009 and we expect it will continue growing this year.

Social Based Applications: Social networking activities certainly do not end when you leave your computer. We are already seeing a great number of people tweeting and updating their Facebook status on-the-go, not to mention those who are posting videos and photos to services like Twitpic and 12seconds using their mobile phone.

Augmented reality(AR): Augmented reality (AR) [14] is a term for a live direct or indirect view of a physical real-world environment whose elements are merged with (or *augmented* by) virtual computer-generated imagery – creating a mixed reality.

High Entry Barriers for Fledgling Developers: Developers that are new to the scene and without much support will face trouble getting user adoption. Not only that there are already tons of mobile applications out there, established developers have better advantages in terms of capability to introduce new features over a short time span as well as the resources to adapt and test their applications on new platforms.

Mobile Application Advertising: The increasing number of mobile application users opens up another advertising channel for brands and businesses. This is definitely good news for developers.

Importance of Marketing for Applications: There are already hundreds of thousands of mobile applications out there. In order to stand a chance, developers or mobile application entrepreneurs need to know how to market applications developed by them.

HTML 5 – Native Apps vs Web Apps: Phones with a mobile browser that has HTML 5 rendering capabilities will be able to run web applications directly without any downloading and installation of apps. As for developers, there will no longer be a need to develop for different mobile platforms.

2.4 Summary

Due to the increasing use of mobile devices by users for everyday communications, for recreational purposes and for business purposes software companies promote programming platforms and development tools to service programmers and software designers. Below, we assess the appropriateness of each of the reviewed platforms with respect to some critical application development requirements [15].

Portability: Portability primarily depends on (a) runtime support, and (b) the feasibility of achieving identical cross-platform look-and-feel and functionality. In terms of runtime support, Java ME is undoubtedly the winner, followed by Flash Lite. Android is likely to extend its deployment base and .NET CF will probably remain a Windows-only platform. On the other hand, Java ME is fragmented with respect to developing cross-platform applications. Flash Lite is more suited to cross-platform development due to the strict control of Adobe over its runtime environment

Functionality: The aim of implementing multimedia-rich full-fledged applications is better served through Java ME due to the numerous APIs (JSRs) implemented by OEMs to exploit handsets capabilities. The .NET CF and Android applications also utilize rather powerful APIs. Flash Lite mostly suits graphics-heavy applications.

Development speed: The safest way to guarantee steep learning curve and higher development speed, is to take advantage of the developers' programming experience on the implementation of desktop applications. For instance, Java developers will naturally fit into Java ME and Android, Flash developers into Flash Lite and so on. Developers not familiarized with any of the platforms' foundation languages will feel more comfortable and productive with Flash Lite's ActionScript. Nevertheless, the development process in 'traditional' platforms such as Java ME and .NET CF is accelerated due to adequate documentation and extended developers community base.

Performance: Mobile applications are becoming increasingly computationally heavy, thereby raising requirements for increased runtime speed and storage. Performance metrics like processing overhead, memory consumption, frame rate and deployment file size also depend on the particular development platform toolset, e.g. support for SVG Tiny, graphics buffering, support for compressed sound files, etc.

3. System Architecture and Design

3.1 Introduction

System design and development is based on 3-tier architecture (Figure 3.1). Three-tier is a client-server architecture in which the user interface, functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms. The three-tier model is a software architecture and a software design pattern.

Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the presentation tier would only affect the user interface code.

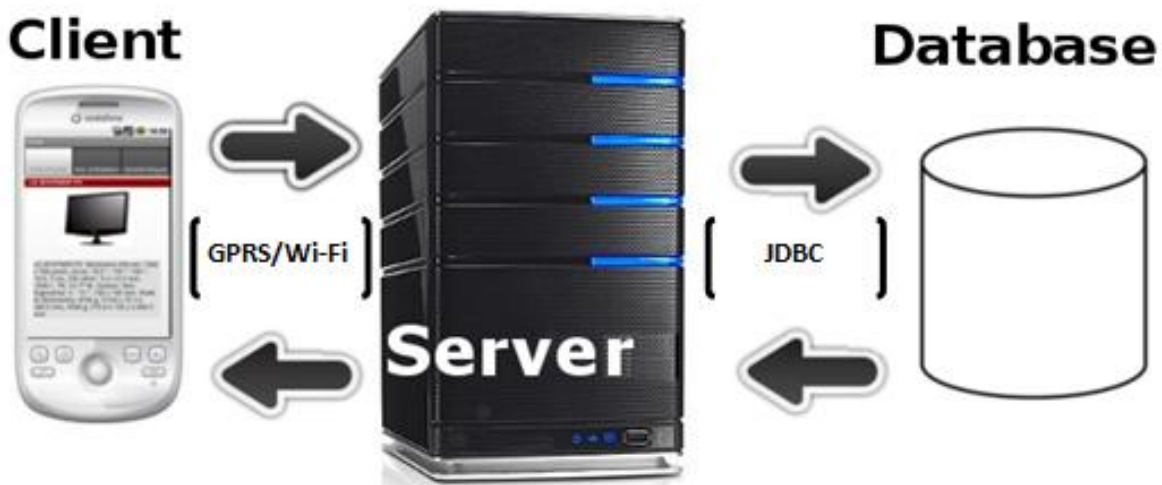


Figure 3.1: Three-tier Architecture

Typically, the user interface runs on a desktop PC, workstation or on a mobile device and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the computer data storage logic.

The developed system has the following three tiers:

— Presentation tier

The mobile device application is part of this level. The functions of this application are connecting to the device's GPS, the storage of the latest data of the system, the sending / receiving data to / from the application server and then the depiction of the traffic conditions in real time on the user's screen.

— Application Server

The server of which this level essentially consists is responsible for communicating the application (presentation tier) to the database system. This level includes functions crucial for the proper functioning of the system, such as user authentication, verification of data and mediation for sending and receiving data from the database to the application.

— Data tier

This level refers to the database responsible for storing data necessary for the use and operation of the system and data obtained from it. The database supports geographic data needed to develop the system described.

3.2 System design

3.2.1 Presentation tier

The functions at this level have to do with the operation of the application and its communication with the application server. These functions are the following: user authentication, database creation / load, connection to GPS, vtl data update, application core process, traffic presentation.

User authentication: The first form that appears when the user starts the application is the user authentication. The user fills in username and moves to the next form. First of all, a certification of user identification is conducted and then data is sent to inform the system that it has added another user in the system

Database creation/load: When first using the application on each device an internal database, for storing data relating to the points where the speed of the moving driver is recorded, is created (VTL Data Table). In the next use the base already exists and simply a link to it is made. This database allows the storage of data used by the application and can be removed during the uninstall process of the application from the user device.

Connection to GPS: The application undertakes to connect to GPS via Bluetooth. The user is simply required to choose which location based services device wants to connect to.

Vtl Data Update: The application through the connection to the GPS device acquires data relating to geographic location (longitude, latitude) and sends a query to the server to verify that the information recorded in the internal database is the latest available. If new updates are available it undertakes to renew them.

Application core process: The application periodically checks whether the point at which you are currently is a point that participates in the assessment of the traffic status of the system. If so it shall send to the server data related to geographic position and speed of the movement.

