

---

Technical report, June 2010

# **Development of an Electronic Business Card System – The Meetia System**

Bachelor's Thesis in Computer Systems Engineering

Fan Zhang  
Mengting Yang



School of Information Science, Computer and Electrical Engineering, Halmstad University

---

## **DETAILS**

First Name, Surname:

Mengting Yang

Fan Zhang

University: Halmstad University, Sweden

Degree Program: Computer Engineering

Title of Thesis: Electronic Business Card

Academic Supervisor: Wagner Ourique de Morais

## **ACKNOWLEDGEMENT**

First and foremost, we would like to thank to our supervisor Wagner Ourique de Moraes for his valuable guidance and suggestions during the whole time of the thesis work. He motivated us contributed to our project and inspired us greatly.

Furthermore, we would like to thank Anders Nilsson and Henrik Abrahamsson who are business students at Halmstad University for the project idea. They gave us an opportunity to participate and learn from each other.

Finally, we would like to thank you our friends for helping us this project.

## **ABSTRACT**

The aim of this thesis project was to develop a Java ME-based application for smart phones. The application is used to create and exchange business cards, as well as for booking meetings.

Special development questions were put around which software development process, solution APIs and user interface to be used according to the needs. The PIM has been chosen because it provides a clear view of the business card and easy to transfer. The Bluetooth APIs has been chosen because it applies to almost mobile phones and it is enough to solve the transmission.

As a result, the developed application overcomes the problems normally found in PIM and Bluetooth, and provides a more usable and richer user interface.

# CONTENTS

DETAILS.....	I
Acknowledgement.....	II
ABSTRACT.....	III
1. Introduction.....	1
1.1. Background.....	1
1.2. Project Objectives.....	1
1.3. Scenario Description.....	1
1.3.1. Scenario 1: Main Menu.....	2
1.3.2. Create Business Card.....	2
1.3.3. Exchange Business Card.....	3
1.3.4. Book a meeting.....	3
1.4. Outline.....	4
2. Methodology.....	5
2.1. Tools.....	5
2.2. Best Practices.....	5
2.2.1. Power:.....	5
2.2.2. User Interface.....	6
2.2.3. Communications.....	8
2.2.3.1. Infrared.....	8
2.2.3.2. Wi-Fi.....	8
2.2.3.3. Bluetooth.....	8
2.2.3.4. GPRS.....	8
2.2.3.5. 3G.....	8
2.2.3.6. Summary.....	9
2.3. API for Data communication and data storage.....	10
2.3.1. The Bluetooth APIs.....	10
2.3.1.1. RFCOMM:.....	10
2.3.1.2. OBEX:.....	10
2.3.1.3. FileServer.onPut().....	11
2.3.1.4. LocalDevice Class:.....	11
2.3.1.5. SDAP-Service Discovery Application Profile.....	12
2.3.1.6. GOEP-Generic Object Exchange Profile.....	13
2.3.2. Data storage.....	15
2.3.2.1. RMS (Record Management System).....	15
2.3.2.2. PIM (Personal Information Management).....	16
2.3.2.3. RMS v.s. PIM.....	18
3. Meetia System.....	19
3.1. Create Business Card.....	19
3.1.1. Problem.....	19

3.1.2.	Solution.....	20
3.1.2.1.	The type of store.....	20
3.1.2.2.	Interface Design.....	20
3.1.3.	Interface.....	22
3.2.	Exchange Business Card.....	24
3.2.1.	Problem.....	24
3.2.2.	Solution.....	24
3.2.2.1.	Link.....	24
3.2.2.2.	Send information.....	24
3.2.2.3.	Send an object.....	25
3.2.3.	Interface.....	25
3.3.	Book a meeting.....	28
3.3.1.	Problem.....	28
3.3.2.	Solution.....	28
3.3.2.1.	Create date and time.....	28
3.3.2.2.	Save it as a vCalendar.....	29
3.3.3.	Interface.....	29
4.	Results.....	31
5.	Conclusion.....	32
	References:.....	33

## LIST OF FIGURES

Figure 1. Main Menu of Meeting.....	2
Figure 2. Create Business Card.....	2
Figure 3. Exchange Business Card.....	3
Figure 4. Book a meeting.....	3
Figure 5. User Interface 1.....	6
Figure 6. User Interface 2.....	7
Figure 7. User Interface 3.....	7
Figure 8. Protocol model.....	11
Figure 9. A typical service discovery scenario.....	12
Figure 10. SDAP stack.....	13
Figure 11. Bluetooth Profile.....	14
Figure 12. Example of configurations.....	15
Figure 13. Overview of J2ME RMS and MIDlet interfacing.....	16
Figure 14. the interface of Main Screen.....	19
Figure 15. The empty form of Create Business Card.....	22
Figure 16. The finished form of Create Business Card.....	23
Figure 17. The save successful tip.....	23
Figure 18. The list of business cards.....	25
Figure 19. The detail of business card.....	26
Figure 20. The open tip of server Bluetooth connection.....	26
Figure 21. The interface of ready for send.....	27
Figure 22. The connection tip of client.....	27
Figure 23. The add tip.....	28
Figure 24. The interface of Choose date.....	29
Figure 25. the interface of Choose time.....	30
Figure 26. The interface after add succeed.....	30

## **LIST OF TABLES**

Table 1. Comparison of different communication technologies commonly found on smart phones.....	9
Table 2. Local Device Methods.....	12
Table 3. The class/interface of package “javax.microedition.pim”.....	16
Table 4. Public interface Contact extends PIMItem.....	17
Table 5. vCard & vCalendar.....	18



# **1. INTRODUCTION**

## ***1.1. Background***

Meetia is a software application designed for smart phones to simplify both the exchange of business cards and event booking. The program offers to the user a template to create business cards and a method to exchange business cards. Users can use the system to book meetings or appointments with their contacts.

The idea for this thesis project came from Anders Nilsson and Henrik Abrahamsson who are business students at Halmstad University. Anders and Henrik want to provide to business people an easy and innovative way for creating and exchanging business cards, as well as booking meetings using mobile phones.

For this purpose Anders and Henrik interview some business people from different areas to check which is the most popular mobile phone that they are using and how business people feel about the design of the mobile application.

According to their results, the most and widely used mobile phones are from Nokia and Sony Ericsson, however, another brands are used too, such as Blackberry and Motorola. As they Anders and Henrik want to provide an software application which will run in a mobile phone, this application must be developed using platform commonly supported by Nokia, Sony Ericsson, Blackberry, Motorola, Siemens, Samsung, among others [1]. Java Micro Edition (Java Me) [2] is a platform supported by all of these mobile telephone manufacturers and supported development of rich user interfaces, robust and security applications, networked and offline applications that can be downloaded dynamically. The main advantage is that Java Me based applications are portable across many devices.

## ***1.2. Project Objectives***

The objective of this project is to develop an application for java enabled smart phones. Such application is called Meetia. Users should be able to:

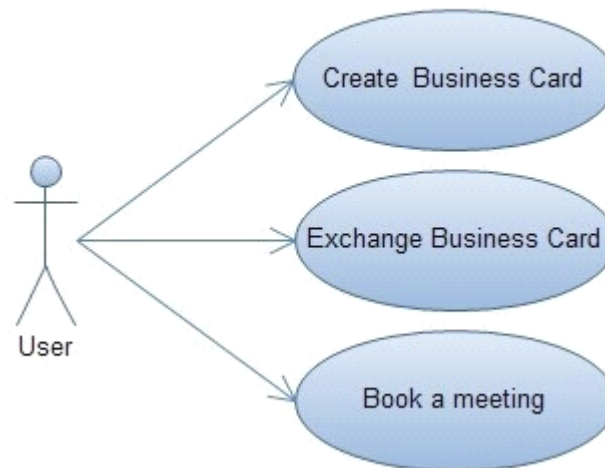
- Create business cards
- Exchange business cards: Send and receive
- Book meetings: Set time, date and members
- Configure the application: Configure calendar

## ***1.3. Scenario Description***

In order to develop this application, the main application requirements were described by different usage scenarios. According to the scenario description, the way of implementing will be clear and direct.

