SMART INFORMATION DISSEMINATION SYSTEM

By

Qurat-ul-ain Shaheen

Nabeel Arshed

Hira Kiani

Faryal Aamir

(BESE-10)

Submitted to the faculty of Computer Science Department Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of BE in Software Engineering

March 2008

ABSTRACT

Campus-wide networks are prevalent in all major educational institutions today. Essentially pervasive intranet, a typical campus network interconnects hundreds of departments across dozens of buildings, providing high-speed access for both students and staff. SIDS application is designed as a cost effective system for keeping the members of computer science department informed about the latest department news while achieving a paperless culture in the administration area of the department.

The project aims at establishing communication and collaboration, which is vital to the success of building a strong knowledge culture and facilitating collaborative research for any educational institute. The project incorporates both wired and wireless communication to allow the students to have access of the administration updates. These updates include event notifications, scheduling information and attendance alerts. A prototype system has been developed which can be tailored for real world applications with a little effort.

Key benefits of the application include ease of integration in the existing network at negligible cost and seamless integration into the campus environment. The educational institutes can be better equipped to achieve their primary objective of providing the best knowledge experience for its students and staff.

DEDICATION

To Allah Almighty with the love and respect that He alone commands

ACKNOWLEDGMENT

All praise and acknowledgement to Allah Almighty who is always there for us. As always, He guided us in the right direction and helped us in the best manner, always showing us the best means to do the right thing. We thank our supervisor Brig. (Rtd) Dr. Muhammad Akbar who was always very cooperative and full of insight in all things especially things that mattered. Our co supervisor Mr. Umar Mahmud also deserves our hearty acknowledgement for taking a keen interest in our progress and for his patience and optimism in always showing us a new route in what seemed like quite a number of dead ends to us. It was a great experience to work alongside so many intelligent and helpful colleagues and friends who deserve our heartiest acknowledgement and thanks for their useful suggestions and creative ideas. We would also like to thank the entire computer science department including all the faculty members and administration staff for their support, cooperation and guidance in all matters from giving us the paper clips to giving us useful project management tips and techniques .Especially the lab staff and administrator Mr. Asif for their extremely efficient and valuable lab management skills. Lastly, we would also like to thank our parents and family for their support and encouragement.

TABLE OF CONTENTS

CHAPTER 4

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTORDUCTION

The project belongs to the domain of “pervasive computing” which aims at making the environment adaptive to user needs, making it intelligent by integrating technology [1]. In the domain of pervasive computing, while dealing with any problem focus is always on the user and their needs. Technology that needs to be employed is derived from the needs of that particular environment.

In a university’s department critical information such as the reschedule of an exam needs to be passed on to the students who may not be assembled at one place so it has to be passed on quickly and reliably. The need for such a system arises which provides fast and reliable communication between students and the department’s administration that has to be inevitably wireless [2]. The key for any system to succeed in such an environment is that it has to use technology that is easily available with students and faculty members and can be utilized using the available architecture. Mobile phones have today become the most commonly available devices and which it is presumed all the users in the targeted environment will have, so they are utilized in this project as a means to communicate with the user.

Smart information dissemination system (SIDS) builds on the Bluetooth technology and attempts to create a pervasive environment in a university campus where students and members of faculty are kept informed about their class timings, attendance records and any other notifications through their mobile phones [3]. The wireless communication between the mobile phone of the user and the system is established using Bluetooth. Bluetooth offers both reliability and cost effectiveness to be used in an environment such as a University's department for which this project is intended [4]. Smart Information Dissemination system strives to achieve a pervasive environment where the users, after authenticating themselves, are wirelessly connected to the system and receive various updates depending upon their profile maintained in the system. In the realm of pervasive computing such information dissemination using Bluetooth has not been realized before but it offers reliability and efficiency without any expensive architecture. The project delivers a reliable information dissemination system, which caters for the needs of the students related to information retrieval while in the university campus.

1.2 BACKGROUND

A pervasive environment is one in which users can access relevant information and services to facilitate their everyday tasks through the use of most relevant technology. Devices that use the technology provide relevant services to appropriate users. Pervasive computing strives towards distributing computational intelligence throughout the physical environment, making devices intelligent so as to perform tasks for the user. Pervasive computing is about embedding technology into the environment that as a result provides efficient solutions to problems and simplifies tasks and processes.

1.3 PROBLEM STATEMENT

SIDS allows students to communicate with dissemination service using mobile devices in addition to personal computers. The project aims at establishing communication and collaboration, which is vital to the success of building a strong knowledge culture and facilitating collaborative research for any educational institute. The project incorporates both wired and wireless communication to allow the students to have access of the administration updates. These updates include event notifications, scheduling information and attendance alerts.

1.4 AIMS AND OBJECTIVES

The aim for this project is to build an information dissemination system that students can use to retrieve information while in the campus. The Smart Information Dissemination system creates a pervasive environment where the mobile phones along with computers are used for making the environment conducive so that information meant for the user can be easily disseminated.

1.4.1 PRIMARY OBJECTIVE:

The primary objective of SIDS is to develop a student information dissemination system in pervasive campus. The students have access of the administration updates including event notifications, scheduling information and attendance alerts.

1.4.2 SUBOBJECTIVES:

The sub-objectives of the project are to establish communication between the computer and the mobile devices using Bluetooth, to provide search based retrieval of information and to provide user authentication.

The communication between the computer and the mobile device is established via Bluetooth. A Bluetooth stack is used for this purpose that has layers defined so that message is passed reliably between two Bluetooth communicating devices [5]. For the user to use the system a GUI's have to be developed both at the computer and the mobile-end. GUI for the mobile device is made simple so that the user can easily process the information it is not too heavy for mobile device's limited processing capability. Before the user can use the system he/she has to be authenticated by providing his/her ID or Password which are then authenticated by the system [6].

1.5 TASKS:

The basic tasks performed by SIDS fall into two categories i-e information provided to students by administration of the computer science department and search based information retrieval

SIDS primarily deals with two kinds of information, first are the notifications or alerts that are issued by the university administration that are transmitted to the desired section of students in the campus and delivered later to those students that were not visible to the system at that time secondly the user may want to retrieve any desired information from the system at any given time, the user can search for the information based to different criteria and the system then responds to the user in reasonable time.

1.6 OUTLINE OF REPORT

In this chapter 1 the overview, background, aims and objectives of the project were discussed briefly. Chapter 2 discusses literature overview that is carried out for the project. Chapter 3 discusses system design and its architectural model. Chapter 4 discussed tools, techniques and implementation details. Chapter 5 looks into the limitations and future enhancements of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter includes a summary of the literature review that was carried out for the understanding the project domain and defining the scope of the project. The technologies, hardware and protocols used in this project for communication are briefly discussed. The comparison of existing systems in pervasive computing is carried out by listing all the differentiating features in form of a comparison table and by giving the architectural designs. The proposed system is also discussed.

2.2 PERVASIVE COMPUTING

The term “pervasive computing” also known as ubiquitous computing refers to the embodiment of computation into our everyday life [7]. In a pervasive computing environment the basic objective is to make the environment adaptive to needs of the user by making it intelligent through the integration of technology into the everyday aspects of the human environment.

2.2.1 OBJECTIVES OF PERVASIVE COMPUTING

Pervasive computing has emerged as a result of advancements in the field of technology. Pervasive computing is about embedding technology into our environment, as a result providing efficient solutions to our problems and making tasks and processes simplified. It aims at combining current network technologies with wireless computing. Pervasive computing is not only restricted to the territory of personal computers, it also deals with almost any device ranging from household items to the hi-tech appliances that can be embedded with chips to connect devices to an infinite network of other devices [8]. The idea behind the pervasive computing is that our main focus is on the users and their tasks and activities. We are not concerned about the technologies or the devices used. The existing hardware or the networking infrastructure however cannot run all pervasive applications because of heterogeneity, network connectivity, automatic configuration with the underlying system, invisibility, scalability, integration, security, piracy and safety, fault tolerance and data loss due to limited wireless links.

2.2.2 HISTORY OF PERVASIVE COMPUTING

In the mid-1970s mobile computing and pervasive computing were considered as major evolutionary steps in the line of research. In the early 1990's mobile computing was born with the advent of full-function laptop computers and wireless LANs. The four key constraints of mobility that forced the development of specialized techniques in addition to many basic principles of distributed system design are erratic variation in network quality, lowered reliance and robustness of mobile elements, limitations on local resources imposed by weight and size constraints, concern for battery power consumption.

By 2000, mobile computing research began to touch upon issues that we now identify as the purview of pervasive computing. In 1991 paper entitled "The Computer for the 21st Century" by Mark Weiser of Xerox PARC laid foundation of pervasive computing. In this paper he observed, "*The most profound* *technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*." Thus, the essence of pervasive computing lies in the creation of environments saturated with computing and communication that are gracefully integrated with human users [9].

The revolution in the area of pervasive computing has begun to affect our everyday lives. In the future it will often be invisible, and the user interface will be natural. One important aspect in pervasive computing is networking that is data, once entered, will never have to be entered again, but will be readily available whenever and wherever needed [10]. The current phase of pervasive computing, in which computers are already being embedded in many devices, can be achieved by assuring that computing is spread throughout the environment, Users are mobile, information appliances are becoming increasingly available and communication is made easier between individuals, between individuals and things, and between things [10]. Architecture of pervasive computing is shown in figure 2.1 [11].

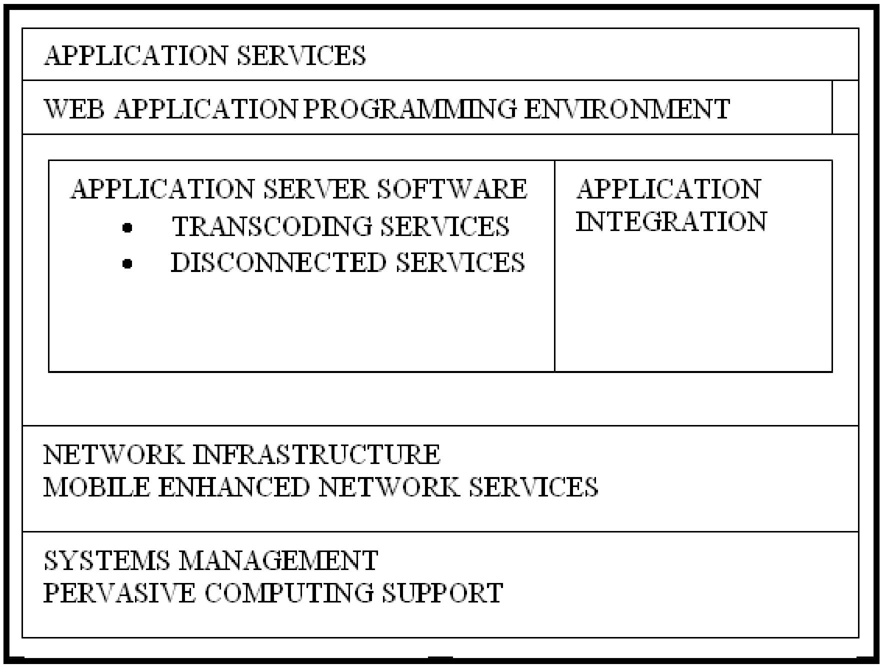

Figure-2.1: Architecture of pervasive computing

2.3 BLUETOOTH

Bluetooth is a wireless technology that enables any electrical device such as mobile phones, headsets, and PDA's and portable computers to communicate and send data to each other wirelessly. Bluetooth connections take place over a 2.4 GHz frequency, and have a range of approximately 10 meters (33 feet). It has been specifically designed as a low cost, low power, radio technology, which is particularly suited to the short range Personal Area Network (PAN) application. It focuses on low cost, low size and low power that distinguishes it from the IEEE 802.11 wireless LAN technology. Bluetooth supports both point-to-point wireless connections without cables between mobile phones and personal computers, as well as point-to-multipoint connections to enable ad hoc local wireless networks [12].