Traffic Presentation: The application periodically contacts the server to obtain information about the traffic on the roads and in turn undertakes to display it to the user.

Below (Figure 3.2) is the **activity diagram** of the client application

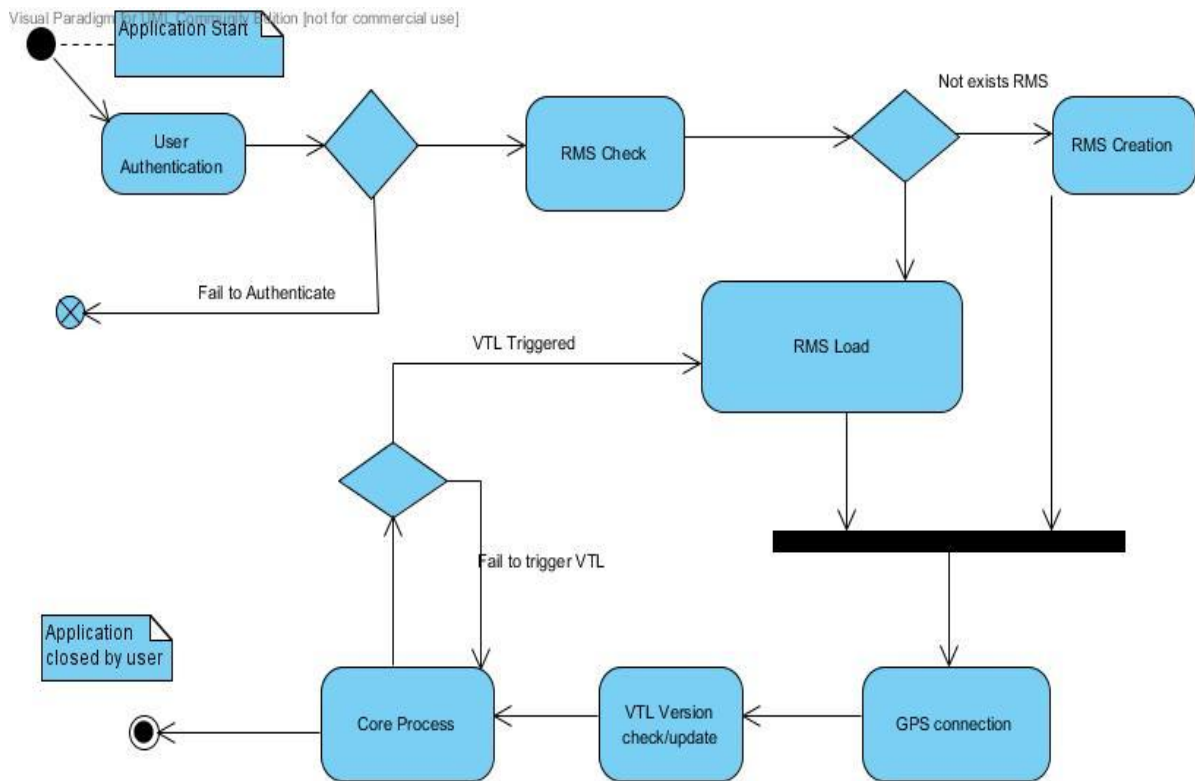


Figure 3.2: J2ME Application Activity Diagram

3.2.2 Application server

The server responsible for communication, exchange of data between application and database belongs to this level. The functions performed at this level support functions of the presentation tier, such as user authentication, VTL data update, Traffic Presentation.

3.2.3 Database Server

The database server is responsible for storing data which originally was created to support the system and which arises during the operation. The database server via the application server provides data to the application of the user's mobile phone.

Operations performed on the database server have to do with calculations about the location of the driver, and especially what area (VTL Area) they belong to.

3.3 System implementation

The system implementation was done using tools and platforms that are freely available on the market within the open source development. The application for mobile devices developed in the programming language JAVA (Mobile Edition) with NetVeans IDE (Integrated Development Enviroment). The application server has been created using the Glassfish Server 3 with Java EE support. Postgres plus Advanced Server v8.4 is used as database server. Advanced edition of postgres serve supports Geospatial (PostGIS) feature.

3.3.1 Presentation tier

Data Storage: Mobile application use Record Management System (RMS) provided by java for persistent storage. When first run the application on any device the creation of the RMS takes place; it is necessary to store data. This data refers to the points where the system automatically sends information to the server and the number of the data publication. This number uniquely identifies with all of those signs that it may be possible to check for any new updates that arise during system operation.

Sample code:

```
try
{
    RecordStore recordStore = RecordStore.openRecordStore("VtlData", true,
        RecordStore.AUTHMODE_ANY, true);
}
catch (RecordStoreException rse)
{
    rse.printStackTrace();
}
```

Connection to GPS: The application gives the possibility to use an external GPS using Bluetooth, for devices that have integrated GPS receiver, or for users who want to use Bluetooth GPS. For the usage of an external GPS the JSR 82 Bluetooth Java API is used.

Sample code:

```
BTUThread=new Thread();

// retrieve the Bluetooth address of the local device
LocalDevice localDevice = LocalDevice.getLocalDevice();

try
{
localDevice.setDiscoverable(DiscoveryAgent.GIAC);
}
catch(Exception e)
{}
discoveryAgent = localDevice.getDiscoveryAgent();
```

Moreover, the application supports connection to GPS (internal, external) with the use of Location Based API (JSR-179) supported by more and more devices. Through the use of the specific library, connection with both external and incorporated GPS receiver is possible.

Sample code:

```
try
{
Criteria cr = new Criteria();
cr.setHorizontalAccuracy(500);

LocationProvider lp = LocationProvider.getInstance(cr);
Location l = lp.getLocation(60);//60 seconds, timeout

Coordinates c = l.getQualifiedCoordinates();
String lat = c.getLatitude();
}
```

Web connection: The application for communicating with the server to send and receive data uses either the General Packet Radio Service (GPRS) or is connected to the network through a wireless local area network (WLAN). The contact is made by using Hypertext Transfer Protocol (HTTP).

Sample code:

```
HttpConnection c = null;
InputStream is = null;
StringBuffer sb = new StringBuffer();
try {
    c = (HttpConnection)Connector.open(
        "http://160.40.60.86:12000/KonpachJSP/TrafficMap.jsp",
        Connector.READ_WRITE, true);
    c.setRequestMethod(HttpConnection.GET); //default
    is = c.openInputStream(); // transition to connected!

}
```

Traffic Presentation: The depiction of motion in the application is made by using Google static map services. The application asks the server for data about traffic on the roads and creates an appropriate URL that will result in correct display of traffic on the map.

Sample code:

```
http://maps.google.com/maps/api/staticmap?" + lat + lon + zoom + width + height
```

The operation of **Application core process** is described at the end of the chapter for better understanding of the process.

3.3.2 Application server

Java Enterprise edition Platform is used for server programming. Glassfish 3.1 is used as the application server. The connection of the database with the application of the user, for data transfer, is not possible without using application server services. The services were developed to support the functions of the mobile device and the communication with the database.