### ***1.3.1. SCENARIO 1: MAIN MENU***

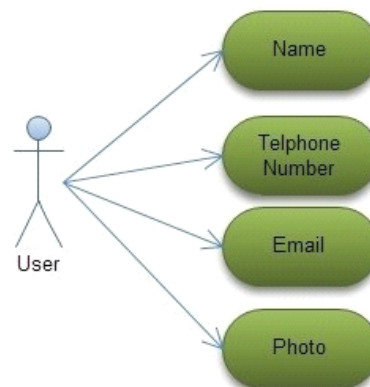
When the user executed the Meetia application, the application will show the main menu which will allow the user to select three functions: “Create Business card”, “Exchange Business Card”, and “Book a meeting”. Each function will be displayed as a button on the screen. This scenario is presented at Figure 1.



**Figure 1. Main Menu of Meeting**

### ***1.3.2. CREATE BUSINESS CARD***

When the user selects “Create Business Card”, the applications will show a list of possible personal information which the user can enter, such as name, telephone number, email, address, etc. The user might select and enter different personal data. Some are optional and others mandatory, such as Name and Email or Telephone. The user will select the option “Save” and then a business card will be created based on the entered user data. The user-case for this scenario is presented in Figure 2.



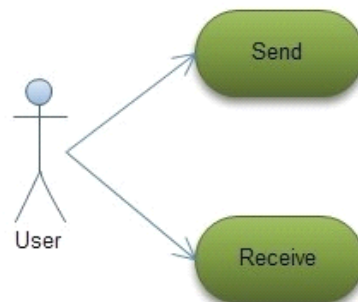
**Figure 2. Create Business Card**

### **1.3.3.**

### **EXCHANGE BUSINESS CARD**

When users choose the function of 'Exchange Business Card', it will be shown two options which are 'send' and 'receive'. This scenario is presented in Figure 3.

At first, the application will hint users when they use the 'Exchange Business Card' function about the automatic Bluetooth activation and deactivation, which is performed and used while searching for other users who also have the Meetia application in the "Receive" mode. After that, the users can choose the member and type the code to finish the exchange. At last, the business card which the user received will be synchronized with the Contact List automatically.



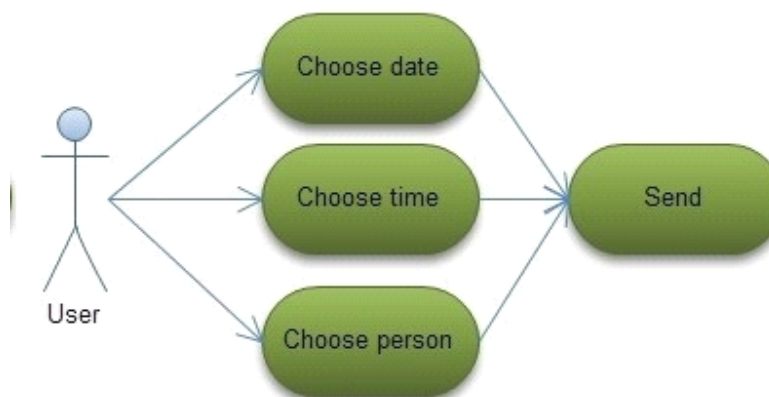
**Figure 3. Exchange Business Card**

### **1.3.4.**

### **BOOK A MEETING**

When the users choose the function of 'Book a meeting', it will show the option of 'Choose data', 'Choose time', 'Choose person'.

The users can choose data and time which they might available or not and after that they can choose the member who will attend the meeting and send them the information about the meeting and wait for the confirmation. This scenario is presented in Figure 4.



**Figure 4. Book a meeting**

### ***1.4. Outline***

This report is organized as follows:

- Chapter 2 presents the methods and tools that are used while developing the project.
- Chapter 3 describes the project implementation.
- Chapter 4 presents the results and explains whether the solution that is solved the problems.
- Chapter 5 presents the project conclusion.

## **2. METHODOLOGY**

Applications designed for mobile devices are different from desktop applications. Mobile devices are battery-operated with restricted processing and storage capabilities. The interface with the user is also limited due to small displays and keys in the keypad, if present. The tools necessary to develop mobile applications are also different.

Thus, different constraints must be addressed while developing mobile applications such as the development platform and tools, power consumption, usability communication capabilities and data storage [3].

### ***2.1. Tools***

As mentioned in Chapter 1, Java Micro Edition (Java Me) has been chosen as the development platform. Java Me is a java platform designed for mobile devices and embedded systems and is supported by the main mobile telephone manufacturers.

In order to develop MIDlets, it is necessary the Sun Java Wireless Toolkit [4]. The Sun Java Wireless Toolkit is based on Java Me's Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP) [4]. It includes many features such as MIDlet signing, certificate management, integrated over-the-air (OTA) emulation, push registry emulation and more [4].

The Sun Java Wireless Toolkit 2.5.2 has been used in the project and includes a set of tools to build and test MIDP applications and can be integrated to different Java integrated development environments (IDE), such as Eclipse or Net Beans. The choice for a Java IDE is optional and complete MIDP applications can be developed by coding in normal text editors. However, EclipseMe 1.7.9 has been chosen as the Java IDE due to the author's familiarity with the tool.

### ***2.2. Best Practices***

As mentioned at the beginning of this Chapter, it is necessary to understand the nature of mobile devices and some considerations which have to be taken into account while developing mobile applications. Mobile applications must address requirements in terms of power, user interfaces, communication capabilities and data storage.

#### ***2.2.1. POWER:***

With the portability of most mobile devices, power saving is very important when using some high power consumption function such as Bluetooth, Wi-Fi, 3G, and GPRS.

Using Bluetooth might increase power consumption and shorten battery life because in mobile phones battery capacity is constrained.

For considering the power saving, this application will be used Bluetooth only when the users choose the function of 'Exchange the Business Card'.

### **2.2.2. USER INTERFACE**

The design of user interfaces for mobile devices is constrained by different display sizes and resolutions. Depending on this design, some user interfaces might be more efficient and intuitive than others. To demonstrate that, 3 different user interface designs are presented. All these 3 user interfaces have the same functionality which is to allow users to input personal information into the system.

The first example is presented in Figure 5. In this interface, users must select “Options” in order to choose which personal information they want to inform. This information might be email, address, and telephone number, among others. After entering all desired information, users need to hit the “Options” button to store (“Save”) or to visualize the information (“Show vCard”).

Such user interface is not intuitive and quite cumbersome because the process demands prior knowledge about the application and many selections to choose which information they want to enter. Even if this design is easy to implement, it is not suitable for the envisioned Meetia application.

Figure 5 displays four wireframe screens for a "Create Business Card" application, illustrating a menu-driven user interface design. Each screen has a title bar "Create Business Card".

- Screen 1:** Features a "Name" label and an input field. At the bottom are "Back" and "Options" buttons.
- Screen 2:** Features a "Name" label and an input field. A menu box is overlaid on the right, containing the options: "1.Add Form", "2.Save", and "3.Show vCard". At the bottom are "Back" and "Options" buttons.
- Screen 3:** Features five input fields labeled "Mobilphone number", "Email", "Address", "Photo", and "Telephone number". At the bottom are "Back" and "Add" buttons.
- Screen 4:** Features two input fields labeled "Name" and "Mobilphone number". At the bottom are "Back" and "Options" buttons.

**Figure 5. User Interface 1**

Figure 6 presents a second design example. In this interface, it shows all the personal information which the users could inform. But each personal information option occupies 2 lines, so the screen shows the personal information forms are limited. If the users want to inform all the personal information, they must scroll up or down the screen to check the other personal information forms. After entering all the desired information, users could hit the “Options” button to store or to visualize the information.

**Create Business Card**

**Name**

**Mobilphone number**

**Email**

Back Options

**Figure 6. User Interface 2**

The third and last design example is presented in Figure 7. In this interface, each form of personal information only occupies one line, so in this way, it can save space to show more personal information on the screen. When the user hit the “Name” or other lists of personal information, the line will be null to let the user fill in the information what they want to inform.

**Create Business Card**

*Name*

*Mobilphone number*

*Email*

*Address*

Back Options

**Figure 7. User Interface 3**

The second example, presented in Figure 6, is simpler and more intuitive than the first example, illustrated in Figure 5. But users must scroll up or down to check the other personal information. In this way, it is inconvenient for the users to use.

For the convenient usage and succinct style, the user interface 3 is the best among the 3 user interface designs presented. It includes more personal information forms than the other two if it shows in the same size of a display. And also it is more intuitive for the user to inform the personal information.

### **2.2.3. COMMUNICATIONS**

Nowadays, smart phones integrate different data communication technologies, such as Infrared, Wi-Fi, Bluetooth, GPRS, and 3G.