2.3.1 ORIGIN OF BLUETOOTH

Bluetooth makes connections just like cables connect a computer to a keyboard, or wire connects an MP3 player to headphones. Bluetooth technology is actually derived from a combination of wireless technologies that are united by the Bluetooth specification under the title “Bluetooth technology”. Bluetooth technology was first developed by Ericsson and then formalized by a group of electronics manufacturers (Ericsson, IBM, Intel, Nokia, and Toshiba) who joined forces to form a private trade association known as the Bluetooth Special Interest Group (SIG). They established a global standard on formal introduction of the Bluetooth specification to the public. In this way they created a universal way for mobile computers, cell phones, and various other devices to wirelessly connect with one another [13].

2.3.2 MODES OF OPERATION

An interesting aspect of the technology is the instant formations of networks once the Bluetooth devices come in range of each other forming a Pico net. A Pico net is a collection of two devices or more than two devices connected via Bluetooth technology in an ad hoc fashion. Multiple independent and non-synchronized Pico nets can form a scatter net. A device can be a part of more than one Pico net by means of time multiplexing. When a device is connected to another device, a point to point connection is established. If it is connected to more that one (up to 7), a point to multipoint connection is formed [14]. The Bluetooth system supports both point-to-point and point-to-multi-point connections as can be seen from figure 2.2 [14].

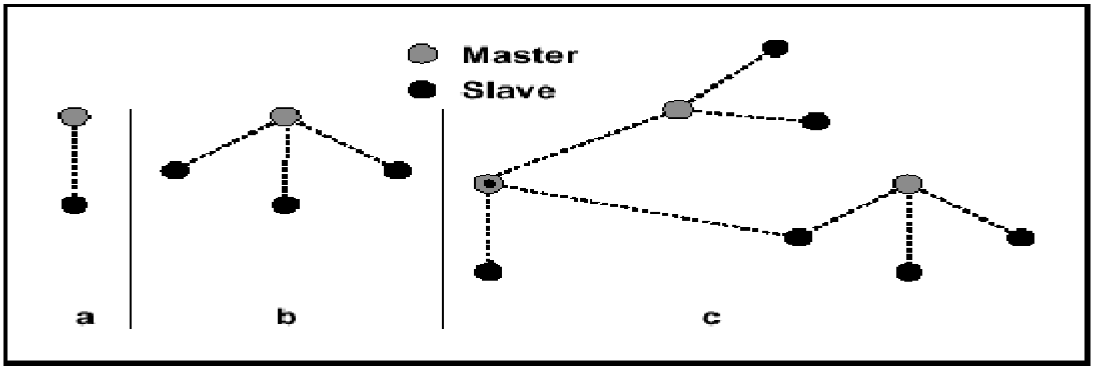

Figure-2.2: Pico nets with (a) single slave operation

(b) multi-slave operation (c) scatter net operation

2.3.3 BLUETOOTH PROTOCOL STACK

Bluetooth is both a hardware-based radio system and a software stack that specifies the linkages between layers. Well-defined layers of functionality ensure interoperability of Bluetooth devices. The protocol stack is the core of the Bluetooth specification that defines how the technology works (figure 2.3).

Radio layer is responsible for the modulation and demodulation of data into RF signals for transmission in the air. Above the radio layer is the base band and link controller layer. The base band portion of the layer is responsible for properly formatting data for transmission to and from the radio layer. The link controller layer is responsible for carrying out the link manager’s commands and establishing and maintaining the link set by the link manager. The link manager translates the host controller interface (HCI) commands it receives into base band-level operations. The HCI (host controller interface) layer acts as a boundary between the lower layers of the Bluetooth protocol stack and the upper layers. L2CAP (logical link control and adaptation protocol) layer is responsible for repackaging the data packets it receives from the higher layers into the form expected by the lower layers. The SDP (service discovery protocol) defines actions for both servers and clients of Bluetooth services. The RFCOMM protocol emulates the serial cable line settings and status of an RS-232 serial port. RFCOMM connects to the lower layers of the Bluetooth protocol stack through the L2CAP layer. OBEX (object exchange) is a transfer protocol that defines data objects and a communication protocol two devices can use to easily exchange those objects [15].

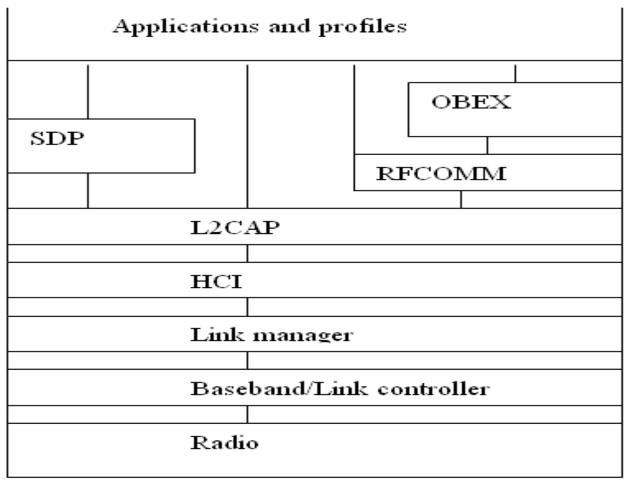

Figure-2.3: Bluetooth protocol stack

(Taken from Bluetooth Device Access guide)

2.3.4 LIMITATIONS

Bluetooth wireless standard has its limitations, which may consequently set limits on any system employing this technology for wireless communication. Some of the significant limitations are discussed below [16]:

2.3.4.1 LOW SPEED

Currently transfer rate of data in Bluetooth is about 3 megabits (375 kilobytes) per second. This slow rate is only obvious when very large files are being sent. However synchronizing of contacts or talking on a wireless headset, tend to work fine.

2.3.4.2 COMMONLY USED FREQUENCY BAND

The frequency range of Bluetooth enabled devices is 2.4 GHz band. It is the same unlicensed frequency that is used by many other electronic devices such as microwaves, cordless phones, and the majority of Wi-Fi devices. This frequency sharing could cause signal collision that leads to slower overall network performance.

2.3.5 ADVANTAGES

The key benefits of Bluetooth wireless technology that make its employment worthwhile are listed below [16].

2.3.5.1 EASE OF CONFIGURATION

Configuration of Bluetooth enabled devices is easy. It only requires getting the two devices near each other and turning them on without the need to install drivers. Seamless communication between any two wireless products featuring Bluetooth technology regardless of brand or country of their origin is possible.

2.3.5.2 INHERENT SECURITY OF THE STANDARD

Wireless security is a primary concern for consumers as identity and information theft is pervasive these days. Bluetooth offers built-in security, unlike Wi-Fi where security is the responsibility of the network administrator. Security in Bluetooth is realized using adaptive frequency hopping (AFH). In AFH "hopping" of frequency between the allotted 79 frequencies 1,600 times per second takes place making it extremely difficult to intercept transmissions.

2.3.5.3 LOW POWER CONSUMPTION

Up to eight devices are able to connect simultaneously to the same Pico net, and that it has very low power consumption. The ability to exchange information across devices gives Bluetooth technology great potential.

2.3.5.4 FREE OF COST

Bluetooth is the most widely supported wireless standard in the world today. It does not require membership fees or services to sign up for.

2.3.5.5 PAIRING OF DEVICES

Pairing of devices prior to communication is required. As a result a unique, synchronized frequency-hopping pattern for their Pico net is established. Typically users have to enter a numerical passkey to pair devices.

2.4 PROBLEM STATEMENT

To allow students to communicate with dissemination service using mobile devices in addition to personal computers.

2.5 EXISTING SYSTEMS IN PERVASIVE COMPUTING

SIDS allows students to communicate with dissemination service using mobile devices in addition to personal computers. The project aims at establishing communication and collaboration, which is vital to the success of building a strong knowledge culture and facilitating collaborative research for any educational institute. The project incorporates both wired and wireless communication to allow the students to have access of the administration updates. These updates include event notifications, scheduling information and attendance alerts.

2.5.1 OXYGEN

Project Oxygen to develop pervasive, human-centered computing is a research project at the Technology. This project uses face identification technology for handheld computers. Capturing face images is less disturbing than fingerprint, iris, or retinal scanning as person's face is exposed to the public. Face identification can be done with a standard low-resolution camera whereas other biometric devices require specialized hardware. In Face identification system the user takes snapshots of his or her face, which can be performed in the background as the handheld is being used. The Oxygen architecture consists of handheld terminals; computers embedded in the environment, and dynamically configured networks which connect these devices. Oxygen enables pervasive, human- centered computing through a combination of specific user and system technologies [17] [18].

2.5.2 ONE WORLD

One world provides an integrated, comprehensive framework for building pervasive applications. It targets applications that automatically adapt to highly dynamic computing environments, and it includes services that make it easier for developers to manage constant change. Pervasive computing provides an attractive vision for accessing information anywhere and anytime. However, a considerable obstacle to realizing this vision is the development of applications that continually adapt to an ever-changing computing environment and still function when people move through the physical world or switch devices. In the one world project an appropriate system is created to support the developers so that they can effectively build and deploy adaptable applications. This architecture includes services that simplify the task of coping with constant change [19].

2.5.3 GAIA

Both physical and virtual entities may be allowed to seamlessly interact by extending the reach of traditional computing systems to include the devices and physical space surrounding the machines. Physical spaces become interactive systems, or in other terms, Active Spaces. Such environments are analogous to traditional computing systems and Active Spaces are viewed as one object, composed of input/output devices, resources and peripherals. However, the heterogeneity, mobility and sheer number of devices makes the system vastly more complex. Applications may have the choice of a number of input devices such as location sensing system, mouse, pen, or finger and output devices, such as an everywhere display, monitor, PDA screen, wall-mounted display, speakers, or phone. Gaia brings the functionality of an operating system to physical spaces as shown in figure-2.4 [20]. Common operating system functions are supported, such as events, signals, file system, security, processes, process groups, etc. Gaia extends typical operating system concepts to include context, location awareness, mobile computing devices and actuators like door locks and light switches. The main goal of the project is to design and implement a middleware operating system that manages the resources contained in an Active Space. An operating system for such a space must be able to locate the most appropriate device, detect when new devices are spontaneously added to the system, and adapt content when data formats are not compatible with output devices. Traditional operating systems manage the tasks common to all applications; the same management is necessary for physical spaces [20].