The functions supported by the application server are:

- **Vtl update process**

The application, at the beginning of its operation, sends to the application server the point where it is located, its position, and the publication of VTL Data Table. This publication is indicated by a serial number that changes with every change made to data in the VTL Data Table database. The application server checks whether this publication to the user corresponds to that of the base. If not, it bears to read the data from the database, converts it into XML format and sends it to the user's device as a response to the request of the application.

Http Response structure

```
<arealD>1</arealD>
<version>4</version>
<totalCount>1</totalCount>
<vtlTable>
  <vtl>
    <id>1</id>
    <sX>22.964508415660404</sX>
    <sY>40.59803216064448</sY>
    <eX>22.965057049908644</eX>
    <ey>40.59804841764106</ey>
    <latD>1</latD>
    <lonD>0</lonD>
  </vtl>
  <vtl>
    <id>2</id>
    <sX>22.964568415660404</sX>
    <sY>40.329803216064448</sY>
```

This structure essentially represents a line on the geographical axis system with Start Point coordinates sX, sY and End Point coordinates eX, eY. The id uniquely

identifies with each line. The attributes latD, lonD are used to define the direction we are interested in at this point. An explanation of these attributes and their usefulness follows in one of the next sub-chapters.

- **Update for traffic information to a specific VTL**

The application during its function realises that the user passed a specific point (trigger) and undertakes to send data about the state of motion (speed of the driver) at this point. The application server bears to serve the request of the application for traffic update at the specific vtl and promotes the information (vtl id, speed) to the database server. Upon completion of this operation the execution of the operation of Vtl **update process** is carried out as well, in order to ensure that the application of the user stays updated throughout the duration of use.

- **Traffic Estimation Results**

The data obtained by each user is used in the process of calculating the traffic situation on the road network. This data through the service of this application is transferred into the application of the mobile phone which then bears to portray it. This data structure consists of points that portray a road and the property that characterizes the state of this road. This property takes the values 1,2,3 describing the movement as free, dense and saturated, respectively.

- **Separation of user authentication data and her/his geographic position**

In any communication between the application and the application server the user authentication is checked. This information is separated from information on the driver's position to the network in order to secure the privacy of the user.

3.3.3 Database Server (Entity-Relationship diagram and database table description)

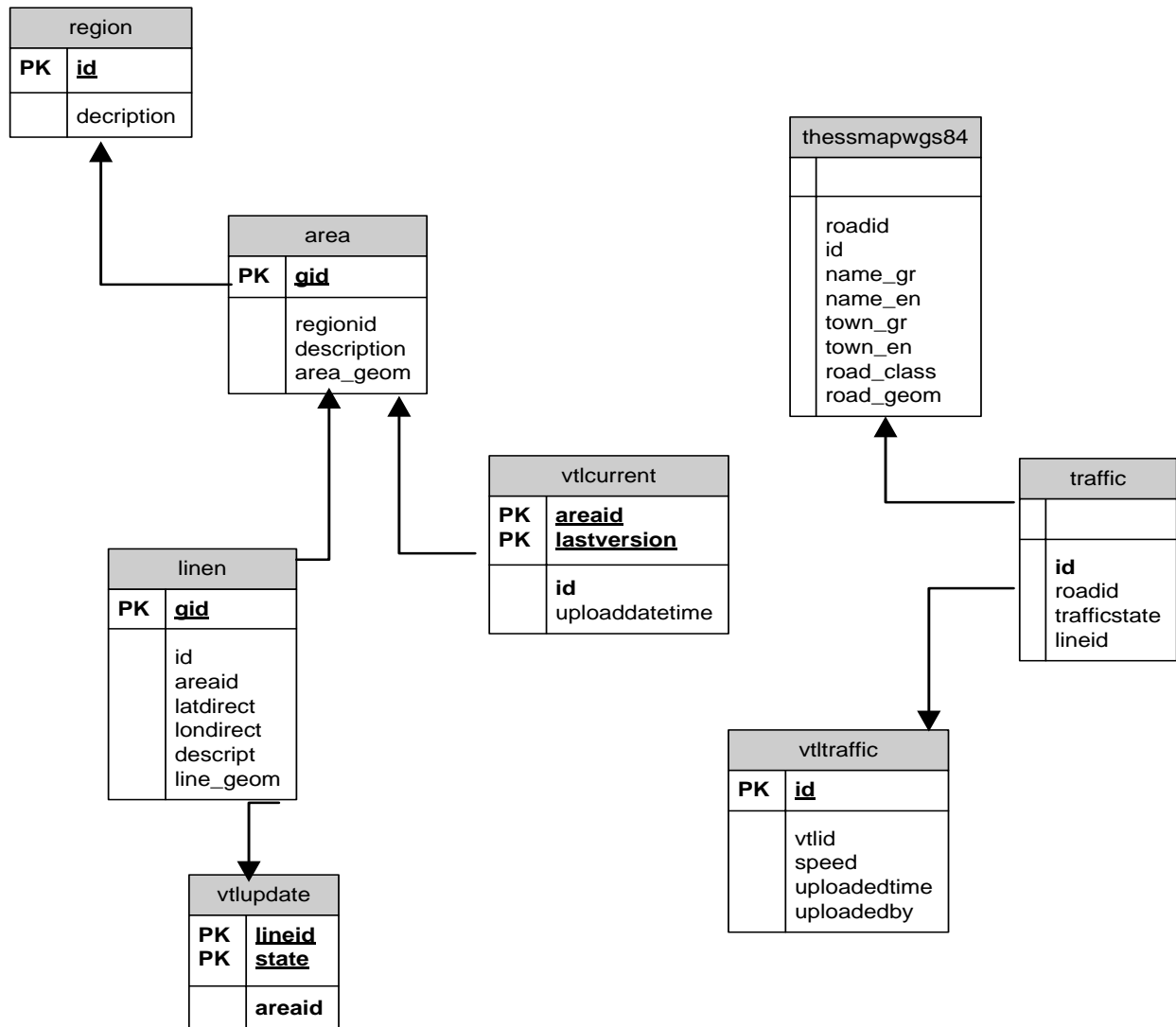


Table: Thessmapwgs84

Description: It contains the geographic information of road network in the city of Thessaloniki. Column **road_geom** contains the geographic information of each record and it is a **Geometry** type.

roadid	id	name_gr	name_en	town_gr	town_en	road_class	road_geom
integer	integer	character varying	character varying	character varying	character varying	integer	geometry
109	5156	ΕΓΝΑΤΙΑ ΟΔΟΣ - ΠΑΡΑΠΛΗΕ	EGNATIA ODOS - SID	Θεσσαλονίκη	Thessaloniki	3	0102000020E61000000200000043A4630F
170	5157	ΣΟΦΟΥΛΗ ΘΕΜ.	SOFULI THEM.	Θεσσαλονίκη	Thessaloniki	3	0102000020E610000002000000C2C08C11
171	5158	ΕΓΝΑΤΙΑ ΟΔΟΣ - ΠΑΡΑΠΛΗΕ	EGNATIA ODOS - SID	Θεσσαλονίκη	Thessaloniki	3	0102000020E6100000040000005362665F

Using **PostGIS** spatial database extension for **PostgreSQL** we run the query below:

SQL Query:

```
SELECT ST_AsText(road_geom)
FROM thessmapwgs84
where id=204486
```

Data Output:

```
"LINESTRING (22.8550273507724 40.6937045330004,22.8536431387669
40.6947828644219)"
```

The geographical information that contains the **road_geom** consists of a sequence of points in the geographical axis system.