Each data communication technology uses different protocols and methods to transmit information. And the information with different type and size should be chosen certain suitable communication technology. For example, if the user wants to transmit a picture in the mobile phone with other users, the most convenient way is to use Bluetooth. So compared with the wireless bandwidth, coverage, popularity and consideration of the Meetia application, Bluetooth is the best way for the function 'Exchange Business Card'.

#### **2.2.3.1. INFRARED**

Infrared radiation (IR) data transmission often used in a short-range communication among mobile phones which usually accept standards published by IrDA, the Infrared Data Association [5]. It is quite reliable because it is one-to-one connection but that means a device can not send to multiple receivers at the same time. And the most inconvenient part of it is that when using infrared a sender must assign with its receiver.

#### **2.2.3.2. WI-FI**

Wi-Fi (Wireless Fidelity) is a wireless Ethernet based on the IEEE802.11 standards provides a replacement of wire networks and it usually used in a large-range communication and connect with large device with high power such as desktops and laptops. It is more adapted for huge data transmission.

#### **2.2.3.3. BLUETOOTH**

Bluetooth is a short distance using proprietary and open short-long radio waves (wireless technology standards for sharing data) from fixed and mobile devices create personal area networks and security but also for low-power design of high-level [6].

#### **2.2.3.4. GPRS**

GPRS (General Packet Radio Service) is the 2G cellular communication systems for mobile communications [7]. GPRS use traditional circuit switching for data communication which is billed with per minute of connection time [7].

#### **2.2.3.5. 3G**

3G is International Mobile Telecommunications-2000 (IMT-2000), a wide-area wireless in mobile environment. 3G is faster than 2G and 2.5G in both of data services and data rates [8].



### 2.2.3.6. SUMMARY

A summary of different data communication technologies are presented in Table 1.

	<b>Infrared</b>	<b>Wi-Fi</b>	<b>Bluetooth</b>	<b>GPRS</b>	<b>3G</b>
<b>Data Transmission Speed</b>	0.072 Mbit/s	11Mbps (although actual speed depends on the connections to one access point.)	721 Kbit/s	56-114 Kbit/s	14.0 Mbit/s on the downlink and 5.8 Mbit/s on the uplink.
<b>Range</b>	Standard: 1 meter; Low power to low power: 0.2m; Standard to low power: 0.3m	Range of 300 - 400 meters	Ranges between 50-250 meters. Depending on the Bluetooth class Type	Depends on the signal of the coverage of the area.	Depends on the signal of the coverage of the area.
<b>Popularity</b>	Only few mobile phones use	Spreading very fast	Adopted Worldwide	Adopted Worldwide, but cost money	New technology, but cost money
<b>Security</b>	High reliability	64 bit encryption	128 bit encryption		

**Table 1. Comparison of different communication technologies commonly found on smart phones**

For the Meetia application, the Bluetooth technology is the best choice because it provides an easy and efficient one-to-many connection, is commonly found in smart phones and designed for low battery consumption [9].

When using Infrared to transmit data, it only allow one-to-one connection and a sender must align with a receiver although it provided high reliability. GPRS is not good enough than Bluetooth because it cost money and the speed of data transmission is not faster than Bluetooth. Although 3G is faster than Bluetooth in many ways, it will cost money from the users. So Bluetooth is the best way for data communication technology to use in the function of “Exchange Business Card”.

### ***2.3.API for Data communication and data storage***

The Meetia application allows users to create business cards, exchange business cards and book meetings using their mobile phones. To do that, the application must store the users’

information into the phone. This stored information, such as an electronic business card or a meeting entry, might be exchanged with other users who also have the Meetia application. In order to be able to perform such operations, data communication and storage APIs are necessary.

### **2.3.1. THE BLUETOOTH APIS**

The Java APIs for Bluetooth provide a set of APIs to access and control a Bluetooth-Enabled device [5].

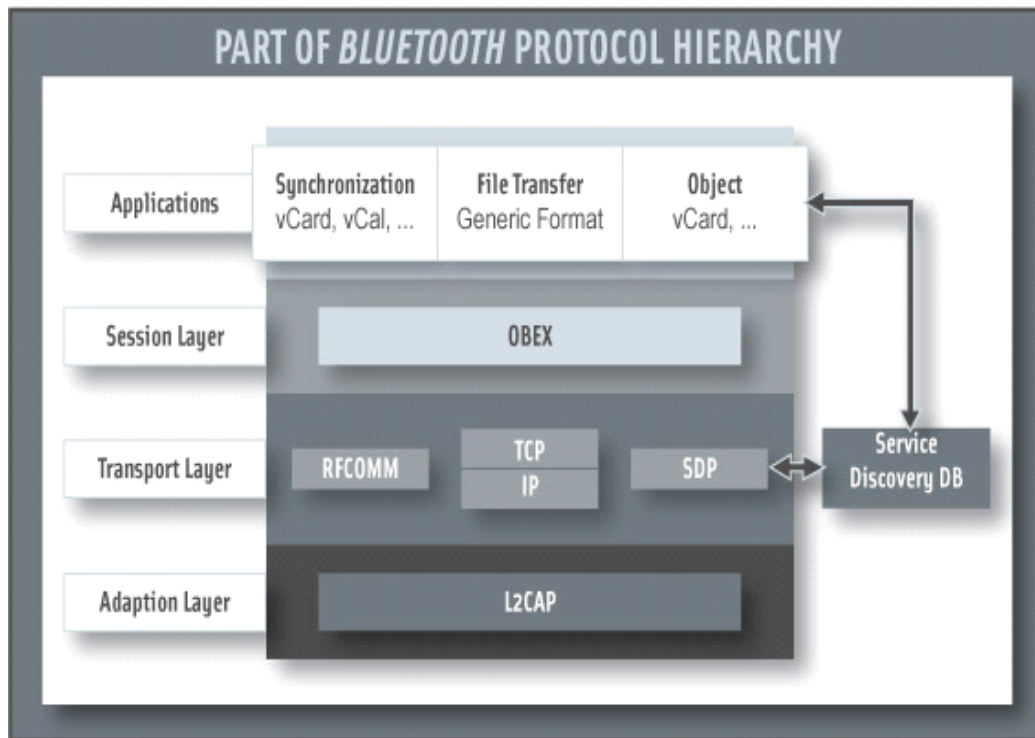
The specification was developed under the Java Community Process which is JSR 82 [10]. There are many protocols of it. SDAP is the service discovery application protocol. JSR 82 can send data with some protocols such as RFCOMM or OBEX. Based on the OBEX, there is another method for transmission, it is GOEP. And inherited from GEOP, there are three different profiles using in the models: Object Push Profile, File Transfer Profile, and Synchronization Profile.

#### **2.3.1.1. RFCOMM:**

Send data byte by byte. By sending a long string data, the loop should be used. If a streaming data will be sent, RFCOMM is the better choice, the same as using a traditional serial port. In the real world, if the traditional serial applications be used and want to use Bluetooth to transfer, the RFCOMM should be used. Bluetooth devices if a simple text string (for example, chat applications) want to be sent, then probably not much advantage to use OBEX. In this case, use RFCOMM or L2CAP.

#### **2.3.1.2. OBEX:**

OBEX can send data as an object. When an object data (such as files) to be sent, the most appropriate choice is OBEX. Not only can send data using OBEX, but also to send on the load of the context or metadata. For example, to send files using OBEX also able to send on the file of other useful information, such as file name, file type, file size or any other description of the file contents. The part of Bluetooth protocol hierarchy and transport layer of service discovery DB are presented in Figure 8.



**Figure 8. Protocol model**

#### **2.3.1.3. FILESERVER.ONPUT()**

The OBEX server will accept a file from a remote Bluetooth device. The file name of the transferred file should be noticed, the file name is used when instantiating the File object.

The Bluetooth Protocols/Profiles of Java APIs which is described in the JSR-82 interface as following:

- Identify and check the Bluetooth devices
- Find Bluetooth devices communicate within the property of their equipment
- Search Services remote Bluetooth device
- Create Bluetooth client application with a remote Bluetooth server
- Create Bluetooth server application, that can requests from Bluetooth clients

#### **2.3.1.4. LOCALDEVICE CLASS:**

Class LocalDevice is the local Bluetooth devices. The actual Bluetooth chip Bluetooth device can update the built-in Bluetooth-enabled laptops and mobile phones, may also be a USB or UART interface to the computer. Whatever the shape or form of equipment uses methods which can be manipulated class LocalDevice (Table 2).