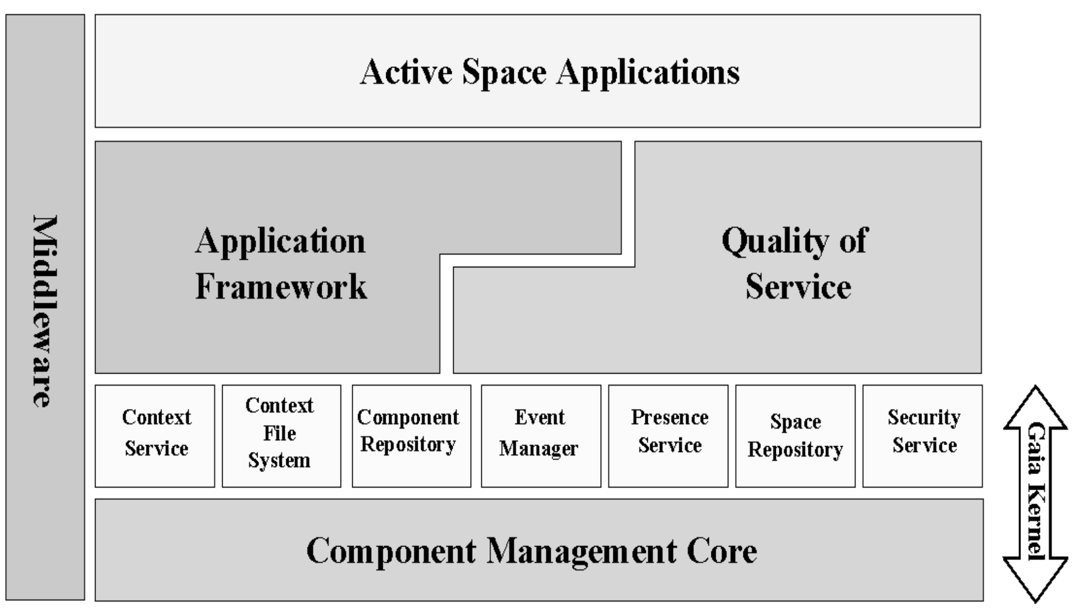

Figure-2.4: Gaia Architecture

2.5.4 CAMUS

Context-awareness is one of the fundamental requirements for achieving user-oriented ubiquity. Context-Aware Middleware for Ubiquitous computing Systems (CAMUS) envisions a comprehensive middleware solution that not only focuses on providing context composition at the software level but also facilitates dynamic features retrieval at the hardware level by masking the inherent heterogeneity of environment sensors. Complexity is handled by providing 'separation of concerns' between environment features extraction, contextual data composition and context interpretation. With a systematic approach, CAMUS is proved to be a flexible and reusable novel middleware framework. The vision of ubiquitous computing, with devices seamlessly integrated into the life of everyday users, and services readily available to users anywhere all the time is becoming a reality. A ubiquitous computing environment is characterized by a diverse range of hardware (sensors, user devices, computing infrastructure etc) and equally diverse set of applications, which anticipate the need of users and act on their behalf in a proactive manner. One of the major goals of context-aware computing is to provide services that are appropriate for a person at a particular place, time, situation, etc [21] [22]. Table 2.1 shows a comparison of differentiating features of the pervasive systems discussed.

Table 2.1 Comparison of SIDS and existing research study.

Project	Underlying communication?	Architecture		Data structure
	Bluetooth [23]/ 802.11 [24]/ Ethernet [25]/	Distributed[26] /centralized	Modular [27] /layered	type
C2K [28][29]	802.11, Ethernet	Centralized	Modular	RDF [41]
CAMUS [30][31]	Not specified	Distributed	2-tier [38] [39]	Ontology based [42]
Aura [32]	802.11	Distributed	Layered [40]	Ontology based
Gaia [33]	Not specified	Distributed	Layered	Ontology based
One World [34]	Not specified	Distributed	Layered	Not specified
Oxygen [35][36]	802.11	Distributed	Layered	Not specified
Endeavour [37]	802.11	Distributed	Layered	RDF
SIDS	Bluetooth	Distributed	Modular	Entity based

2.6 CONCLUSION

In this chapter the overview of the domain of the project, protocols, technology, existing systems in pervasive computing were discussed. The analysis of existing pervasive systems revealed that the contribution is mainly research oriented. Almost all of the projects listed provide 802.11 wireless supports mainly because they are aiming to achieve wide range wireless connectivity without taking into consideration the budget constraints. SIDS however, is based on a simple design and the concept of using easily accessible devices to meet the low budget constraint.

CHAPTER 3

SYSTEM DESIGN

3.1 INTRODUCTION

This chapter describes in detail the design approach used for developing the SIDS application and justifies the said approach by highlighting the key advantages resulting from it. Software engineering practices were followed side by side in the design and the design approach is explained in the light of these aids as well.

The SIDS application is modular in design and the distributed paradigm followed is middleware-oriented, in which a disseminator component is added between client and server to access the information. Figure 3.1 shows the system architecture, which is three tier, modular and distributed. The client tier provides the GUI to interact with the user. The client end is installed on the user's java enabled mobile device. The user then communicates with the system using Bluetooth. Middle tier is essentially the service layer. It is responsible for making the communication with the user over the Bluetooth. It also receives alerts/notifications from the application server on the data tier and forwards any search queries to the application server. The most important function of the middle tier is to authenticate the users. For this purpose it has the authentication database, which has all the authentication details of the users. The middle tier is connected to the system database on the data tier through a wired link (local area network).

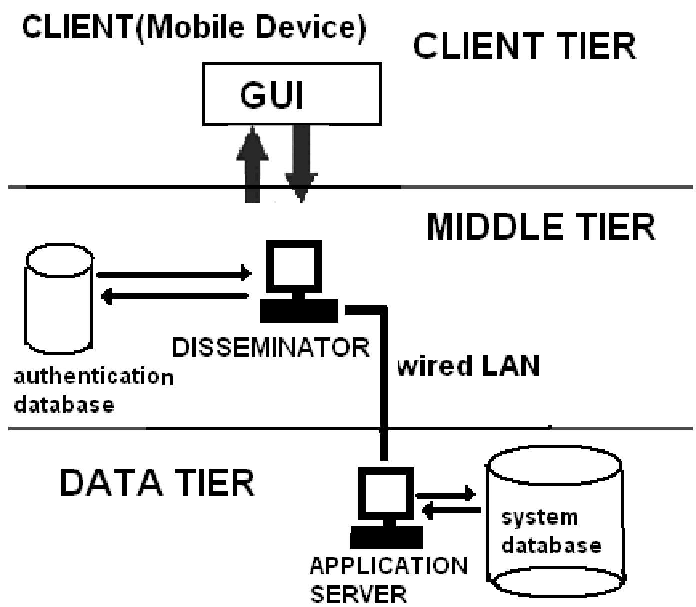

Figure-3.1: block diagram of SIDS architecture

This is the part of the system where all the service details of the system are kept, since SIDS deals with the alerts and notification related to a department at a university campus. This will include classes' schedules, any event notification, and attendance alerts. All this information is stored in the system database.

Since the system has wireless links involved so the amount of unnecessary traffic should be reduced to make it more efficient. For this purpose, information needs to be disseminated in a smart way that is, maintaining an information profile of our users, logging the information they have already received so only the latest information is passed onto them and it is not duplicated. So before broadcasting a message the relevant audience is decided here.

The architecture diagram shows three primary modules namely client (mobile device, disseminator (Session Manager) and application server.

3.2 PRESENTATION – TIER (Client Tier)

The first tier is the client tier, basically the application interface. This tier includes the GUI application running on the mobile devices of the users. There is a wireless connection between the mobile device and the second tier: the disseminator, which is the session manager. This wireless communication is established using the Bluetooth protocol stack.

3.3 MIDDLE-TIER

The second tier includes the disseminator that maintains its own database for the user authentication. The disseminator has the Bluetooth dongle attached with it. It is responsible for session management and service discovery. It performs the following basic operations:

3.3.1 Authenticating Client Devices

Authentication of the client devices is being carried out in the middle tier. The database at this level provides efficiency and simplicity of design, the authentication details don't have to travel to the data tier and acknowledgment is sent back to client. It receives the login parameters from the user which it authenticates using the associated user authentication database.

3.3.2 Receiving Updates from the Application Server

The updates are received from the application over wired LAN. The updates received are then transmitted to the user using Bluetooth.

3.3.3 Forwarding User Request to the Application Server

The user may request for an update at some point of time, this request is then forwarded to the application server. The request is not a complete query but some integer value received from the mobile device that represents a particular query. The query is then forwarded to the application server by the disseminator residing at the middle tier.

3.3.4 Delivering Updates to the User Using Bluetooth

The communication between the client and the middle tier is established using Bluetooth.

3.4 DATA-TIER

The third tier is the data tier that includes the application server and the system database. The application server has a wired-LAN connection with the disseminator and a GUI interface for managing the database. This is the module that makes the system intelligent in that it is capable of automatically detecting the updates in the service and delivering these updates timely to the disseminator, which in turn sends notifications and alerts to the client device (mobile device). The modular architecture of SIDS provides the following key advantages:

3.4.1 Ease of Maintenance

The system is modular which means that there are relatively independent components making up the system as a whole so maintaining them is easy. The database additions are done in the data tier so the client end and the middle tier remain static once they are integrated to the system.

3.4.2 Ease of Integration

The system being modular is easier to integrate. The central service layer or the data tier can have several disseminators integrated with it to increase scalability. The user devices can connect with the disseminators using Bluetooth.

3.4.3 Flexibility

The system is intended to work in a campus environment. Since the system is modular it gives us the flexibility. Disseminators can be placed as per requirement to allow the users to connect via their Bluetooth devices. By making the service layer and disseminator layer separate, connecting to users becomes more flexible, giving additional room to work with the physical constraints. Thus, the application layer can be maintained at a central place and then several disseminators can be placed at convenient locations for user connectivity.

3.4.4 Scalability

The SIDS application also addresses the scalability issue. By implementing the concepts of scatter net and Pico net we can enhance the scalability factor for the application.

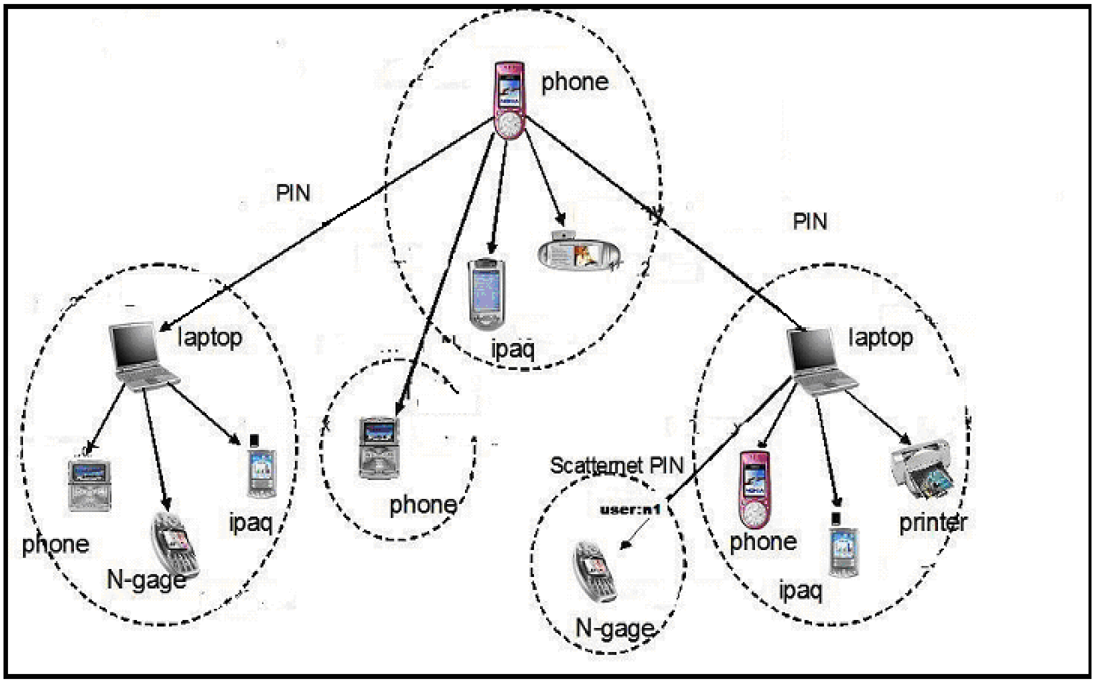

Figure- 3.2: Bluetooth scatter net example

If the Bluetooth device attached with the disseminator is over loaded by the amount of connections, and additional disseminators are not available, implementing scatter net or Pico net can solve this issue. The application since it employs Bluetooth can fully utilize the wireless environment, as the Bluetooth technology is master/slave by nature. A Bluetooth device that will connect to the disseminator can be assigned as a master by function. A master can connect to seven slaves at the same time forming a Pico net. This is the point to multipoint feature that makes the Bluetooth technology more adaptable and useful for environments like that of a university campus since scalability can be achieved in a temporary fashion without much cost.