Table: Region

Description: This table contains information relating to the areas- prefectures involved and supported by the system.

	id [PK] serial	decription text
1	1	Thessaloniki
2	2	Chalkidiki
*	<input type="text"/>	

Table: Area

Description: It consists of data relating to areas- zones in which each **region** is divided.

	gid [PK] serial	regionid smallint	description character vai	area_geom geometry
1	1	1	Anatolika	0 1060000 20E6 1000000 10000000 1030000000 100
2	2	1	Kentro	0 1060000 20E6 1000000 10000000 1030000000 100
3	3	1	Dutika	0 1060000 20E6 1000000 10000000 1030000000 100
*	<input type="text"/>			

Table: linen

Description: This table contains the Virtual Trip Lines. Later in the chapter a description of this very structure is provided.

	gid [PK] serial	id integer	areaid smallint	latdirect smallint	londirect smallint	descript character	line_geom geometry
1	1	0	1	-1	0		0 105000020E6 1000000 10000000 10200000
2	2	1	1	1	0		0 105000020E6 1000000 10000000 10200000
3	7	7	1	0	-1		0 105000020E6 1000000 10000000 10200000
4	10	10	1	1	0		0 105000020E6 1000000 10000000 10200000

Table: Vtlcurrent

Description: This table contains the serial number of the publication of the latest available update on the VTL for each **area** involved in the system.

	id serial	areaid [PK] integer	lastversion [PK] bigint	uploaddate timestamp w
1	1	1	4	
2	2	2	2	
3	3	3	3	

Table: Vtltraffic

Description: This table contains the information received by the system from the user of the application. This information includes data on the VTL and the speed with which the user has passed at that moment.

	id [PK] integer	vtlid integer	speed text	uploadedtime timestamp with time zone
1	26	1	4.3	2010-11-04 10:47:07.734+02
2	27	2	4.4	2010-11-04 10:49:26.781+02
3	28	2	4.4	2010-11-04 10:49:39.609+02
4	30	2	4.4	2010-11-04 10:50:42.625+02
5	31	1	5.14	2010-11-04 11:20:25.843+02
6	32	2	61.70	2010-11-04 11:23:59+02

Finally, the tables **traffic**, **useractions**, **category**, **vtlupdate**, **category** assist in the support and function of the system.

3.4 Issues arising from the implementation of the system

3.4.1 Virtual Trip Line concept

The proposed traffic monitoring system builds on the novel concept of virtual trip lines. A virtual trip line (VTL) is a line in geographic space that, when crossed, triggers a client's location update to the traffic monitoring server. More specifically, it is defined by [id; x1; y1; x2; y2; d] where id, is the trip line ID, x1, y1, x2, and y2 are the (x; y) coordinates of two line endpoints, and d is a default direction vector (e.g., N-S or E-W). When a vehicle

traverses the trip line its location update comprises time, trip line ID, speed, and the direction of crossing.

The trip lines are pre-generated and stored in clients. Virtual trip lines control disclosure of location updates by sampling in space rather than sampling in time, since clients generate updates at predefined geographic locations, compared to sending updates at periodic time intervals. The rationale for this approach is that in certain locations traffic information is more valuable and certain locations are more privacy-sensitive than others. Through careful placement of trip lines the system can thus better manage data quality and privacy than through a uniform sampling interval. In addition, the ability to store trip lines on the clients can reduce the dependency on trustworthy infrastructure for coordination.

3.4.2 Creation of VTL and transfer of geographic information in the database

The design of the Virtual Trip Line (VTL) (Figure 3.3) has been using geographical information system (GIS). The GIS used is the ArcGIS of ESRI. Taking as a base a layer of roads of Thessaloniki, ArcMap was used for the design of VTL.

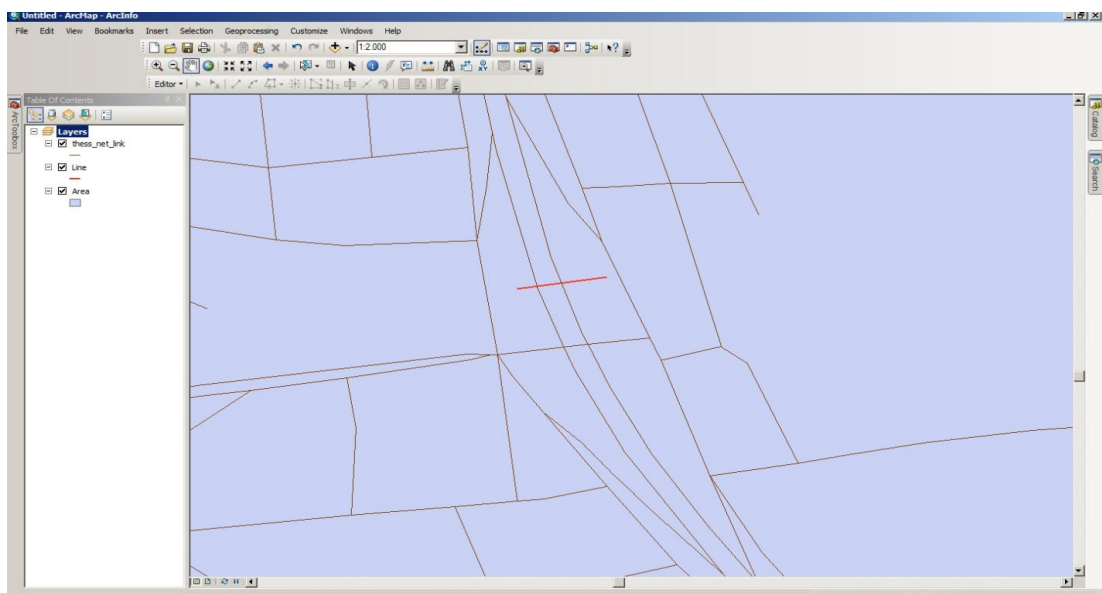


Figure 3.3: VTL Creation

The Shape File created by this process through the tool **shp2pgsql** was converted and uploaded as geographical information in the database.

3.4.3 Spatial results from sql query

The management tool for the database used did not provide the ability to display spatial results. As a utility tool the application **OpenJUMP GIS** was used; it provides connectivity to the database and displays SQL Spatial query results. The **OpenJUMP** is an open source Geographic Information System (GIS) written in the Java programming language. It is developed and maintained by a group of volunteers from around the globe.

The execution of the query is done by using the command **ST_AsBinary** of the PostGIS command collection.

```
Select ST_AsBinary(line_geom)
from linen
```

3.4.4 Application core process, how to detect a trigger

The main purpose of the mobile application is to send information about the traffic condition (speed of movement) at predetermined measurement points (VTL). The VTL as mentioned above are essentially virtual lines on a road. Their placement has been done perpendicularly to the road for which we want to obtain data (Figure 3.4).