Local Device Methods	Description
<b>.getBluetoothAddress()</b>	Return the Bluetooth device address
<b>.getDeviceClass()</b>	Return the Bluetooth device class
<b>.getFriendlyName()</b>	Return the friendly name of the device, Bluetooth device name is usually the user to set a Bluetooth control center
<b>.getRecord()</b>	Return a specified record of Bluetooth connection service
<b>.getDiscoverable()</b>	Return the device status which can be found
<b>.setDiscoverable()</b>	Set the device status which can be found
<b>.getProperty()</b>	Returns a property of Bluetooth device

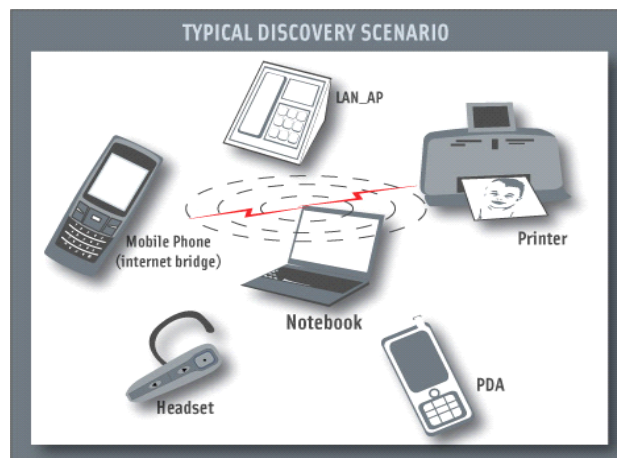
**Table 2. Local Device Methods**

#### **2.3.1.5. SDAP-SERVICE DISCOVERY APPLICATION PROFILE**

SDAP describes how an application should use SDP to discover services on a remote device [11].

Use this function the servicer can search around and find the SDP client. Then get the information of the client from Service records and send the SDP request.

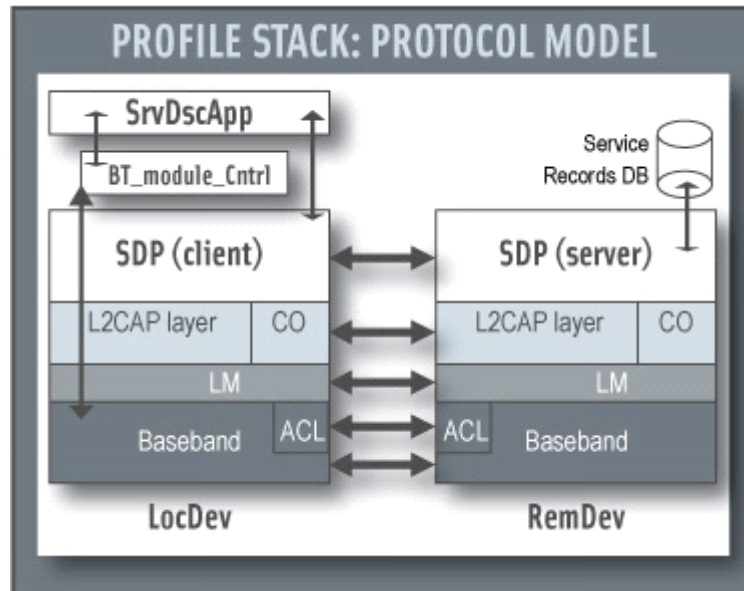
The Figure 9 shows the notebook (a local device) inquiring for services among any other of remote devices [11], such as mobile phones, printers, PDAs, headsets and so on.



**Figure 9. A typical service discovery scenario**

The SDAP defines two roles, one is Local Devive (LocDev) and other is Remote Device(s) (RemDev(s)) [11].

The Figure 10 shows SDAP profile stack of the protocol model. The Bluetooth SDP client with the service discovery user application (SrvDsc) in a local device (LocDev) interfaces can be sent service inquiries and receive service inquiry responses from the SDP servers of remote devices (RemDevs) [11].

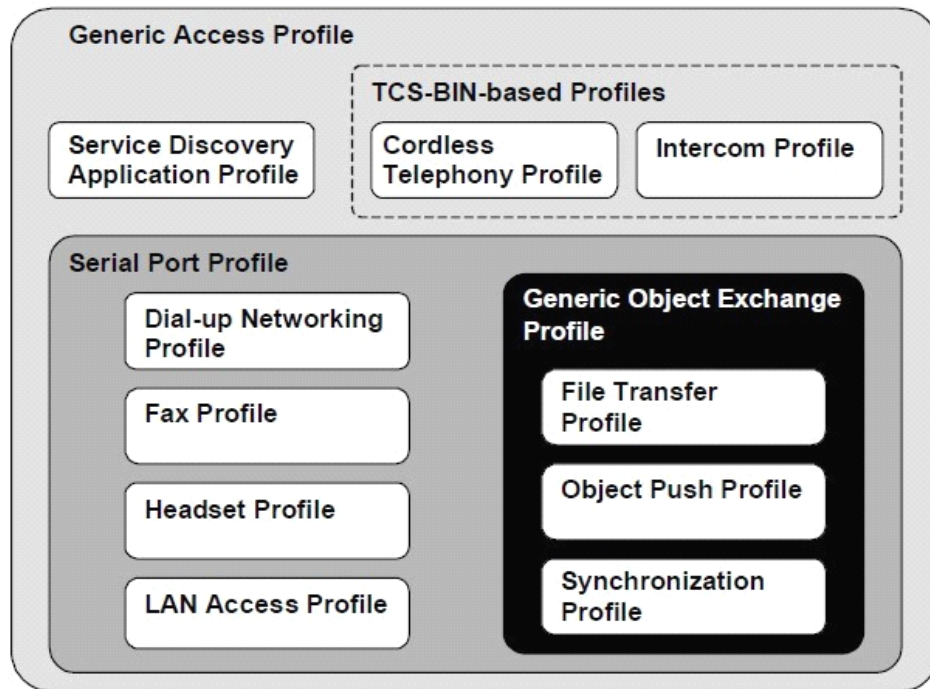


**Figure 10. SDAP stack**

#### **2.3.1.6. GOEP-GENERIC OBJECT EXCHANGE PROFILE**

The Generic Object Exchange profile defines the protocols and procedures that shall be used by the applications providing the usage models, such as Synchronization, File Transfer, Object Push model, which need the object exchange capabilities in many common devices, for example notebook PCs, PDAs, smart phones and mobile phones [14].

Synchronization Profile, Object Push Profile, and File Transfer Profile are all the specific application examples of protocol subset inherited from the GOEP. There is the same application model for Bluetooth protocol stack in all of them which presented in Figure 11.



**Figure 11. Bluetooth Profile**

### **Object Push Profile:**

It basically defines a one-dimensional object transmission. The Object Push Profile's theoretical foundation is the usage model of exchanging business card. Object Push Profile defines three functions: Object Push function, Business Card Pull function, Business Card Exchange function [15]. Object Push function is the most basic operation among these functions, and also is the essential function. Business Card Pull function is an optional function, it can output all of the contacts and also input any contact.

The object push profile defines two roles, a Push Server and a Push Client [15].

### **File Transfer Profile:**

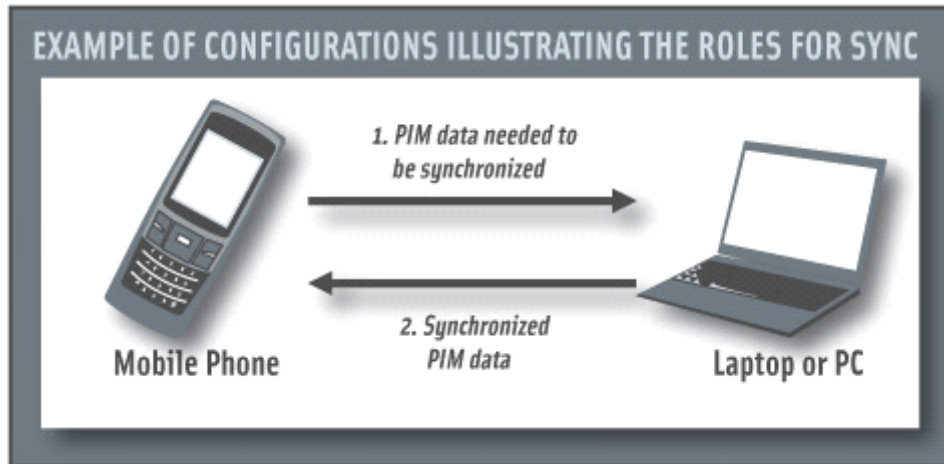
This profile defines how folders and files on a server device can be browsed by a client device, for example, one PC browsing the file system of another PC [16]. In this project, this profile is not necessary.