3.5 SEQUENCE DIAGRAMS

The sequence of events followed in the system are shown in figure 3.3 where the user instantiates the process. The user will use his mobile device to interact with the system. For this the Bluetooth has to be turned on the client device. The mobile device searches for the Bluetooth connection with the disseminator. The disseminator accepts the connection request. Next the authentication details are sent to disseminator, which include the username and the password. The update request is sent to the disseminator, which is duly forwarded to the application server, the application server then uses its service database to get the latest update. The application server maintains a list of the last updates sent to the users so that always the latest updates are sent.

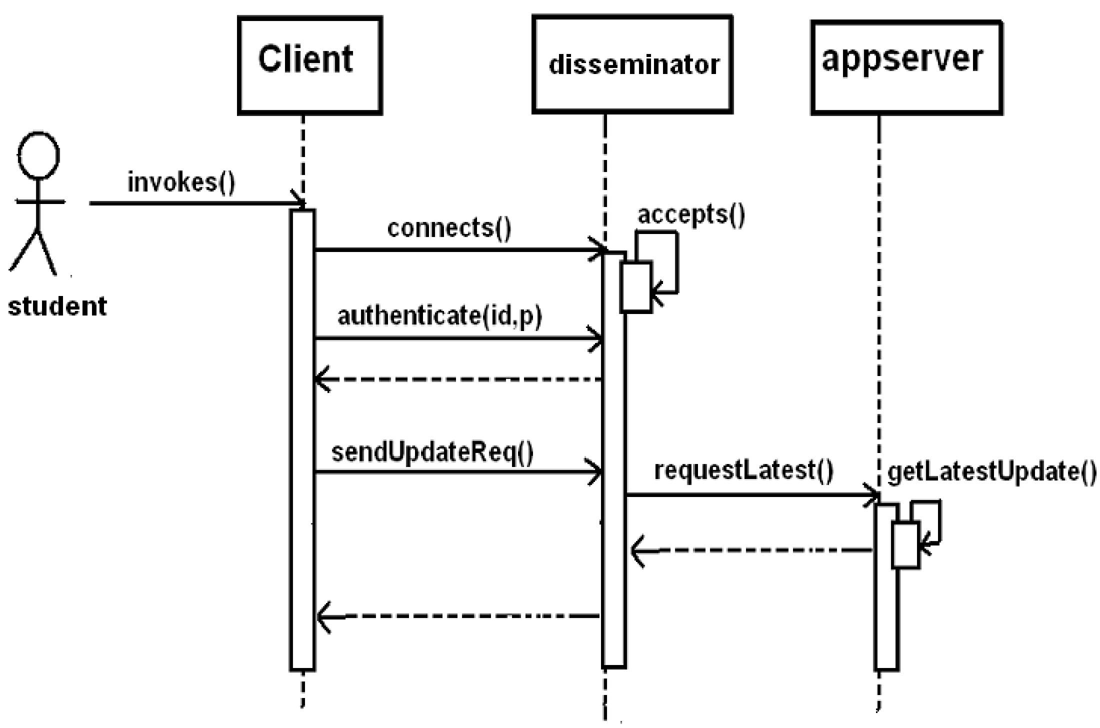

Figure- 3.3: sequence of events for requesting updates from the system.

In the event of an updating of the service database at the application server, the notification is then generated for the disseminator, which is then forwarded to the client. The notification is only delivered to the clients who have opted to receive update notifications. This sequence is shown in figure 3.4.

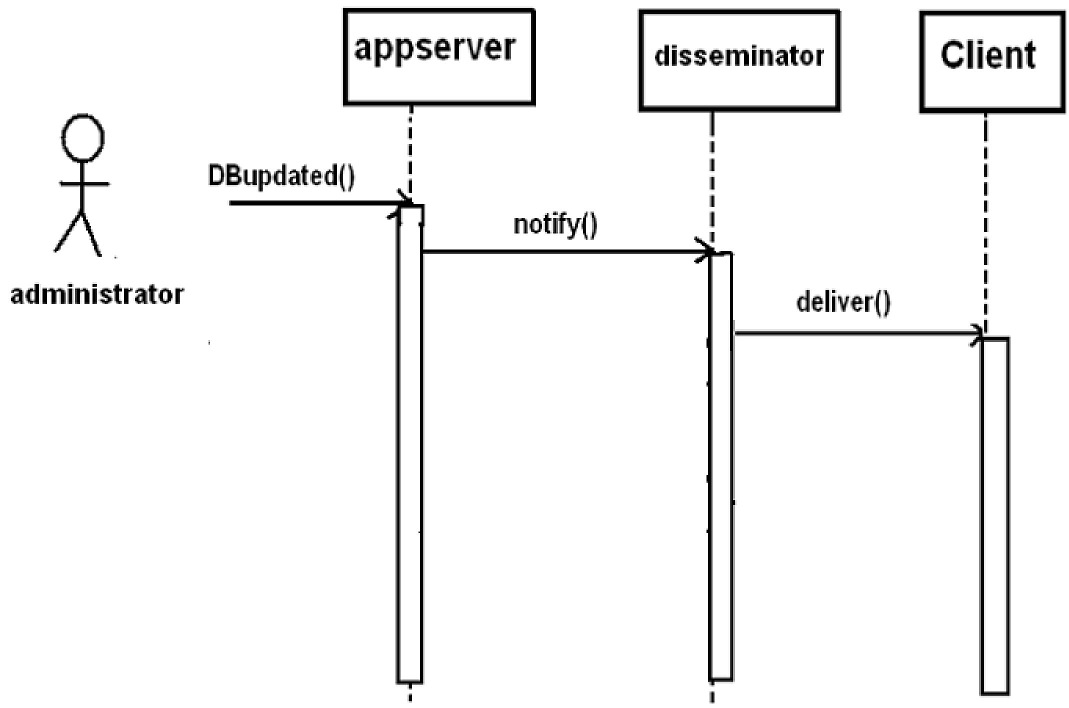

Figure- 3.4: delivering automatic updates to the users

3.6 FLOWCHART DIAGRAMS

The flowchart diagram in figure 3.5 shows the client end flow in the system. The user name and password is entered by the user, which is then authenticated by the disseminator. The user then has two options, one it can get the latest updates or he can search for the particular updates. In case of automatically receiving latest updates he may be prompted for receiving the pending messages that he can refuse or accept. In case of searching for updates the user will get the latest update/notification, which he has not received yet.

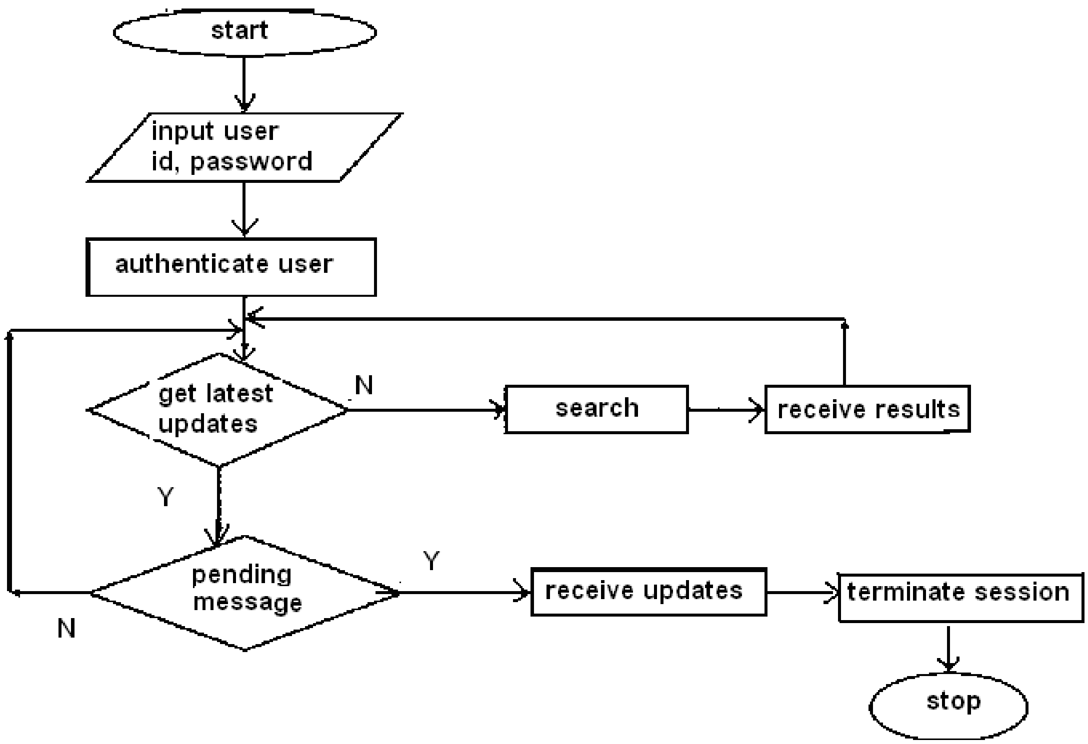

Figure- 3.5: flow chart showing the client participation

The service record is added to the service database at the application layer. The flow chart in figure 3.6 shows how information travels from the application layer to the user depending on the actions taken by the user. The system will wait for an incoming connection by the user. Once the user has provided the authentication details, these are authenticated by the system. A session is then initiated with the user and the user's mobile device gets connected with the disseminator. The user at this point can choose to receive automatic updates or take no action. On receiving automatic updates the user can also receive pending alerts depending on user input.

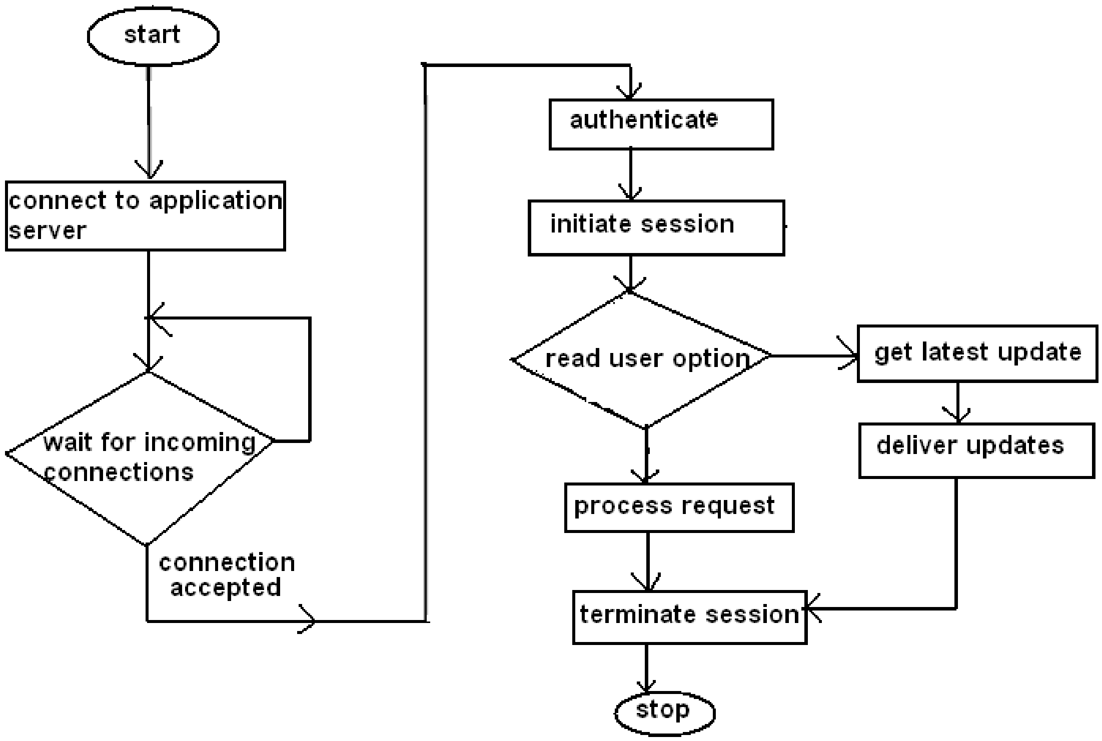

Figure -3.6: flow chart showing the server participation.