Figure 3.4.:VTL Placement

The vertical placement was designed to allow checking by the application that the user is passing at that moment by her/his vehicle the trigger point. To detect this, the following technique has been used. The VTL is actually a line with two geographical points, the beginning and the end of it. By receiving from GPS at regular intervals the driver's position (latitude and longitude), we form a Driver Trip Line (DTL) assuming beginning of the line, the first point and end the next one. To detect if the user has passed a VTL what happens is to check whether these two lines (VTL, DTL) intersect at some point and then whether that point is a point on the VTL as they can intersect and beyond it.

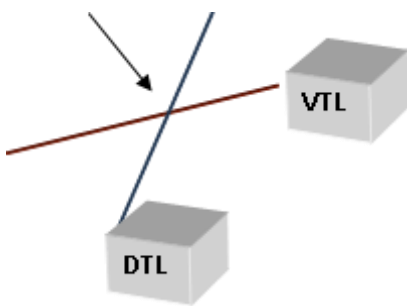


Figure 3.5 Intersection point inside of VTL

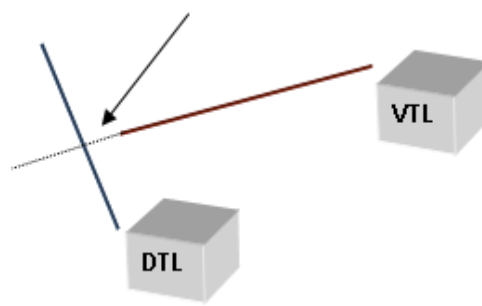


Figure 3.6 Intersection point outside of VTL

This means that certainly some points which the driver passed will not be recording and sending information but this is part of the system and a desired factor. Additional not recording may occur due to non-precision GPS signal that may appear. This is a desirable factor as not only continuous transmission of information to the server does not occur (low cost) but there is also no complete record of the path of the driver (privacy issues).

3.4.5 How to set the direction

Through the operation of the **Application core process** described above is checked when the driver has gone through the VTL. The VTL covers the entire width of the road in order to absorb the differences marked by GPS. The result of this implementation is to receive information on the two-way traffic on some streets. This creates the problem of separation of information. To allow the separation the procedure described below was followed.

During the creation of VTL, inclusion of direction has been done on the geographical axis system we are interested in. This entry was done by using attributes such as latD (latitude direction), lonD (longitude direction) that is the direction in longitude and latitude respectively (Figure 3.5).

id integer	areaid smallint	latdirect smallint	londirect smallint	descript character vai	line_geom geometry
0	1	-1	0		0105000020E61000
1	1	1	0		0105000020E61000
7	1	0	-1		0105000020E61000

Figure 3.7: VTL Table Structure

Taking into account the geographical axis system (Figure 3.6) we conclude that during its movement towards the east longitude increases while moving to the north latitude is increasing.

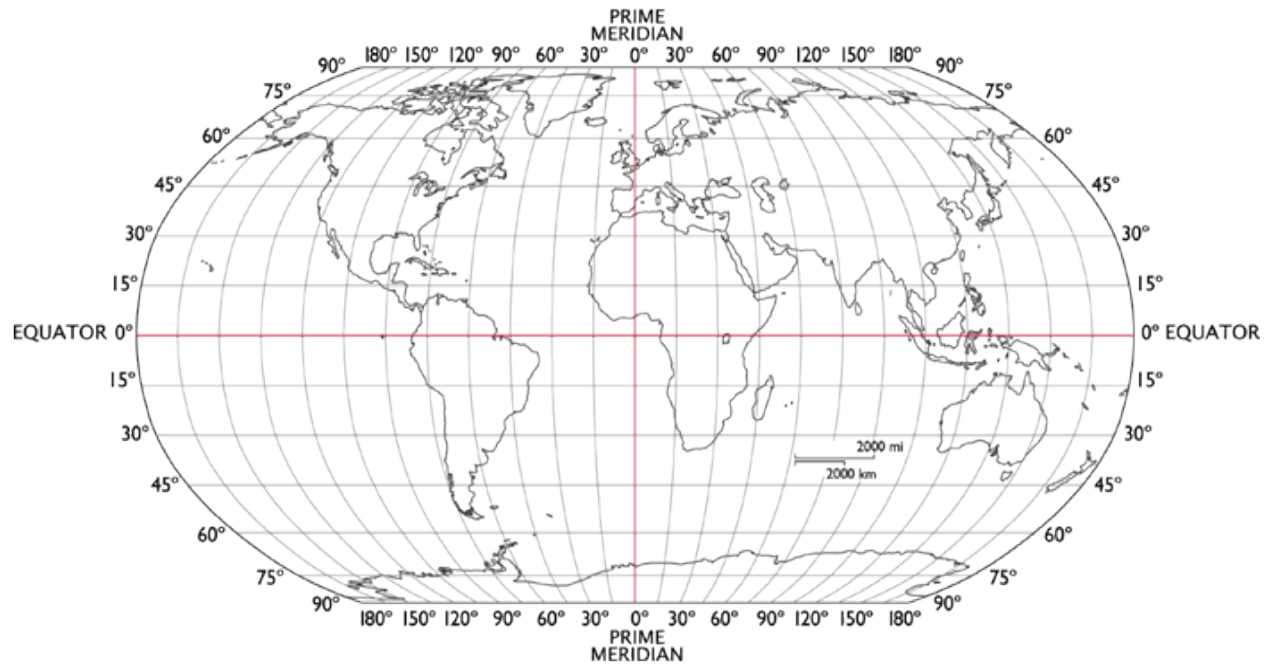


Figure 3.8: Geographical Axis System

For a part of the road we want to take data, with the driver's movement at the east end to the equator, put the attribute latD to 1. If according to the placement of the road on the geographical map we are not interested in the movement of the driver as to the longitude, then we put the property lonD to 0. Having the two points from the DTL, start point and end point, remove the latD and the lonD of the movement of the driver. Comparing these values with those of VTL that a user triggered, we determined whether the driver passed by the VTL towards the direction we are interested in.

3.4.6 Need of clustering VTL in areas.

To reduce the volume of the dispatched information to the user and for the better functioning of the system, the separation of VTL in geographical areas where they belong was advisable. The separation for more efficient operation of the system occurs after transport studies. For purposes of testing and implementation of the system the separation of Thessaloniki (Greece) was done in three regions (central-eastern, central, central-western) (Figure 3.7).



Figure 3.9: Separation of Thessaloniki, Greece.

This has resulted in that the operation of the Application core process to take place in a smaller number of data depending on the position of the moving driver. When contacting the application with Application Server a check is conducted for the area in which that moment the driver is and the system (data sent) adapts to it. With this implementation the computational requirements of the application and the volume of the information sent are reduced.

3.4.7 Display of the results of the information system

The representation of the results of the estimation of traffic conditions of the network is also available through the Internet and directly to the user's device through the application developed in the presentation tier. The display on the Internet was developed primarily for control of the system by the administrator.

For display purposes through a website (higher usability), the Google Map JavaScript API has been used. For display purposes of the traffic in the user's device, the Static Map API provided by Google has been utilized.

The incorporation of the road network in the user's device was considered as an inefficient option because of the disadvantages that emerge mainly due to high requirements in calculating force of the device and big capacity. Moreover, the dynamic character of the implementation is limited while it is not possible that the changes in the road network occur in all levels.