The file transfer profile defines two roles, that of a Client and a Server [16].

### **Synchronization Profile:**

The synchronization profile is used in the conjunction with GOEP to enable synchronization of calendar and address information (personal information manager [PIM] items) between Bluetooth enabled devices [17]. In this thesis, this profile can be solved the synchronization's problem.

The synchronization profile defines two roles, IrMC Server and IrMC Client [17]. The example of configuration for the roles of synchronization profile is presented in Figure 12. It shows the transmission and synchronization of a mobile phone and laptop or PC with PIM data.



**Figure 12. Example of configurations**

### ***2.3.2. DATA STORAGE***

There are two different ways of data storage of mobile applications, one is RMS, and another is PIM.

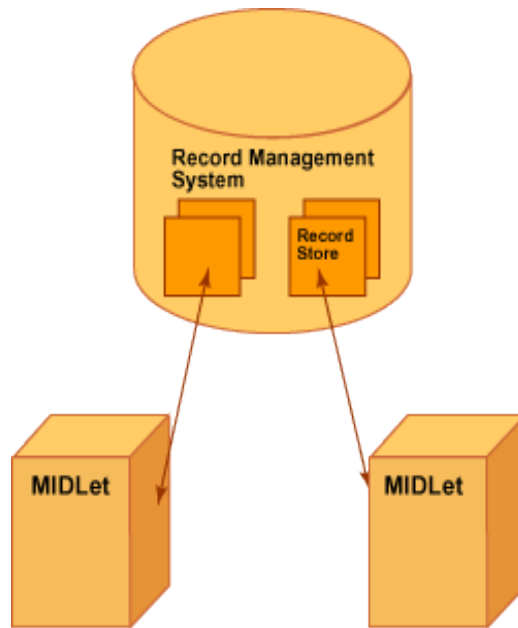
RMS that contains multiple record stores can store data persistently and retrieve it with using MIDlets in J2ME [13]. Figure 13 is an overview of J2ME RMS and MIDlets interfacing.

PIM can be used with the mobile phone that supports JSR 75 (PDA Optional Packages of the J2ME Platform).

#### ***2.3.2.1. RMS (RECORD MANAGEMENT SYSTEM)***

Because of the lack of many small computing devices to access the disk drives and network database server or file server, it is in the J2SE and J2EE applications using the persistence of the typical resources. Thus, J2ME application must store information in nonvolatile memory using the records management system (RMS).

RMS is an application programming interface that usually can store and manipulate data in small computing devices. It use both file system and database management system to store the data in it like a table of database. We can build a structure to store some information in one space. RMS can identify the row in the record store, which is called the record ID. It is considered the primary key of the record store. And we can do some operation by using the record ID which presented in Figure 13.



**Figure 13. Overview of J2ME RMS and MIDlet interfacing**

#### **2.3.2.2. PIM (PERSONAL INFORMATION MANAGEMENT)**

The PIM API can provide MIDP mobile phone or PDA interface and class that contains the application access the database in order to read and modify other applications and stored into the database, and then by other applications to access your PDA information data.

PIM include Contact, Event, ToDoList. And their interface is ContactList, EventList , and ToDoList. And the The PIM APIs are defined in the package “javax.microedition.io.pim”. Here are some descriptions about PIM class and interface (Table 3).

<b>Class/Interface</b>	<b>Description</b>
<b>PIM</b>	Provide access devices for personal information management method of the list
<b>PIMItem</b>	Personal information that item, as a project entity (contact, event reminders, etc.)
<b>PIMList</b>	Personal information management project, as a list of items for the collection of personal information management
<b>Contact</b>	Contact item of information, including all contact-defined fields
<b>ContactList</b>	Contact list of information items
<b>Event</b>	Event item of information, including all contact-defined fields
<b>EventList</b>	Event list of information items
<b>ToDo</b>	To-do item of information, including all the to-do-defined fields
<b>ToDoList</b>	To-do list of information items

**Table 3. The class/interface of package “javax.microedition.pim”**



ContactList, EventList, and ToDoList can open with three modes: read-only, write-only, or read/write. And this API include many useful function that we can used directly, like Contact createContact() and so on.

In ContactList, the different information can be store in different field(Table 4).

Field	Description
<b>Contact.ADDR</b>	Field for an address for a Contact
<b>Contact.EMAIL</b>	Field for email address
<b>Contact.NAME</b>	Field for a contact name
<b>Contact.PHOTO</b>	Field for a photo of a Contact
<b>Contact.TEL</b>	Field for a telephone number

**Table 4. Public interface Contact extends PIMItem**

### **vCard & vCalendar**

vCard and vCalendar are two file format in PIM. They are used for Personal Data Interchange ( PDI ) which occurs every time two or more individuals communicate, in either a business or personal context, face-to-face, or across space and time [18]. Both vCard and vCalendar are defined by accepted international and industry standards [18].

vCard is a file format standards for electronic business cards. It can be used for exchange. vCard can contain information are: name, address information, phone numbers, URL, logo, photos.

vCalendar is a data format for exchange calendar and scheduling. It could include one or more vCalendar objects. It can do spme operations about event and "to-do" items in PIM.

Here is some compare about their format (Table 5).

<b>vCard/vCalendar</b>	<b>Format</b>
<b>vCard</b>	BEGIN:VCARD VERSION:2.1 N:Gump;Forrest FN:Forrest Gump TEL;WORK;VOICE:(111) 555-1212 TEL;HOME;VOICE:(404) 555-1212 EMAIL;PREF;INTERNET:forrestgump@walladalla.com REV:20080424T195243Z END:VCARD
<b>vCalendar</b>	BEGIN:VCALENDAR VERSION:1.0 BEGIN:VEVENT CATEGORIES:MEETING STATUS:TENTATIVE DTSTART:19960401T033000Z DTEND:19960401T043000Z SUMMARY:Your Proposal Review DESCRIPTION:Steve and John to review newest proposal material CLASS:PRIVATE END:VEVENT END:VCALENDAR

**Table 5. vCard & vCalendar**

### **2.3.2.3. RMS V.S. PIM**

In summary, PIM presents the most suitable choice for the Meetia application because the data can be imported or exported in the mobile phones and store in the local device or SIM. The main reasons for choosing are:

- PIM can store and retrieve data like an Address Book, Calendar and Contact List. But RMS only can store data as a general database. Contact, Calendar is a single entity in PIM, and can store them as a vCard or vCalendar, but it just can store as a byte array in RMS.
- RMS only can access in local device and does not have the capability to export or import. But PIM can export and import from external sources, like send or receive vCard or vCalendar then store it in phone.
- The PIM is more flexible than RMS in data explanation.

### 3. MEETIA SYSTEM

This application has three functions: create business card, exchange business card and book a meeting (Figure 14).

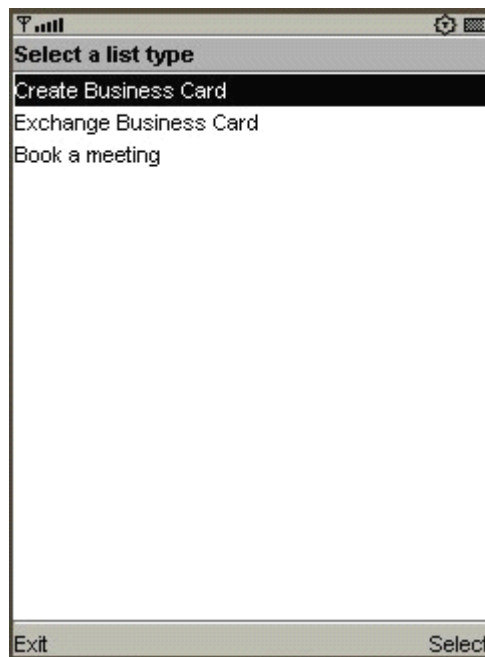


Figure 14. the interface of Main Screen

#### ***3.1. Create Business Card***

In this function, the business card can be created. It includes some personal information such as: name, phone number, email and so on. And then the business card should be stored in the user's mobile phone.

Although the function of 'Create Business card' is quite simple in the development of the Meetia application, the problems such as data storage, user interface and so on are quite important for the whole the application and the usage of the application.

##### ***3.1.1. PROBLEM***

In this function, some problems have been met.