3.7 USE CASE DIAGRAMS

The use case diagram in figure 3.7 shows two actors, the application server and the system administrator. The application server delivers notifications to the users and keeps a look out for new notifications so that it can inform the users. The administrator adds new updates to the system database.

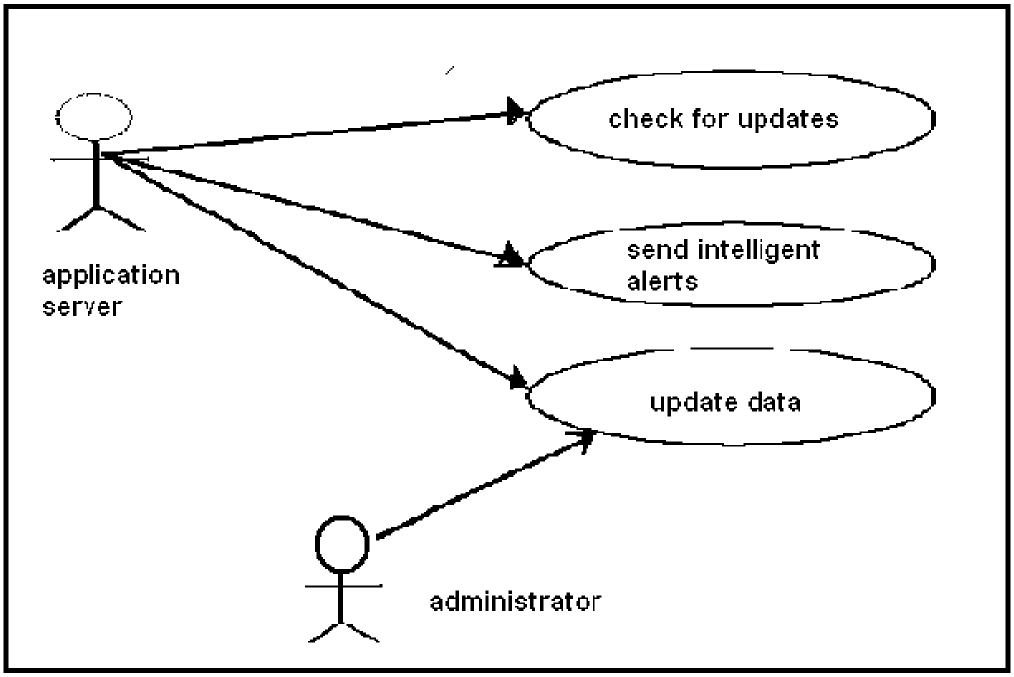

Figure- 3.7: Server use case diagram

In the use case diagram in figure 3.8 the actor is the user who can choose to initiate a number of actions in the system.

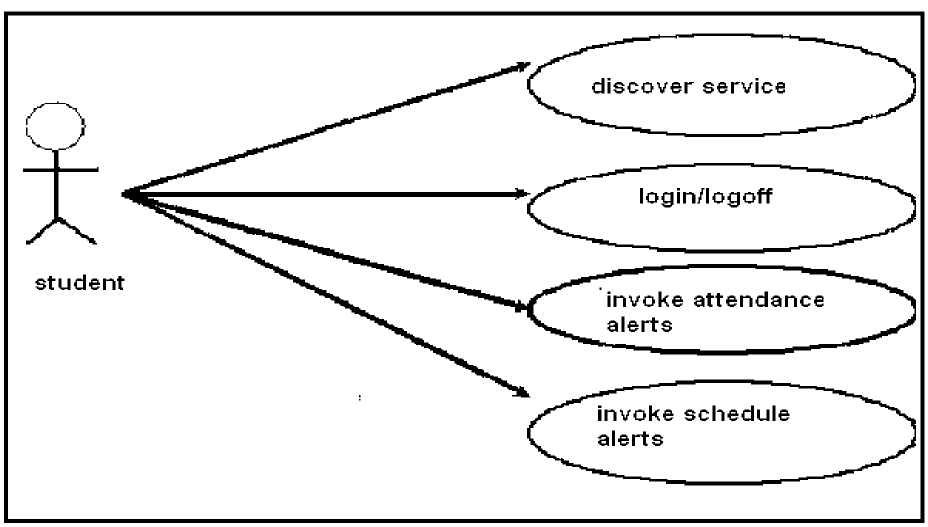

Figure- 3.8: use case diagram

3.8 CLASS DIAGRAMS

This section shows the class diagrams of the server and client end classes.

Some of the important methods of the classes are mentioned and the relationship between the classes is also shown.

3.8.1 SERVER END CLASS DIAGRAM

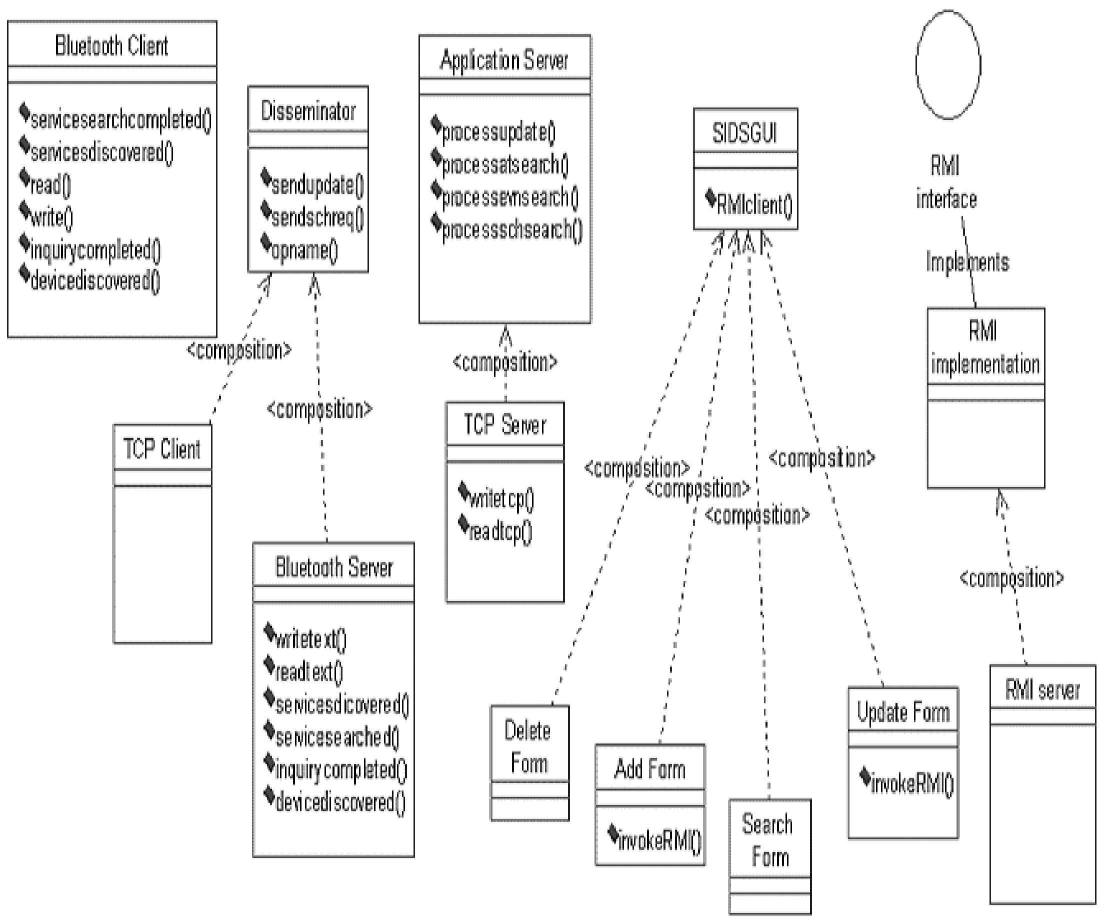

Figure- 3.9: Server end class diagram

The class diagram in figure 3.9 shows the server end classes and their relationships. The classes are divided on the multi tier architecture. The Bluetooth client class resides on the client device. It offers methods for service, device discovery and for reading and writing. The disseminator class has the methods to send requests to the application server and sending updates to the Bluetooth client. It uses the TCP client to communicate with the application server. The application server has the methods to update or search the service database on the basis of attendance, event or schedule.

3.8.2 CLIENT END CLASS DIAGRAM

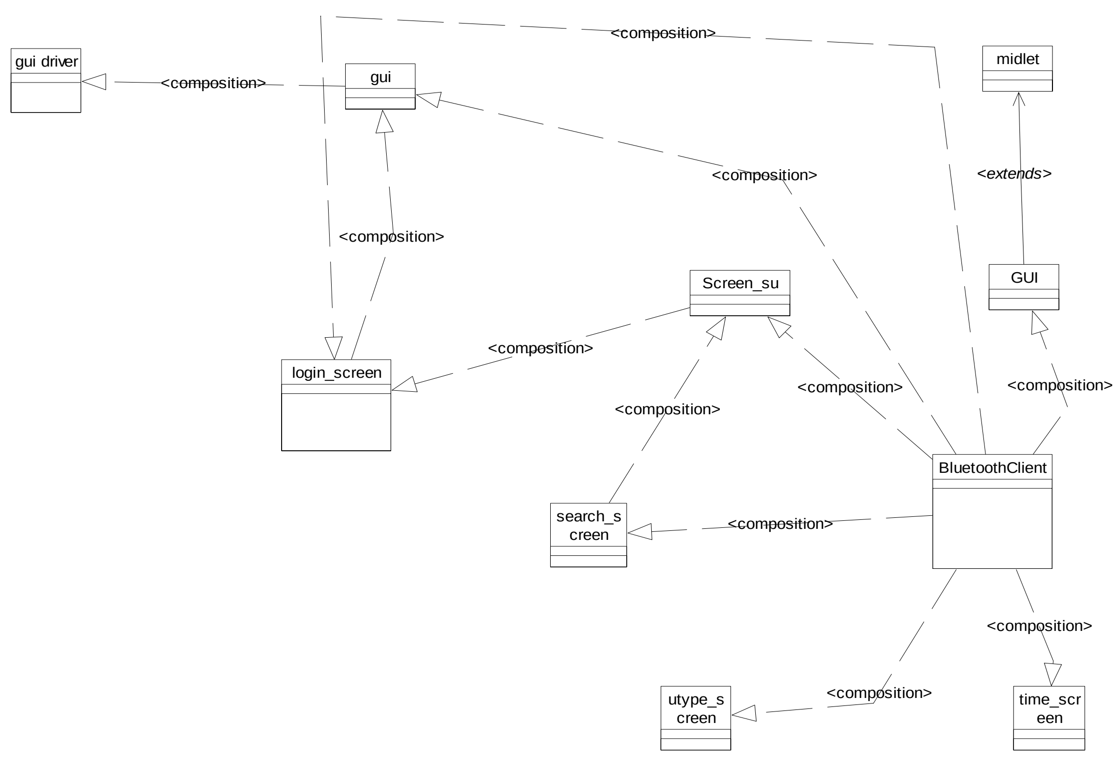

Figure- 3.10: Client end class diagram

The Bluetooth client class has the GUI extended by the midlet class. The GUI has login screen, search screen and the update screen.

3.9 CONCLUSION

This chapter explained the system architecture and design approach used. The architecture is distributed, modular, three tier architecture. The system is flexible because of its modular nature and hence ideal for its utilization in an environment where flexibility in terms of robustness and scalability is required. Wireless link is used between the client and the service end of the system to create utility for the system in terms of providing seamless service to its users.