The device in regular time intervals asks from the application server information that concerns the situation of movement. The server transmits to the application a sequence of points that essentially portrays a segment on the network. The application creates a URL according to Google's API and asks for that service's reply. The reply to the demand of the application is a map of the region that was asked in certain zoom, width, height having displayed the points that were asked via the URL. The form of the reply is a PNG picture (Figure 3.8).

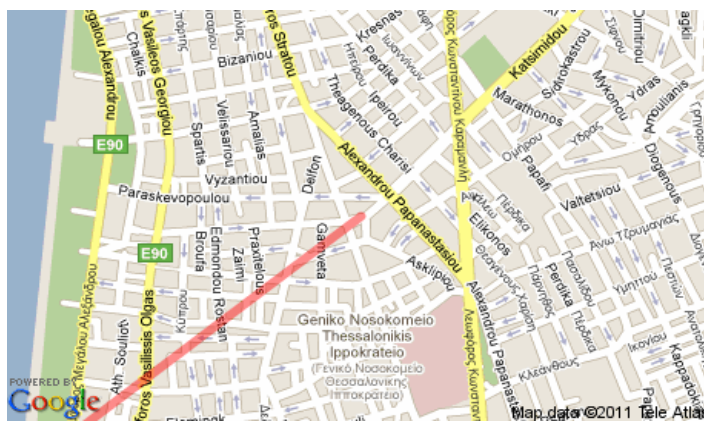


Figure 3.10: PNG Picture as a Google service reply

4. Commercial Exploitation

4.1 Overview

The commercial success of a system, such as the described traffic estimation system depends on many parameters. Actually, the current chapter deals with the parameters of commercial exploitation of the designed system. Market research on the customers and the current competitors of this application has been undertaken and the results are then analyzed.

As already mentioned, the purpose of the system is to inform users about the traffic condition of the road network in real time. The cost of implementing the system and issues concerning the safety of personal data of the users involved both directly and indirectly to the operation are seriously taken into account for the study of commercial exploitation of the application. The current chapter also refers to the benefits of creating this system (prototypes, maturity of innovation, contribution to the improvement of everyday life, etc.) and a SWOT [3] analysis of the system is carried out.

4.2 Benefits

4.2.1 Originality compared with international benchmarks

The proposed system is a highly original application in relation to international benchmarks. All the relative applications that inform users on traffic conditions on road networks (both those provided via mobile phones and those offered by personal pilots) are based on traffic data coming from expensive and difficult to maintain systems for recording movement (inductive loops, traffic measuring cameras etc.). The proposed system requires no infrastructure to function as the 'measuring stations' are the users themselves while the system operates completely anonymously and can thus work on

all road networks, not just on those which until now provided (because of the existing infrastructure) the respective information services.

4.2.2 Description of the maturity of innovation

The maturity level of the proposed innovative application is quite high, as no specific future actions are required both for development and actual use. At the level of acceptance by users, the service is very mature, given the ever increasing use of information services and devices for the traffic on road networks.

4.2.3 Feasibility of practical application

The possibilities of practical implementation of the proposed system are very good, with great potential for success and acceptance by users. The proposed system is entirely designed to be put into practice by all users with a mobile phone that desire to be informed through it on traffic situation on road networks without additional hardware requirements.

4.2.4 Evaluation of commercial exploitation

The potential commercialization of the proposed system first of all depends on the acceptance of the system by users, which could possibly be estimated through a preliminary inquiry in the market. The contact / consultation with mobile phone companies on matters relating to the proper service offered by the system to users is mandatory. As there are no other special requirements, the commercial exploitation of the system can be started directly. The potential users however should be relatively informed through appropriate channels of communication / advertising. The system can be used commercially in various ways, either by institutions / individuals who develop / evolve it or via mobile phone companies that deal with services of travel information.

4.2.5 Evaluation of the contribution to the extroversion of the country

The contribution of the proposed system in the extroversion of the country may be firstly regarded positive. The proposed system can be easily used by users abroad because of the general concept that governs it, it is independent of specific spatial constraints for its implementation. Thus, if the application receives real commercial exploitation, there will be the possibility of promoting know-how and technology developed in Greece outwards, contributing thus to the extroversion of the country. Also, because of particular interest and utilitarian purpose of the service provided through the system, it is likely that there will be interest from either individuals or mobile telephone companies or information providers for travelling abroad to exchange knowledge or cooperation in the provision of the respective service to other countries.

4.2.6 Evaluation of the contribution to viable development

The system described in this very statement positively contributes to viable development because of information that provides mobile users with the current traffic conditions on road networks and thus the possibility to avoid areas with heavier traffic - and thus environmental - conditions. Also, through proper informing of mobile users, they may avoid areas with high traffic congestion, saving thus both time and money because of the reduced fuel consumption and contributing this way to the reduction of air pollution due to emissions from cars. In this way, the system contributes positively in all areas related to viable development.

4.2.7 Evaluation of the contribution to the improvement of everyday life

The system is estimated to contribute very positively to improving the daily lives of citizens - both those who move about and all the others - in different ways as shown below:

- Avoidance of traffic congested road sections
- Avoidance of environmentally aggravated road sections
- Reduction of travel time

- Reduction of the time lost in travel due to traffic congestion
- Reduction of emissions from vehicles
- Improvement of the environmental conditions due to the better vehicle traffic on the network.
- Creation of a sense of participation in a community of users who both nurture and exploit the information of other users in total.

4.3 SWOT Analysis

SWOT is the acronym for Strengths, Weaknesses, Opportunities and Threats. A SWOT analysis is a general technique which has been applied to this system which involves generating and recording the strengths, weaknesses, opportunities and threats of the designed system. The objective of this thesis was the design and implementation of a Traffic Estimation system. Therefore, the SWOT analysis assumes that the system has been completed and is ready for installation. A detailed discussion of each part of the SWOT analysis is provided below.

4.3.1 Strengths

Zero-infrastructure system: An advantage of the system is the zero structure of the infrastructure. To implement the system only the existing infrastructure is required. The usage of the users as dynamic meters replaces the static meters of the existing infrastructure.

Low Cost: Due to the zero structure of the system's infrastructure, the cost of operation concerns only the use of service of sending / receiving data by the providers of mobile telephony. This cost is quite small compared to the cost of installation and maintenance of meters around the road network.

Real-time data: The structure of the system ensures the evaluation of the traffic conditions of the road network in real time, providing the user with immediate update.

Dynamic character of the system application: The structure of the system with the use of VTLs offers the possibility to study the efficiency and effectiveness of the system and in the next stage the change of the measurement points and frequency of data collection according to the results of these procedures. Moreover, due to the structure of the system, its usage and application in each geographic region is possible. The only requirement here is to design new VTLs.

4.3.2 Weaknesses

Cost of using data services: The use of the system requires the user's connection to the Internet and the use of data services. Telecommunication providers apply different pricing policies which, according to every indication, provide the user with quite economic packages of data services because of increased competitiveness. The cost of usage might once have been a prohibitive drawback to the development of web systems of this kind but now it is observed that this fact is changing.

4.3.3 Opportunities

Originality: All the relative applications that inform users on traffic conditions on road networks are based on traffic data coming from expensive and difficult to maintain systems for recording movement (inductive loops, traffic measuring cameras etc.). The proposed system requires no infrastructure to function as the 'measuring stations' are the users themselves while the system operates completely anonymously and can thus work on all road networks, not just on those which until now provided (because of the existing infrastructure) the respective information services.