When create a business card, the information which are filled should be stored in the phone. But how the information can be stored in PIM? It could not just save it as a structure in J2ME. The format that calls the PIM should be used correctly.

When create a business card, how to make the interface best of this function? Not just use a textbox to input information. The canvas should be used. But how to solve the focus event is a big problem.

In J2ME, how to change form? If one function wants to call another function which is wrote in another Java document. Just call it directly or use some interface?

### **3.1.2. SOLUTION**

#### **3.1.2.1. THE TYPE OF STORE**

As the consideration of the function ‘Exchange Business Card’ by using Bluetooth, PIM is better than RMS not only for the data storage but also for the communication. When create a business card, the format that call the PIM should be used correctly. Here is how PIM store the data in the contact list.

```
if (list != null)
{
    //email
    item.addString(Contact.EMAIL, PIMItem.ATTR_NONE, m_sEmail);
    System.out.println("[1]");
    //tel
    int iTel = 0;
    try
    {
        iTel = Integer.parseInt(m_sTel);
    }
    catch(Exception e)
    {
        iTel = 0;
    }
    if (iTel != 0)
        item.addString(Contact.TEL, PIMItem.ATTR_NONE, String.valueOf(iTel));
    System.out.println("[2]");
    //name
    if (list.isSupportedField(Contact.FORMATTED_NAME))
    {
        item.addString(Contact.FORMATTED_NAME, PIMItem.ATTR_NONE, m_sName);

        System.out.println("[3]");
    }
}
```

#### **3.1.2.2. INTERFACE DESIGN**

As previously mentioned in Chapter 2, the interface 3 described by Figure 7, is the most appropriate while adding personal information. In this design, “Canvas” is used to improve this interface to let it more efficient and intuitive.

```

public static final void setFullScreen(Canvas canvas, boolean bMode)
{
    if (canvas != null)
    {
        canvas.setFullScreenMode(bMode);
        SCREEN_WIDTH = canvas.getWidth();
        SCREEN_HEIGHT = canvas.getHeight();

        BUTTON_WIDTH = SCREEN_WIDTH / 3 - 10;

        m_nCommandW = (short)BUTTON_WIDTH;
        MobileConfig.m_font = Font.getDefaultFont();
        FONT_HEIGHT = m_font.getHeight();
        ROW_HEIGHT = FONT_HEIGHT + 6;

        m_nCommandH = (short)ROW_HEIGHT;
    }
}

```

And many attributes of the canvas be set in this function, like: the height and width of the screen and font.

```

public static int SCREEN_WIDTH = 176;
public static int SCREEN_HEIGHT = 208;

public static int BUTTON_WIDTH = 0;
public static int FONT_HEIGHT = 24;
public static final byte TITLE_HEIGHT = 32;

```

Show all the information on the screen use Graphics:

```

g.setColor(0x000000);
Pub.drawTitle(g, "Create Business Card");

if (m_sName == null) {
    m_sName = "";
}
if (m_sTel == null) {
    m_sTel = "";
}
if (m_sTel == null) {
    m_sTel = "";
}
if (m_sEmail == null) {
    m_sEmail = "";
}
if (m_sAddr == null) {
    m_sAddr = "";
}

```

In order to solve the focus event, different background color and font color be set to identify the focus lose or not in input event.

```

public static int BGCOLOR = 0xDCf9FC;;
public static int FOCUSBGCOLOR = 0x2892c9;
public static int iFOCUSCTLColor = 0xB00000;
public static int FOCUSBORDERCOLOR = 0x57e2fc;
public static int FOCUSBORDERCOLOR2 = 0x1c4e96;
public static int FOCUSFontColor = 0xFFFFFFFF;
public static int FontColor = 0x0C3088;
public static int NORMALBORDERCOLOR = 0x5A9FA5;
public static int FontColor2 = 0x067cb7;

```

And different time, the text shows in different ways. When the text is null, just show the text name. When the information be adding, just show the things which be adding. When add is ok and the text is not null, it shows text name and information.

```

Pub.drawTextBoxAll(g, "", (currentItem == 0 && m_sName.length() > 0)
    ? m_sName : "Name:" + m_sName, itemsX, itemsY[0], 0, 0,
    (currentItem == 0) ? 1 : 0);
Pub.drawTextBoxAll(g, "", (currentItem == 1 && m_sTel.length() > 0)
    ? m_sTel : "Tel:" + m_sTel, itemsX, itemsY[1], 0, 0,
    (currentItem == 1) ? 1 : 0);
Pub.drawTextBoxAll(g, "", (currentItem == 2 && m_sEmail.length() > 0)
    ? m_sEmail : "Email:" + m_sEmail, itemsX, itemsY[2], 0, 0,
    (currentItem == 2) ? 1 : 0);
Pub.drawTextBoxAll(g, "", (currentItem == 3 && m_sAddr.length() > 0)
    ? m_sAddr : "Addr:" + m_sAddr, itemsX, itemsY[3], 0, 0,
    (currentItem == 3) ? 1 : 0);

```

### 3.1.2.3 Back

When press the back command, the screen from one form back to the main form. Could not just use `mainScreen()` directly. It should use an interface to back to another screen.

```

private final Meettia midlet;
private final Displayable caller;
this.midlet = midlet;
this.caller = caller;

if (command == backCommand) {
    Display.getDisplay(midlet).setCurrent(caller);
}

```

## 3.1.3. INTERFACE

After press “Create Business Card” in main screen (Figure 14), the interface of this function will showed in the screen. It includes name, telephone number, E-mail and address. Button “Save” and “Back” can be pressed. Choose “Save” can store the information which is filled. Choose “Back” can back to the main screen. When the form is empty, it just shows the form name. The information can be filled in it like Figure 15.

**Figure 15. The empty form of Create Business Card**

After filled all the information, it looks like Figure 16. When the information is adding, it show the things which be adding. When add is ok and the text is not null, it shows text name and information. Choose the button “Save” can store the information which is filled. Choose the button “Back” can back to the main screen.

**Figure 16. The finished form of Create Business Card**

When add is ok, press “Save”. The information which is added will be saved in the phone. And the screen shows “Save succeeds” like Figure 17. It means save is succeeded.



**Figure 17. The save successful tip**

### ***3.2.Exchange Business Card***

In this function, the users can exchange business card with others who also has the Meetia application. It includes send and receives. When the send command or receive command be chosen, it asked to open the Bluetooth. Server can search around and choose the client to link. Then the vCard can be sent.

#### ***3.2.1. PROBLEM***

In this function, the phone will search the client around, but how could 2 Bluetooth link. It is no reaction after press “send” or “receive”. Is there any special variable to connect client and server?

After link with another phone, how to send information with each other? Is there any command to send information by using Bluetooth? And how receive the information?

The command writeUTF() just can send information byte by byte . But how send a vCard? A vCard include many information, could it send as object. That will more convenient.



### **3.2.2. SOLUTION**

#### **3.2.2.1. LINK**

By using the Bluetooth, 2 phones be connected at first. Every Bluetooth has a UUID. The UUID in client and server should be the same. So the corresponding client can be found.

```
private DiscoveryAgent discoveryAgent=null;
private static final UUID Server_ID=new UUID("F0E0D0C0B0A000908070605040302010",false);
private UUID [] uuidSet;
private String rURL=null;
private ServiceRecord sr=null;
```

#### **3.2.2.2. SEND INFORMATION**

After link with another phone, the command writeUTF() should be used to send information. It can send information byte by byte.

```
try{
    dos.writeUTF(tf.getString());

    dos.flush();

    tf.setString("");
} catch (Exception ex) {
    ex.printStackTrace();
}
}
```

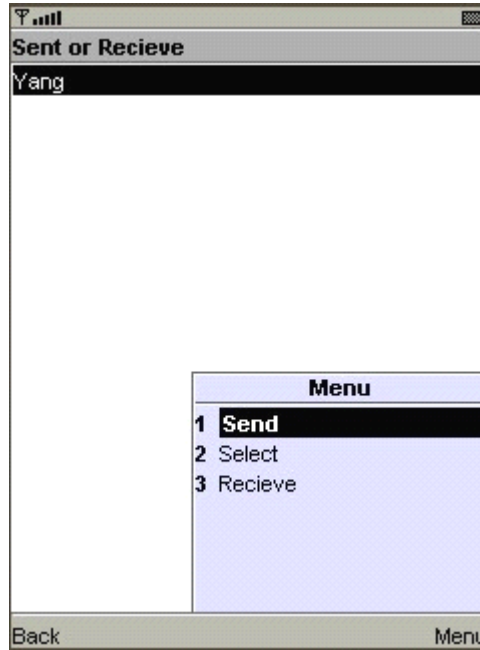
#### **3.2.2.3. SEND AN OBJECT**

A vCard include many information, the command writeUTF() just can send information byte by byte . So the vCard should be sent as an object. The OBEX can send it like a object.

```
File f =new
File((String)op.getReceivedHeaders().getHeader(HeaderSet.NAME))
;
FileOutputStream fos = new FileOutputStream (f);
```

### **3.2.3. INTERFACE**

After press “Exchange Business Card” in main screen (Figure 14), the application will enter the interface of “Exchange Business Card”, the list of business cards which are created in the first function is shown in the screen like Figure 18. And user can choose “Send” to send business card with other user who open the Meetia application and choose “Receive”. Choose “Select” can see the detail about the business card. Choose “Receive” can receive business card which the server sends.