The system employs Bluetooth associated classes in java that implement the jsr-82 specification of the Bluetooth technology. Bluetooth comes as the most viable option for wireless communication while working in the pervasive domain. Other options such as Infrared [44], RF [45], Home RF [45] and IEEE 802.11 have their constraints such as cost and lack of architecture support among users which prevent employing these options for this system. The mobile application is developed using J2ME and it will run on mobile device having at least 500kb of memory [46].

CHAPTER 4

TOOLS, TECHNOLOGIES AND IMPLEMENTATION

4.1 INTRODUCTION

This chapter covers all the details regarding the implementation of SIDS. Various technologies and tools implemented in the application are discussed here. The overall implementation of SIDS is divided into three phases which includes the implementation of the client end, the disseminator and the application server. The flow and sequence of events followed while implementing SIDS has been stepwise discussed in the chapter.

4.2 TOOLS AND TECHNOLOGIES USED

The core language used for the implementation of the project is JAVA. All the three modules of SIDS are implemented in java. The technology incorporated in the project is that of Bluetooth and the packages of JAVA used to support this technology are J2SE with Bluetooth and J2ME with Bluetooth. The JABWT API for Bluetooth is primarily used for development of SIDS application. The Bluetooth specification is designed to cover a diverse range of devices and usage scenarios. Java's complete platform including all API's and Platform's support is depicted clearly in the figure 4.1:

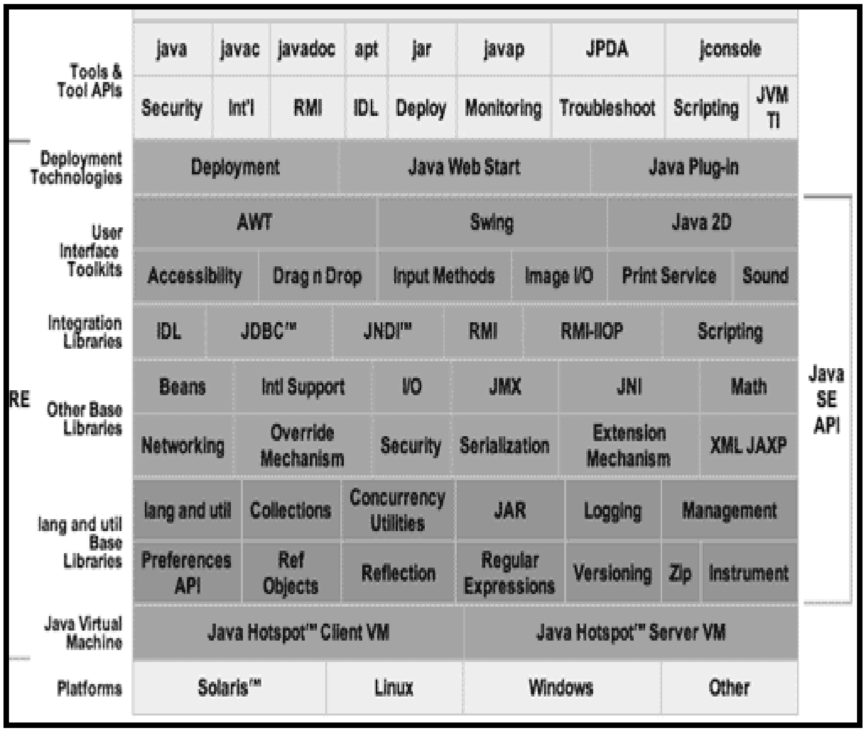

Figure-4.1: JAVA APIs

SIDS has a modular architecture and each module implements different tools, technologies and the libraries. The tools and technologies used in each of the module therefore would be explained with reference to each module in the application.

4.2.1 CLIENT-END APPLICATION

The Client-end of the SIDS application consists of two mediums to communicate with the server. One client-end interface is through the mobile device and the other client end interface is on the PC-client .So, the two sub-modules of the client end application are as under The Midlet and the PC-Client Interface.

4.2.1.1 The Midlet

The java application which runs in the mobile is termed as the midlet and is built on the J2ME platform.J2ME is a set of java API's that run on a JVM with limited resources (also called KVM).J2ME targets devices with resource constraints such as limited power and memory. A J2ME stack has three layers. Two of the most widely used J2ME profile and configuration are the MIDP and CLDC [47]. MIDP 2(based on JSR 118 specification) is an improved version of the MIDP 1.0(based on JSR 37) and provides enhanced UI capabilities, multimedia and network connectivity among other features [48]. MIDP 2 combined with CLDC provides a very efficient ready to use run time environment for devices such as mobile phones, PDA's [49].SIDS client midlet supports MIDP 2 profile it communicates with the disseminator via Bluetooth wireless connection. This communication is done using the JAWBT library. JABWT is based on the JSR 82 specification and is an optional j2me package for Bluetooth communication with handheld devices [50].The Client-end Midlet is implemented using Sun Java(TM) Wireless Toolkit 2.5 for CLDC and JCreator Pro V3.10.008. The Midlet code is written using the JCreator Pro V3.10.008.However, the code compilation, building and debugging is done using the SUN Java ™ Wireless Toolkit 2.5.The Toolkit has an emulator that simulates the working of the midlet and displays a mobile like interface. Figure 4.2 and 4.3 depict the toolkit's snapshot:

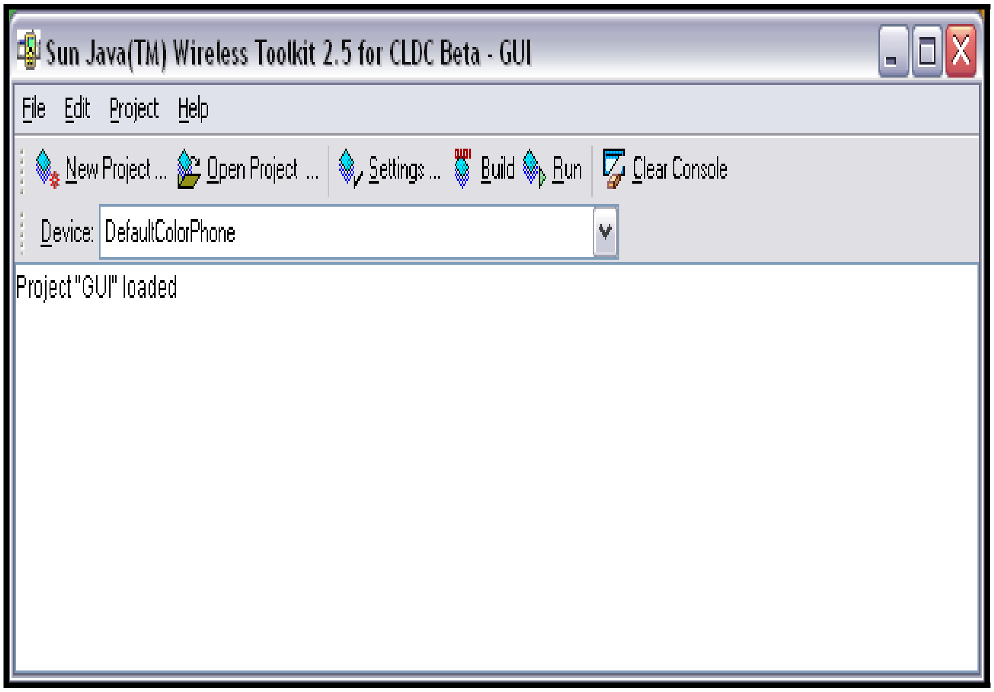

Figure- 4.2: - Sun Java Wireless Toolkit 2.5 for CLDC

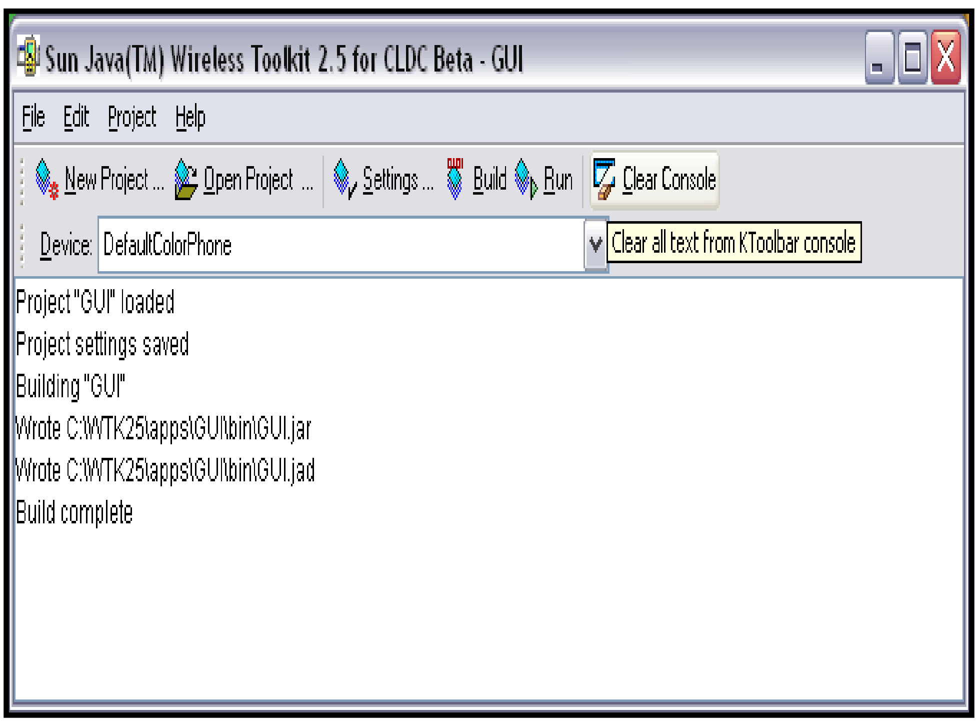

Figure- 4.3: - Sun Java Wireless Toolkit 2.5 for CLDC

4.2.1.2 PC-Client Interface

The Client end Interface on PC is implemented using the JDK 1.5.0, JCreator Pro V3.10.008,Bluetooth Dongle.

Billion ton long range Bluetooth USB Adapter is used as means for conducting the communication between the PC-Client and the disseminator. Mobile devices have in built Bluetooth support but to incorporate the Bluetooth communication on PC we need to have Bluetooth hardware (Bluetooth with J2SE).

4.2.2 DISSEMINATOR

The middleware between the client and the application server is the disseminator. The tools and hardware used to implement the disseminator are JDK 1.5.0, JCreator Pro V3.10.008, Bluetooth Dongle (Billionton long range Bluetooth USB Adapter), and Microsoft Access 2003.

The disseminator has no graphical interface and is operates in a console mode. No direct interaction is implemented for the disseminator, it serves as the disseminator that manages a session with the client, keeps a record of the last latest update received, sends alerts to the client either on request or automatically and also maintains a user authentication database (UAD).UAB is built using the Microsoft Access 2003.The figure 4.4 depicts the table-1 that maintains the authentication database.

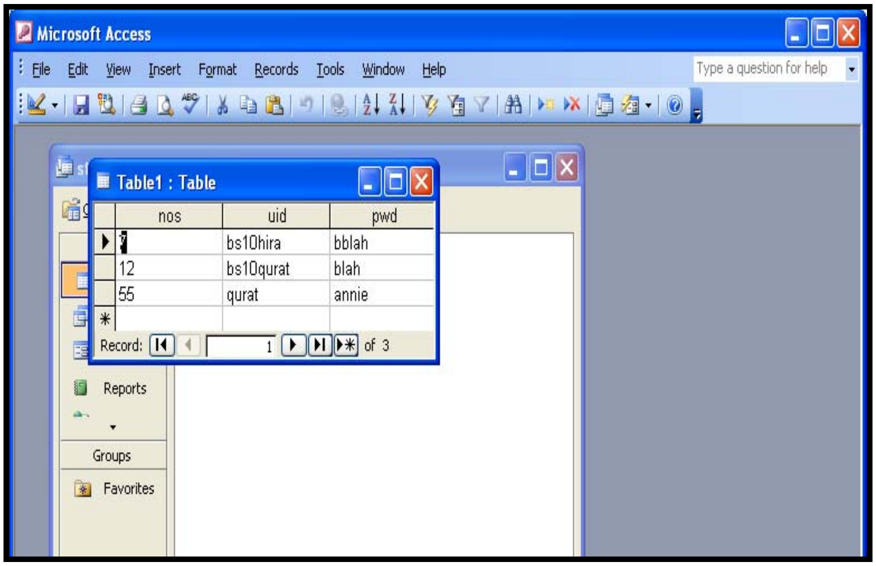

Figure -4.4: Table-1(UAD)

The disseminator is connected with the Application Server via a wired LAN. There exists a TCP Connection between the application server and the disseminator. However the connection with the PC-Client and the Midlet is wireless through Bluetooth dongle.