4.3.4 Threads

Collaborative system: The structure of the system presupposes the participation of many users for proper testing, study and operation of the system. The collaborative

character of the implementation adds a weakness in the system, which through proper organization and promotion of the project can be overcome.

4.4 Summary

The system described, in view of the aforementioned study and the SWOT analysis, are considered to have a lot of opportunities and high potential for implementation and operation. The low cost of implementation because of the zero infrastructure requirements and the increasingly accessible to the user data services create such conditions for testing and implementation of the system. Although the collaborative nature of the implementation adds difficulties and disadvantages, it creates the conditions for real-time update and dynamic management of the system.

5. Conclusion and recommendations for Further Work

5.1 Conclusions derived from the current thesis

The scope of this thesis was the design and implementation of a system in order to assess the traffic situation on the road network. The aim was to create a zero-infrastructure system that offers the user the possibility of updated assessment of the traffic situation on the roads. Zero infrastructure is supported by the usability of the users of the system as dynamic meters. By using the VTLs, the structure of the system acquires a dynamic character, while ensuring the low cost of implementation and issues about personal data of the users.

The presented system consists of three parts. The application for mobile devices (first part), the application server (second part) which undertakes the communication between the application and the database, which is the third and final part of the system.

A user can easily take part since it is sufficient enough to have a device that supports MIDlet applications, GPS receiver (internal or external) and Internet connection (GPRS, Wi-Fi). Application server required is GlassFish 3 and the database used is Postgres Plus Advanced Server 8.4 with geographic data support (PostGIS). IDE used for application development is NetBeans 6.9.1. A mobile phone (Nokia N95) has been used to test the system. For the use of location services both the internal GPS of the device and two external receivers with a Bluetooth connection have been used, more specifically VGPS EMTAC, SLIM Fortuna. Desktop PC, HP Compaq 6000 Pro MT, used as database and application server.

The placement of the VTLs on the network for testing the system was done without any particular method, but the effective use of the system requires a further study by public transport experts.

During the testing of the system in real conditions it was found that the application had been successfully performing all the procedures of connection to the GPS,

authentication of user, updating the VTL table and sending data to the server during the passage through a VTL. Poor performance was observed during the depiction of the traffic on the road using Google Static Map services due to a delay in transferring packets using GPRS service . When using WLAN services no malfunction to the service was observed.

5.2 Recommendations for Further Work

In this subsection we record ideas and practices to optimize the system. These improvements concern both the technical side of the implementation as well as the design of the system.

The display of traffic using Google Static Map service creates several problems concerning both the efficiency of the mobile application and the cost of using data services. Using multimedia means, through the Internet, significantly increases the volume of the data exchanged. A solution to this issue is the integration of road network in the user's device. This fact requires knowledge of the roads that will be used to meet the needs of the user. The user will record the initial preferences for the area concerned and they will receive the application having the roads of that area. Later through the possibilities offered by Over The Air (OTA) [4] technology they will be able to receive another version of the application that will cover their needs. OTA is the deployment of wireless Java applications (MIDLet suites) from the Internet to wireless devices over a wireless network. Users need not connect their devices to the desktop with a data cable or visit a service center to install or upgrade software.

To maintain competitiveness and to improve the functionality of the application a technique for automatic network selection for the use of data services by the application could be developed [5]. Due to the exponential growth of free Internet to municipalities it is sure that the user will move on roads where free Internet access will be provided. The need for use of service data from providers will be reduced by the implementation of the automatic network selection and thus there will be reduction to the cost of using the application.

An important criterion for the efficiency and effectiveness of the system is the installation of VTLs along the roads. Through proper study of transportation agencies the basis for ensuring the accuracy of the results of the system will be improved and created [1].

Finally, in order to improve the graphical interface of the application we could utilize libraries supporting full graphical interface [2] (Appendix A contains a list of available GUI libraries).

6. References

- [1] Ban, X., Herring, R., Margulici, J.D., and Alex Bayen, "Optimal sensor placement for freeway travel time estimation". *Transportation and Traffic Theory*, Chpt. 34 (W.H.K. Lam, S.C. Wong, H.K. Lo eds.), pp. 697-721, 2009.
- [2] J2ME User Interface Libraries. At: <http://j2me.ngphone.com/opensource/ui.htm>. Last Time Accessed: Jan 2011.
- [3] SWOT Analysis. At: <http://www.quickmba.com/strategy/swot/>. Last Time Accessed: Dec 2010.
- [4] Over-The-Air Programming. At: http://en.wikipedia.org/wiki/Over-the-air_programming. Last Time Accessed: Nov 2010.
- [5] S.V. Saboji, C.B. Akki "A Client-based Vertical Handoff System in 4G Wireless Networks", *Journal of Advances in Information Technology*, Vol 1, No 4 (2010), 197-203, Nov 2010.
- [6] iOS Developer Program. At: <http://developer.apple.com/programs/ios/develop.html>. Last Time Accessed: Feb 2011.
- [7] iOS (Apple). At: http://en.wikipedia.org/wiki/IOS_%28Apple%29#SDK. Last Time Accessed: Feb 2011.
- [8] Advantages of J2ME. At <http://technology.ezinemark.com/advantages-of-j2me-platform-16d1131fd8e.html>. Last Time Accessed: Aug 2010.
- [9] .NET Compact Framework. At http://en.wikipedia.org/wiki/.NET_Compact_Framework. Last Time Accessed: Dec 2010.
- [10] Andorid Developers. At: <http://developer.android.com/guide/basics/what-is-android.html>. Last Time Accessed: Jan 2011.
- [11] Android Operating System. At: http://en.wikipedia.org/wiki/Android_%28operating_system%29#Software_development. Last Time Accessed: Jan 2011.
- [12] Mobile Application Development Trends. At: <http://www.wayneliew.com/10-mobile-application-development-trends-2010/>. Last Time Accessed: Jan 2011.

[13] Mobile Development Trends. At: <http://www.mobilopen.org/2010/06/08/mobile-application-development-trends/>. Last Time Accessed: Jan 2011.

[14] Augmented Reality. At: http://en.wikipedia.org/wiki/Augmented_reality. Last Time Accessed: Jan 2011.

[15] D.Gavalas, D.Economu "Development Platforms for Mobile Applications: Status and Trends", IEEE Software, Vol.28, No.1, pp. 77-86, Jan/Feb 2011.

[16] B.Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. Bayen, M. Annavaram, Q. Jacobson," Virtual trip lines for distributed privacy preserving traffic monitoring", *Mobile Systems and Applications (MOBISYS)*, June 17-18 2008.