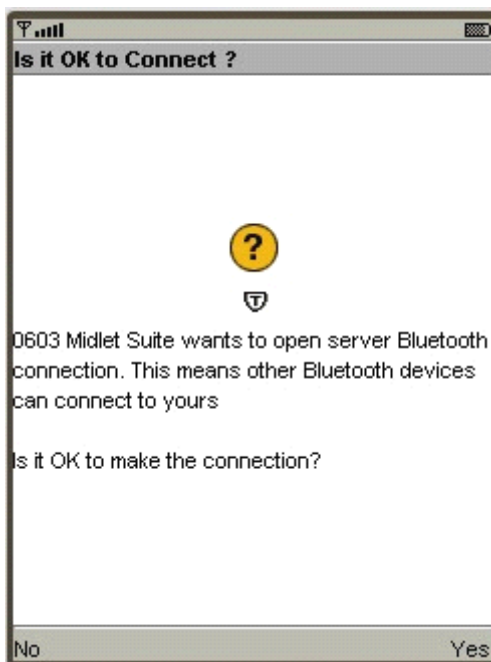
**Figure 18. The interface of Exchange Business Card**

After choose “Select”, it show the detail of this business card on the screen like Figure 19. It is the information that user filled in the first function. Choose “Back” will back to the interface of Exchange Business Card (Figure 18).



**Figure 19. The detail of business card**

If “Send” be chosen, the application asks to open the Bluetooth like Figure 20. Choose “Yes” can open the Bluetooth and get connection with client. Choose “No” will back to the interface of “Exchange Business Card”(Figure 18).



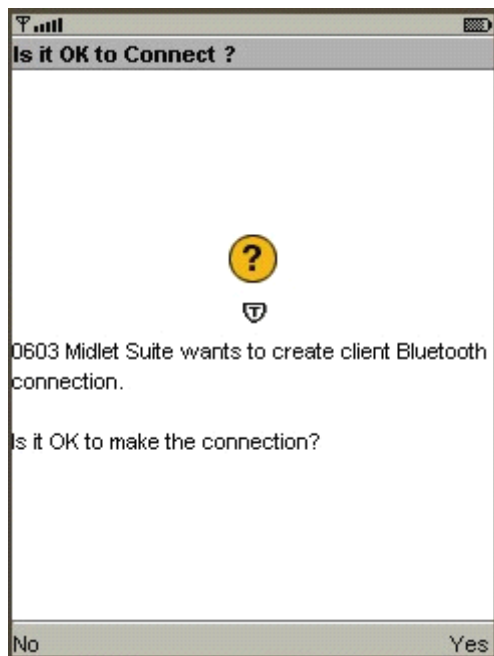
**Figure 20. The open tip of server Bluetooth connection**

After press “Yes” to open the Bluetooth, the business card will be sent (Figure 21). It shows the information of the sending business card on the screen. Choose button “Back” will break the sending and turn off the Bluetooth, back to the interface of “Exchange Business Card”(Figure 18).



**Figure 21. The interface of ready for send**

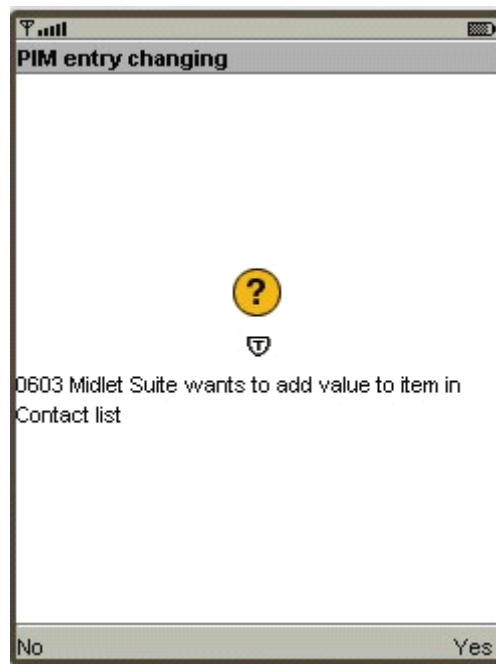
Choose “Receive” from the interface of Exchange Business Card (Figure 18), the application ask to connection (Figure 22). Choose “Yes” can open the Bluetooth and get connection with server, then receive the business card. Choose “No” will back to the interface of Exchange Business Card.



**Figure 22. The connection tip of client**

After press “Yes” to connection, the Bluetooth will turn on and the business card will be received. After the business card received, the application will ask whether to store the information into contact list (Figure 23). Choose “Yes” can store the information of the business card into contact list. Choose “No” will back to the interface of Exchange Business

Card (Figure 18).



**Figure 23. The add tip**

### ***3.3.Book a meeting***

In this function, the users can choose date, time, and member. Then you can send this information as a vCalendar with Bluetooth. Then the receiver can store it in the calendar.

#### ***3.3.1. PROBLEM***

When book a meeting, the meeting date and meeting time want to be created, how to call the calendar and clock in mobile phones?

After the users choosing the date and time, it should be stored as a vCalendar. vCalendar is more convenient in data transmission. But how to save it as a vCalendar is a problem.

#### ***3.3.2. SOLUTION***

##### ***3.3.2.1. CREATE DATE AND TIME***

In order to create the meeting date and meeting time, the calendar and the clock should be called to make users easily to choose. The variable with DATE\_TIME type could do this.

```
nameDateField = new DateField("Date", DateField.DATE_TIME);
append(nameDateField);
```

### 3.3.2.2. *SAVE IT AS A VCALENDAR*

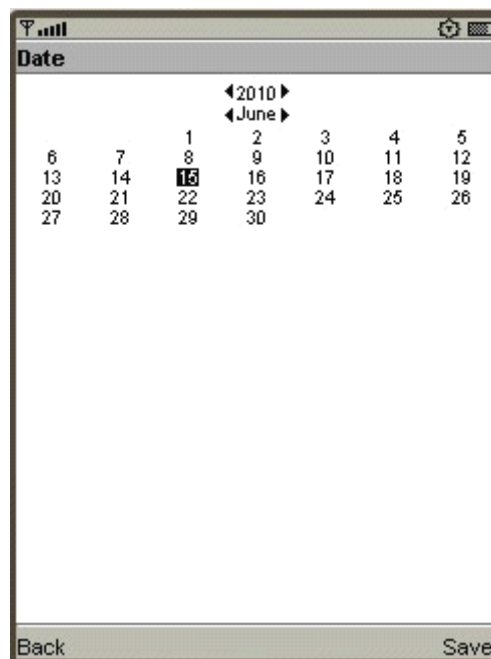
After the users choosing the date and time, it should be stored as a vCalendar. The Event in PIM could do this.

```
item = ((EventList) list).createEvent();
Date aDate = new Date();

if (list.isSupportedField(Event.START))
    item.addDate(Event.START, PIMItem.ATTR_NONE, aDate.getTime());
```

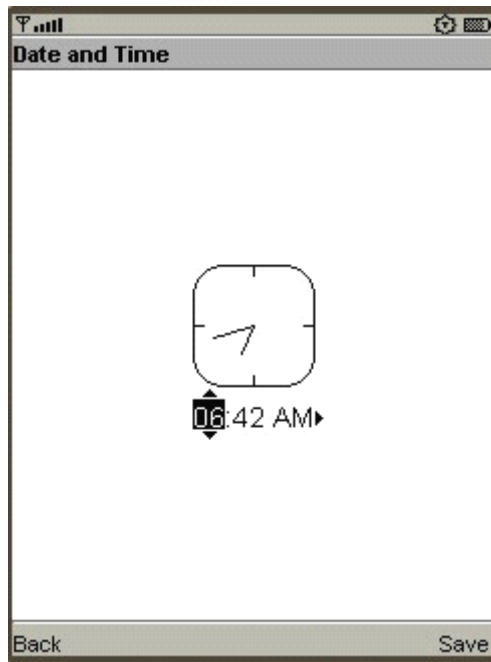
### 3.3.3. *INTERFACE*

After press “Book a meeting” in main screen (Figure 14), the application can let user choose date and time, then save them as a vCalendar and send it to book a meeting. After press “Choose date”. A calendar be showed on the screen to let user choose the date (Figure 24). Choose “Save” can save the date. Choose “Back” will back to rechoose the date and time.