4.2.3 APPLICATION SERVER

The third module of the project is the Application server that maintains a database.The database contains three tables, these are: Attendance Record Table, Attendance Update Table, and Event/Schedule Update Table. Figure 4.5 depicts the attendance record table and attendance update table.

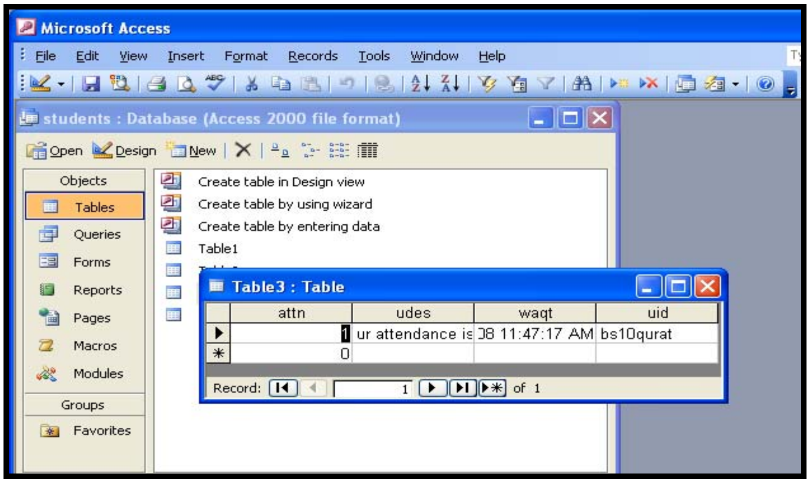

Figure- 4.5: Attendance Update Table

The Application Server provides a direct interface for the administrator to update, enter, delete or search records. The Graphical interface for the administrator is implemented in Java using the tools that are JDK 1.5.0, JCreator Pro V3.10.008.

4.3 IMPLEMENTATION DETAILS

In this section all the implementation details including the flow or sequence of events would be discussed. The tools and technologies used for each of the module have been discussed in the previous section i.e.4.1.

4.3.1 IMPLEMENTATION OF BLUETOOTH APPLICATION

The client-server Bluetooth application is categorized as the Server-end application and the Client-end application. Functionality of each module is as described:

-Client: A client consumes remote services. It first discovers any nearby devices, then for each discovered device it searches for services of interest.

-Server: A server makes services available to clients. It registers them in the Service Discovery Database (SDDB), in effect advertising them. It then waits for incoming connections, accepts them as they come in, and serves the clients that make them.

4.3.1.1 INITIALIZING THE BLUETOOTH APPLICATION

The Bluetooth-enabled application, server or client, first always initializes the Bluetooth stack. First the application retrieves a reference to the Bluetooth Manager from the LocalDevice. Client applications retrieve a reference to the DiscoveryAgent, which provides all the discovery-related services. Server applications make the device discoverable [23]. In the following code snippet, the initialization method initialize () performs initialization of Bluetooth protocol stack:

```
public  void initialize(){
if (!isInit) {
      try {
        conn = null;
        localDevice = LocalDevice.getLocalDevice();
        localDevice.setDiscoverable( DiscoveryAgent.GIAC );
        notifier = (StreamConnectionNotifier)Connector.open(serverUrl,2,true);
        ServiceRecord record = localDevice.getRecord(notifier);
        int annie[]=new int[10];
        annie=record.getAttributeIDs()
 }
      catch (BluetoothStateException e)
        {
              e.printStackTrace();
              System.err.println( "BluetoothStateException: " + e.getMessage() );
        }
      catch (IOException e)
        {
              e.printStackTrace();
              System.err.println( "IOException: " + e.getMessage() );
       }
      isInit=true;
      System.out.println( "nOw wAiTiNg fOr ReSpOnSe........  :)" );
}
}
```

Figure-4.6 : Initializing the Bluetooth Protocol Stack

DiscoveryAgent.GIAC specifies General Inquiry Access Code. No limit is set on how long the device remains in the discoverable mode.

4.3.1.2 ESTABLISHING CONNECTIONS

The Connections established in the application are based on Serial Port Profile (SPP) RFCOMM. The Bluetooth connection is established using the GCF connection factory javax.microedition.io.Connector, passing to its open() method a connection URL argument describing the connection endpoint to create. The URL format for an RFCOMM StreamConnection:

btspp://hostname:[CN | *UUID];parameters*

Where:

Btspp: is the URL scheme for an RFCOMM StreamConnection.

Hostname: is either localhost to set up a server connection or the Bluetooth address to create a client connection.

CN: is the Channel Number value, used by a client connecting to a server - also similar in concept to a TCP/IP port.

UUID: is the Universally Unique Identifier used when setting up a service on a server. Each UUID is guaranteed to be unique across all time and space.

Parameters: include name to describe the service name, and the security parameters authenticate, authorize, and encrypt.

4.3.2 CLIENT-END APPLICATION

The implementation of client end is divided into two sub implementations, including the midlet and the pc-client. Both the midlet and the pc-client have the same implementation of Bluetooth Communication established between the server and the client. The difference between the two client-end applications is that the midlet runs on the mobile whereas the pc-client application runs on the personal computer.

4.3.3 SERVER-END APPLICATION-THE DISSEMINATOR

The Disseminator module responsible for the transfer of messages from the client to the application server and vice versa is implemented in JCreator Pro V3.10.008 using the Bluetooth Protocol Stack Blue cove V2.0.List of classes for the disseminator are BluetoothServer.java Disseminator.Java and TCPClient.java. The figure below shows the connection acceptance at the disseminator and the figure below shows the login parameter being received at the Disseminator.

4.3.3.1 SETTING UP A BLUETOOTH SERVER:

The Bluetooth server i.e. the disseminator is setup by making a service available for consumption. There are four main steps:

-Creating a *service record* for the service you want to make available.

-Adding the new service record to the Service Discovery Database.

-Registering the service.

-Waiting for incoming client connections.

4.3.3.2 CREATING A SERVICE RECORD:

The Bluetooth implementation automatically creates a service record when application creates a connection notifier either a StreamConnectionNotifier or an L2CAPConnectionNotifier.

The Bluetooth service and service attribute has its own Universally Unique Identifier. The UUID class represents short (16- or 32-bit) and long (128-bit) UUIDs. The next snippet in the figure defines and instantiates an RFCOMM connection notifier, resulting in the creation of the service record:

```
...
StreamConnectionNotifier streamConnectionNotifier;
// Create notifier (and service record)
streamConnectionNotifier = (StreamConnectionNotifier)
    Connector.open(connectionURL);
...
```

Figure-4.7 : Service record creation

4.3.3.3 SERVICE REGISTERATION AND WAIT

Invoking the notifier's acceptAndOpen() method causes the Bluetooth implementation to insert the service record for the associated connection into the SDDB, making the service visible to clients. The acceptAndOpen() method then blocks, waiting for incoming connections, which are accepted as they come in. The figure shows a code snippet for registering the service.

```
...
// Insert service record into SDDB and wait for an incoming client
StreamConnection conn =
streamConnectionNotifier.acceptAndOpen();
...
```

Figure-4.8 : Incoming connection wait

When a client connects, acceptAndOpen() returns a connection, a StreamConnection, that represents the client endpoint that the server will read data from. The figure shows the code snippet in which the server waits for and accepts an incoming client connection, then reads from it.

```
...
// Wait for client connection
StreamConnection conn =
streamConnectionNotifier.acceptAndOpen();
// New client connection accepted; get a handle on it
RemoteDevice rd = RemoteDevice.getRemoteDevice(conn);
System.out.println("New client connection... " +
   rd.getFriendlyName(false));
// Read input message, in this example a String
```

Figure- 4.9: Connection Acceptance

4.3.3.4 CLOSING THE CONNECTION AND REMOVING SERVICE RECORD

When the service is no longer needed, it is from the SDDB by closing the connection notifier:

```
...
streamConnectionNotifier.close();
...
```

Figure-4.10 : Closing Connection

4.3.4 APPLICATION SERVER

The third module of the SIDS Application is the Application Server. It's implemented in JCreator Pro V3.10.008 and it also maintains a database as mentioned in section 4.1.3.It provides a graphical interface for the administrator to update, add, delete or search records. Classes used for the Application Server are TCPServer.java, appserver.java, sidsgui.java, addform.java, deleteform.java, updateform.java and searchform.java.

The Application sever is connected with the disseminator through a wired LAN. The connection between the disseminator and the application server is TCP.The application server acts as a TCP-Server whereas the disseminator acts as the TCP-Client. Before starting the Bluetooth communication between the client-end and the disseminator, first the TCP-connection between the TCP-Client and the TCP-Server is established i.e. the disseminator and the application server respectively.

4.4CONCLUSION:

In a nutshell, the chapter has discussed the flow of events and the overall implementation details. The screen shots of each module give an idea of how the actions are performed and what are the action listeners.

CHAPTER 5

FUTURE WORK AND LIMITATIONS

5.1 INTRODUCTION

This chapter takes a look into the innovations and enhancements that can be brought in the present SIDS application and explores the possible constraints on the system put by the technology and design used.

Expanding the scope of an existing project is challenging since it involves rethinking on the design and making it suitable for future use. This may or may not be possible since the existing design may have some limitations when it comes to extending the systems capabilities. SIDS is designed for employment in a department of a university campus. As the audience is a limited to the college faculty and the students of the department, this put a constraint on the size and complexity of the system.

5.2 ENHANCEMENT POSSIBILITIES

The system can be extended to include additional features. Some of the possibilities for enhancement are explored below:

5.2.1 ACCESS BASED FACILITY

The system could be extended to provide access based facility to the students and teachers, delivering confidential notifications to the teachers only. This would also include delivering schedule specific updates to the students of the concerned course only.

5.2.2 EXTENDING THE ATTENDANCE FACILITY

Presently the system implements the attendance based facility for one course only since the object and scope of this project was to introduce an idea, the deployment in a real world scenario would obviously introduce additional considerations like implementing the attendance facility for the students of all the courses.

5.2.3 ENHANCING THE USER END SECURITY

The security of the current system could be enhanced so that in addition to the user authentication on the basis of user id and password only, device based authentication based on Bluetooth device authorization is also incorporated.

5.2.4 INTRODUCING SECURITY FOR THE ADMINISTRATOR INTERFACE

The SIDS administrator interface provided to manipulate the database presently does not include any authorization to log into the database. Administrator authorization based on username and password could be introduced for greater security.

5.2.5 ENCRYPTION OF SENSITIVE DATA

For disseminating confidential notifications for example to the teachers, encryption of the transmitted data can also be considered.

5.3 LIMITATIONS

The SIDS application was intended for small-scale deployment so the underlying technology used for transmission was Bluetooth. Bluetooth comes as the best low cost, short-range interconnection solution. At the same time employing Bluetooth technology introduces certain limitations in the system in terms of expanding the scope in future.