7. Appendix

A. GUI Libraries for J2ME

Name	Comments	License
Paxmodept	Special features: Expandable list items and tab-menus	Commercial
TWUIK	Very powerful and really an eye-candy. Looks very sophisticated and gives your application the iPhone-Look. Unfortunately it's not free.	Commercial
J2ME Polish	Very sophisticated. Is not only a library, but a whole development-framework. Allows to deploy different variants of applications for different mobile phones or different customers. Has built-in features like auto-completion of textfields. Possibility to separate application logic and design through CSS-like layout. Plugins for Netbeans and Eclipse. Has ListLayout, TabLayout, GridLayout and TreeLayout.	GPL
Tinyline	Seems to be better for graphical drawing, than for GUIs with components like textfields and navigation.	Tinyline License
Synclast	Has containers like tablecontainer. Text in Textfields can only be entered all lower-case or all upper-case, there's no possibility to switch the enter-mode during runtime.	GPL
APIME	Seems to work only by mouse pointer navigation. Plus the javadoc is only in spanish. Nevertheless there are some nice components like progress bars and tooltips.	GPL

fire j2me	Possibility of screen rotation and popup panels. As far as I can see, it has not layout managers, it assumes that all elements are aligned as lists.	LGPL
jMobileCore	Some very clean components like checkboxes etc. but seems to support only normal alignment. no fancy containers like table-containers. Furthermore, there is no native use of the cell-phone text entry, only upper-case text fields without numbers for text entry.	LGPL
MWT - Micro Window Toolkit	Demo-Applications were quite confusing in the emulator. Some weird behavior (command-keys didn't work) and navigation via 2,4,6,8 instead of direction-keys. (Felix in the comments added, that the direction keys DO work.)	LGPL
Thinlet	Uses xml to add components. contains all the components in one class-file last midp-version was in 2002. respectively in 2005 (redesigned by Porsche Engineering (without documentation, only the java-file). the now developed version is mainly thought as a lightweight component library for J2SE and Pocket PCs (J9 VM).	LGPL
Light Weight Visual Component Library	Grid, Tree, list: A lot of possible components. Just a draft-version available at the moment	GPL
microEWT	Has BoxLayout, GridLayout and BorderLayout and FlowLayout, but doesn't have tab-layout.	GPL

J4ME	Professional looking UI, support themes, logging framework included, get GPS information from internal GPS and external GPS pucks	Apache 2.0 License
Kuix	User interface components can be arranged in an XML file. The style of the components is defined in a CSS file. Also has special components like tabbed forms. Very good looking interfaces.	GPL
TagsME	Provides the possibility to create GUIs with XML that work across different devices.	not known yet
Swing ME	A J2ME implementation of Swing GUI	LGPL
kUI	a Canvas based replacement for the high level LCDUI classes	GPL
LWUIT	UI Library, supports theming, transitions, animation and more.	GPLv2
J2ME GUI		Special license
Flemil	Uses less memory and supports themes and layout switching.	Commercial license

B. Sample: Code for VTL trigger detection during trip-- J2ME

```
for(int k=0;k<vtlTable.length;k++)
{
    if(myTrip.IntersectsDirec(vtlTable[k]))
    {
        System.out.println("INTESCTED VTL ID: "+vtlTable[k].getId());
    }
}

public boolean IntersectsDirec(VirtualLine vtl2 )
{
    Coordinates interPoint= findIntersectPoint(vtl2);

    if(interPoint==null)
    {
        System.out.println("intersect point is null");
        return false;
    }

    if (interPoint.distance(vtl2.EndOfLine)<=vtl2.EndOfLine.distance(vtl2.StartOfLine))
    {
        if(interPoint.distance(vtl2.StartOfLine)<=vtl2.EndOfLine.distance(vtl2.StartOfLine))
        {
            if (interPoint.distance(EndOfLine)<=EndOfLine.distance(StartOfLine))
            {
                if(interPoint.distance(StartOfLine)<=EndOfLine.distance(StartOfLine))
                {
                    if((vtl2.latDirection!=0) & latDirection!=vtl2.latDirection)
                    {
                        return false;
                    }

                    if ((vtl2.lonDirection != 0 & this.lonDirection != vtl2.lonDirection))
                    {
                        return false;
                    }

                    return true;
                }
            }
        }
    }
}
```

C. Sample: Code for detection of VTL and DTL intersect point - J2ME

```
Coordinates intersectPoint=null;

public Coordinates findIntersectPoint(VirtualLine vt12 )

{
    if (IntersectPoint(vt12)) //checking if exists an intersect point
    {
        double w=(EndOfLine.getLongitude()-StartOfLine.getLongitude())
        / (EndOfLine.getLatitude()-StartOfLine.getLatitude());

        double z=(vt12.EndOfLine.getLongitude()-vt12.StartOfLine.getLongitude())
        / (vt12.EndOfLine.getLatitude()-vt12.StartOfLine.getLatitude());

        double x=(vt12.EndOfLine.getLongitude()-
        StartOfLine.getLongitude()+w*StartOfLine.getLatitude()-
        (z*vt12.EndOfLine.getLatitude())) / (w-z) ;

        double y=(EndOfLine.getLongitude()+midlet.formatDouble(midlet.formatDouble(w)*(x-
        EndOfLine.getLatitude())));

        intersectPoint=new Coordinates(midlet.formatDouble(x),midlet.formatDouble(y),0);

        return intersectPoint;
    }else
    {
        return intersectPoint;
    }
}
```


D. Sample: Code to detect user region- PG SQL

```
CREATE OR REPLACE FUNCTION findclientregion(lat text, lon text, clientversion text)
  RETURNS text AS
$BODY$
Declare pointX double precision;Declare pointY double precision;
Declare result text;
Declare clientRegionID int; Declare LastVer text;
Declare comp boolean;
Begin
    pointX = to_number(lat,'99.999999');
    pointY = to_number(lon,'99.999999');
--find client area ID and store it to clientRegionID
Select into clientRegionID gid from area where ST_Within(ST_SetSrid(St_MakePoint(pointX
,pointY),4326),ST_SetSrid(area_geom,4326));
IF NOT FOUND THEN
    return 'NotInArea';
END IF;
--find Last vtl table update version and store it to Last Version
Select into LastVer LastVersion from VTLCCurrent where areaid=clientRegionID;
--compare client version and last active version
select into comp (LastVer=clientversion);
if (comp) then--return true if client have the last version
    return 'True';
else
return clientRegionID;--return false if client have to update current version
end if;
return 'error';
```

E. Sample: Code for RMS Creation- J2ME

```
boolean rmsHaveToCreate=false;;

try {

    rs= RecordStore.openRecordStore("VtlUpdateTable",false);

}catch(Exception e)

{

    rmsHaveToCreate=true;

    midlet.displayMSG("RMS not exists, creating..");

}

try{

    if(rmsHaveToCreate==true)

    {

        //open or create if needed RMS

        rs= RecordStore.openRecordStore("VtlUpdateTable",true);

        //record 0-vtl version

        String txt1="1";

        byte textVtIs[]=txt2.getBytes();

        rs.addRecord(textVtIs, 0, txt2.length());

        rs.closeRecordStore();

        midlet.displayMSG("RMS created succesfully");

    }

}

}
```

```
// Open an HTTP Connection object
    httpConn = (HttpConnection)Connector.open(url);
// Setup HTTP Request to POST
    httpConn.setRequestMethod(HttpConnection.POST);

    httpConn.setRequestProperty("User-Agent",
        "Profile/MIDP-1.0 Configuration/CLDC-1.0");
    httpConn.setRequestProperty("Accept_Language","en-US");
//Content-Type is must to pass parameters in POST Request
    httpConn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
```