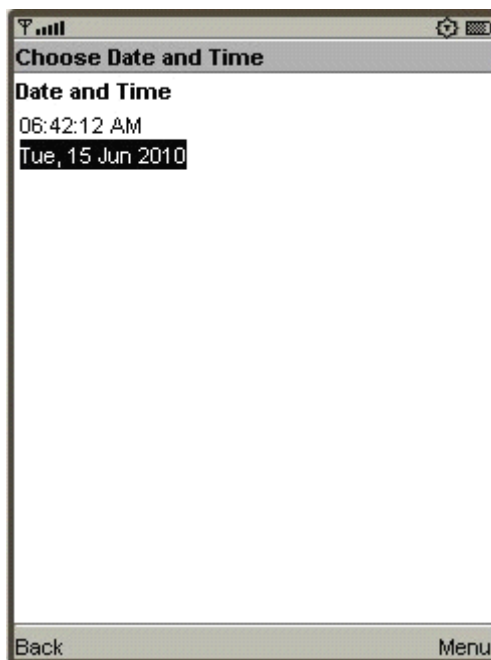
**Figure 24. The interface of Choose date**

After press “Choose time”. A clock will be showed on the screen to let user choose the time (Figure 25). Choose “Save” can save the time. Choose “Back” will back to rechoose the date and time.



**Figure 25. the interface of Choose time**

After the chosen date and the chosen time are saved, the date and time which are chosen would show in the screen like Figure 26. Choose "Menu" can select "Send" to send the date and time with Bluetooth. Choose "Back" will back to the interface of main screen (Figure 14).



**Figure 26. The interface after add succeed**

## 4. RESULTS

The aim of this thesis was to develop an application to be executed on smart phones. The Java ME platform has been chosen due to its high availability on the smart phone market. The application is to solve the three main functions of Meetia.

The main functions of Meetia application are Create Business Card, Exchange Business Card, and Book a meeting.

The function of “Create Business Card” which has been finished first, a business card can be created including the basic personal information such as name, telephone number, E-mail, address and so on. PIM could be used in this function to store the information in user’s mobile phones. How to connect to the contact list is the main problem in this function. And the user interface has been changed for many times to make it better for the users to use.

Then second function which has done is “Exchange Business Card”. In this function, the list of business card which has created in the first function will be shown in the display. Each business card can be chosen to send by Bluetooth and selected to see the information of the business card. The most important API which has been used in this function is the Bluetooth API because it is related to the problem how to send vCard. OBEX has been chosen to solve this problem because OBEX can send information as an object.

The last function is “Book a Meeting”. In this function, the date and the time which user has chosen can be created as a vCalendar and it also can be sent through the function of “Book a meeting” by using Bluetooth.

How to develop a mobile phone application is a big problem for a beginner at first. So many problems have been met during the programming. As far as we have concerned, although a mobile phone application is based on JAVA, it is much different from an ordinary JAVA application which we usually developed. Instead of main function, startApp(), pauseApp(), destroyApp() has been learnt to be used. Then how to let each function work? The type of storage, the type of transmission has been compared to find the best solution. And the PIM and Bluetooth have been chosen in this application.

So J2ME is an appropriate tool for programming mobile phone application. And Meetia is designed for smart phone for person to connect and exchange with each other. The functions of “Exchange Business Card” and “Book a Meeting” provided the user the convenient way for business aspects, and also save paper card.



## 5. CONCLUSION

In order to develop this application more efficiently, the most important aspect is to understand the protocols of different APIs of J2ME and then use them to develop the application. With the consideration of different aspects in Chapter 2, PIM and Bluetooth APIs are chosen in this project. The design of user interface is also quite important to develop because it shows directly for the users while the users use Meetia application.

There are three main functions of Meetia application: Create Business Card, Exchange Business Card, and Book a meeting. The function of “Create Business Card” is the most basic function among them because the functions of “Exchange Business Card” and “Book a meeting” are related to it.

To implement Meetia application, J2ME has been used to develop for the smart phone application. To implement every function, there are three basic steps: develop the interface, achieve the function, and improve the interface. The function of “Creating Business Card” has been implemented first, which links it to the contact list. After finished one function, it will be downloaded to the mobile phone to test and to check whether it should be improved.

Much information about the development of smart phone had been learnt; many different methods had been used and then compared which is the best one should be chosen.

In summary, according to the requirements, we have developed these basic functions. By this learning process, we mastered the knowledge of how to develop a mobile application and from the user’s points the design of user interface is also very important which we neglected at first. The experience for this project we obtained is very valuable.

## REFERENCES

- [1] A project from the java.net JUGs community, TastePhone Server version Alpha 2  
[http://www.club-java.com/TastePhone/J2ME/MIDP\\_mobile.jsp](http://www.club-java.com/TastePhone/J2ME/MIDP_mobile.jsp)
- [2] Sun Developer Network, Java Me  
<http://java.sun.com/javame/index.jsp>
- [3] Mike Kirkup, Best Practices for mobile development  
<http://www.blackberry.com/DevMediaLibrary/view.do?name=best>
- [4] Sun Developer Network, Sun Java Wireless Toolkit for CLDC,  
<http://java.sun.com/products/sjwtoolkit/download.html?feed=JSC>
- [5] Qusay H.Mahmoud, Wireless Application Programming with J2ME and Bluetooth, February 2003  
<http://developers.sun.com/mobility/midp/articles/bluetooth1/>
- [6] Bluetooth traveler, Retrieved 9 April 2010  
<http://www.hoovers.com>
- [7] General packet radio service from Qkport  
[http://about.qkport.com/g/general\\_packet\\_radio\\_service](http://about.qkport.com/g/general_packet_radio_service)
- [8] Clint Smith, Daniel Collins. "3G Wireless Networks", page 136. 2000.
- [9] Sun Developer Network, Wireless Application Programming with J2ME and Bluetooth  
<http://developers.sun.com/mobility/midp/articles/bluetooth1/>
- [10] Qusay H.Mahmoud, Part II: Wireless Application Programming with J2ME and Bluetooth, April 2003  
<http://developers.sun.com/mobility/midp/articles/bluetooth2/>
- [11] Service Discovery Application Profile  
[http://www.bluetooth.com/SiteCollectionDocuments/SDAP\\_SPEC\\_V12.pdf](http://www.bluetooth.com/SiteCollectionDocuments/SDAP_SPEC_V12.pdf)
- [12] Generic Object Exchange Profile (GOEP)  
<http://www.bluetooth.com/English/Technology/Works/pages/goep.aspx>
- [13] J2ME record management store, Soma Ghosh, 01 May 2002  
<http://www.ibm.com/developerworks/library/wi-rms/>
- [14] Generic Object Exchange Profile of BLUETOOTH SPECIFICATION Version 1.1  
[http://www.bluetooth.com/SiteCollectionDocuments/GOEP\\_SPEC\\_V12.pdf](http://www.bluetooth.com/SiteCollectionDocuments/GOEP_SPEC_V12.pdf)
- [15] K11-Object Push Profile, palo wireless  
[http://www.palowireless.com/infotooth/tutorial/k11\\_opp.asp](http://www.palowireless.com/infotooth/tutorial/k11_opp.asp)
- [16] File Transfer Profile of BLUETOOTH SPECIFICATION Version 1.1  
[http://www.bluetooth.com/SiteCollectionDocuments/FTP\\_SPEC\\_V12.pdf](http://www.bluetooth.com/SiteCollectionDocuments/FTP_SPEC_V12.pdf)
- [17] Synchronization Profile  
<http://www.bluetooth.com/English/Technology/Works/pages/sync.aspx>
- [18] vCard and vCalendar  
<http://www.imc.org/pdi/>