5.3.1 LOW RELIABILITY

Transfer of information using Bluetooth means compromising on reliability since the radio environments are subject to interference because of their dynamic nature. Wireless links are prone to errors caused by interference, or by fading of the signal as mobile devices reach the limit of their Bluetooth range. The Bluetooth base band implements error check on data, but these checks will not catch every single error [51]. For this purpose, greater data integrity checks could be incorporated in the system. It might be a good idea to implement extra error checking on data to be sure any errors that are not caught by the Bluetooth protocol stack are caught at the application level.

5.3.2 MULTIPLE SESSION MANAGERS FOR ENHANCED COVERAGE

Currently the Session Manager offers services to about seven users at a time since the Bluetooth transmitters allows for that many devices to be connected simultaneously [52]. If an increased audience is to be supported by the system, a number of Session Managers need to be distributed over the entire campus.

5.3.3 SLOW RESPONSE TIME

The greater the wireless user traffic, the slower the response time will be for each of them.

5.4 CONCLUSION

This chapter explored some of the limitations of SIDS and the corresponding enhancement possibilities. The limitations of the project mostly resulted from the Bluetooth constraints and the scope of the project which was limited to information dissemination inside the computer sciences department only. The project scope can be increased in the future by offering more information-based services to the users and by targeting a wider audience such as the entire campus. But the most important fact emphasized by the project is the idea that it presents-that of utilizing Bluetooth to achieve information dissemination for a pervasive environment in a campus which can be easily extended to other domains such as the office environment.

BIBLIOGRAPHY

[1] John Thackara, ”The Design Challenge of Pervasive Computing,” May-June 2001

http://www.cs.brown.edu/courses/cs227/2002/pervasive/Thackara.pdf

[2] Tjalling Vonk and Rolof Mulder “Wireless Communication” IICD Research Brief – No 6, March 2003

http://www.ftpiicd.org/files/research/briefs/Brief6.pdf

[3] Dr.-Ing. H. Ritter, “Bluetooth, Bluetooth Basics, Bluetooth and Linux, Bluetooth at AG Tech”, http://www.holtmann.org/lecture/bluetooth/Bluetooth.pdf

[4] Mats Andersson,” Using Bluetooth in an Industrial Reliability and Robustness” Version 1.1 - December 2000

http://www.connectblue.se/fileadmin/Connectblue/Web2006/Documents/White_papers/Bluetooth_Reliability.pdf

[5] “Bluetooth protocol stack core- Bluetooth Protocol Stack Core for integration into new and existing applications

“www.rtx.dk/Admin/Public/DWSDownload.aspx?File=Files%2FFiler%2FPDF%2FBluetooth_Host_Stack_Technical_190506.pdf

[6] Richard Chang and Vitaly Shmatikov, the University of Texas at Austin, “Formal Analysis of Authentication in Bluetooth Device Pairing”

http://www.cs.utexas.edu/~shmat/shmat_fcs07.pdf

[7] W. Mark , “Turning Pervasive Computing into Mediated Spaces”,IBM systems journal, volume 38, Number 4, May 12, 1999 pervasive computing

http://researchweb.watson.ibm.com/journal/sj/384/mark.html

[8] “Pervasive computing”,

http://www.webopedia.com/TERM/P/pervasive_computing.html, Thursday, September 27, 2007

[9] “Mobile and Pervasive Computing Research in the Computer Science Department at Carnegie Mellon”,

http://www.csd.cs.cmu.edu/research/areas/mopercomp/

[10] W. S. Ark ,” A look at human interaction with pervasive computers”

IBM systems journal, volume 38, Number 4, 1999 pervasive computing

http://www.research.ibm.com/journal/sj/384/ark.html

[11] Krishna Akella and Akio Yamashita, “Application Framework for e-business Pervasive computing,

”http://www.ibm.com/developerworks/library/wa-pvc/index.html, Nov 01, 1999.

[12] “Bluetooth Overview,”

http://www.thewirelessdirectory.com/Bluetooth-Overview/Bluetooth-Overview.htm, December 03, 2003

13]”What is Bluetooth Technology?”

http://www.bluetomorrow.com/content/section/10/37/

[14] “How Bluetooth Technology Works”,http://www.mobileinfo.com/Bluetooth/how_works.htm[15]http://developer.apple.com/documentation/DeviceDrivers/Conceptual/Bluetooth/BT_Bluetooth_Basics/chapter_2_section_4.html

[16]http://www.crutchfieldadvisor.com/ISEO-rgbtcspd/learningcenter/home/bluetooth.html?page=2

[17] Eugene Weinstein, Purdy Ho, Bernd Heisele, Tomaso Poggio, Ken Steele1, And Anant Agarwal,” Handheld Face Identification Technology in a Pervasive Computing Environment”
http://cbcl.mit.edu/cbcl/people/heisele/pervasive-2002.pdf

[18] http://en.wikipedia.org/wiki/Project_Oxygen

[19] Robert Grimm New York University ,“One world: Experiences with a Pervasive Computing Architecture”
http://ieeexplore.ieee.org/iel5/7756/29257/01321024.pdf?arnumber=1321024

[20] http://gaia.cs.uiuc.edu/

[21] Hung Q. Ngo, Anjum Shehzad, Kim Anh Pham, Maria Riaz, Saad Liaquat, and S. Y. Lee Computer Engineering Dept. Kyung Hee University, “ Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework”
http://uclab.khu.ac.kr/resources/publication/J_12.pdf

[22] Aekyung Moon, Minyoung Kim, Hyoungsun Kim, Kang-Woo Lee and Hyun Kim Software Robot Research Team, ETRI, “Development of CAMUS based

Context-Awareness for Pervasive Home Environments” International Journal of Smart Home Vol. 1, No. 1, January, 2007

http://www.sersc.org/journals/IJSH/vol1_no1_2007/IJSH-2007-01-01-07.pdf

[23] “Using the Java APIs for Bluetooth, Part 2 - Putting the Core APIs to Work”, By C. Enrique Ortiz, February, 2005,

http://developers.sun.com/mobility/apis/articles/bluetoothcore/

[24] Ken Noblitt, Manager, Technical Marketing North America, Cambridge Silicon Radio, “A Comparison of Bluetooth and IEEE 802.11”, http://www.btdesigner.com/pdfs/KenNoblittComparison.pdf

[25] Surasak Sanguanpong, applied network research group, Department of Computer Engineering, Kasetsart University,”Ethernet”, 11 July 2007

http://www.cpe.ku.ac.th/~nguan/presentations/datacom/ethernet.pdf

[26] Simon Roberts, Colorado Software summit: October 24-29, 2004,”Issues in distributed Architecture”,

http://www.softwaresummit.com/2004/speakers/RobertsDistributedArchitecture.pdf

[27]Professor Ron Sanchez, IMD Lausanne. Interview, Alan McCluskey, Business and media Magazine, ”Modularity: Upgrading to the Next Generation Design Architecture”,

http://www.connected.org/media/modular.html

[28] Beth Simon, Richard Anderson, Steven Wolfman. “Activating Computer Architecture with Classroom Presenter”,

http://www4.ncsu.edu/~efg/wcae/2003/submissions/simon.pdf.

[29] Rob Watt, “C2K lead future”.

http://www.govopps.co.uk/magazine/previous/0709/newhorizons.pdf

[30] Aekyung Moon, Minyoung Kim, Hyoungsun Kim, Kang-Woo Lee and Hyun Kim, "Development of CAMUS Based Context Awareness",http://www.sersc.org/journals/IJSH/vol1_no1_2007/IJSH-2007-01-01-07.pdf

[31] Hung Q. Ngo, Anjum Shehzad, Kim Anh Pham, Maria Riaz, Saad Liaquat, and S. Y. Lee Computer Engineering Dept. Kyung Hee University, " Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework"

http://uclab.khu.ac.kr/resources/publication/J_12.pdf

[32]Satya and Tracy Farbacher, "Project Aura Distraction Free Ubiquitous Computing",http://www.cs.cmu.edu/~aura/research-thrusts.html

[33] University of Illinois, "Gaia Active Spaces for Ubiquitous Computing", http://gaia.cs.uiuc.edu/

[34] Robert Grimm, "One World: Experiences with a Pervasive Computing Architecture",

http://ieeexplore.ieee.org/iel5/7756/29257/01321024.pdf?arnumber=1321024

[35] Eugene Weinstein, Purdy Ho, Bernd Heisele, Tomaso Poggio4, Ken Steele,And Anant Agarwal "Handheld Face Identification Technology in aPervasive Computing Environment",

http://cbcl.mit.edu/cbcl/people/heisele/pervasive-2002.pdf

[36]MIT Project Oxygen, Computer, "MIT-Project Oxygen".

http://oxygen.csail.mit.edu/

[37] Home Page," The Endeavour Expedition: Charting the Fluid Information Utility", http://endeavour.cs.berkeley.edu/proposal/9907IIB32.html[38] Gallaugher, J. & Ramanathan, S. "Choosing Client/Server Architecture, "A Comparison of Two-Tier and Three-Tier Systems." Information Systems Management Magazine 13, 2 (Spring 1996):7-13, http://www.sei.cmu.edu/str/descriptions/threetier.html ,

[39] Robert Chartier,"Application Architecture: An N-Tier Approach - Part 1",http://www.15seconds.com/issue/011023.htm

[40] Fidel CACHEDA, Alberto PAN, Lucia ARDAO, ÁngelVIÑA,Departamento de Electrónica y Sistemas,Facultad de Informática, Universidad de A Coruña

Campus de Elviña s/n, 15071 ACORUÑA, "A Layered Architecture based on Java for Internet and Intranet Information Systems",

http://www.tic.udc.es/~fidel/docs/publications/emmsec99.pdf [41] Axel Rauschmayer,Institut f¨ur Informatik,LMU M¨unchen,ISWC 2005.Galway, Ireland, "RDF as a Data Structure for Software Engineering"

http://www.mel.nist.gov/msid/conferences/SWESE/repository/presentations/axel.pdf

[42] Zhan Cui, Dean Jones and Paul O'Brien, Intelligent Business Systems Research Group, Intelligent Systems Lab, BTexact Technology ,"Issues in Ontology-based Information Integration",

http://www.csd.abdn.ac.uk/~apreece/ebiweb/papers/cui.pdf

[43] "Bluetooth Protocol Stack Core", V0.2 May 2006, RTX Telecom A/S Stroemmen 6 · DK-9400 Noerresundby

www.rtx.dk/Admin/Public/DWSDownload.aspx?File=Files%2FFiler%2FPDF%2F Bluetooth_Host_Stack_Technical_190506.pdf

[44] Patrick Bobbie, Jennifer Uboh, Brian Davis, "A Project in Embedded Systems Design and Development: A Partnership with Area High School Scholars", http://fie.engrng.pitt.edu/fie2004/papers/1676.pdf ,October 20, 2004.

[45] Jim Lansford "Home RF: Bringing Wireless connectivity Home", http://www.ieee802.org/11/Tutorial/90548S-WPAN-HomeRF-Tutorial.pdf , March 9, 1999

[46] Dr. Stephan Rupp, Borja Gomez Zarceno "Mobile Applications with J2ME and JSR82: Bluetooth Enabled Java Applications for Mobile Phones" http://www.linecity.de/INFOTECH_ACS_SS04/acs4_top_1.pdf , July 27, 2004

[47] http://today.java.net/pub/a/today/2005/02/09/j2me1.html

[48]http://java.sun.com/products/midp/

[49]]http://java.sun.com/products/midp/faq.html

[50]http://wireless.klings.org/main.php/J2ME/JABWT

[51] Ryan Wooding, Derek Joos, Trevor Clifton, Charles D. Knutson, "Rapid Heterogeneous Connection Establishment: Accelerating Bluetooth Inquiry Using IrDA", http://faculty.cs.byu.edu/~knutson/publications/IrDA_Assisted_BT_Discovery.pdf, july18, 2006

[52] "Understanding Bluetooth", http://www.hp.com/sbso/wireless/images/understandingBluetooth.pdf, January 2002.