TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A
MINOR PROJECT REPORT
ON
"BARCODE READER"

SUBMITTED BY:

Arvind Sah (061bct504)	sah_arvind@yahoo.com	9841468680
Jaya Ram Sedai (061bct519)	js.jayas@gmail.com	9841804661
Rajan Kumar Kharel (061bct534)	najar_85@hotmail.com	9841816588

SUBMITTED TO:
Department Of Electronics and Computer Engineering
Pulchowk Campus
Date: Mon, February, 25, 2008

AKNOWLEDGEMENT

For completing successfully this project, many people have given us valuable suggestions and helped us providing useful ideas. Most of we are grateful to Department of Electronics and Computer which has provided us golden opportunity to design a minor project in suitable high level programming language. Even equally we would like to thank our respected teachers **Jayaram Timilsina, Bikash Shrestha , Dipen Chapagain** who have been so kind in giving us suggestions and resources as required for the project. We are thankful to our project examiner **Niraj Sakhakaarmi** . We can not forget all the people who have helped in the NOKIA and sun java forum, especially to **Robert Adelman**.

Any helpful suggestions regarding our project from teachers and friends would be kindly appreciable.

THANKS

Arvind Sah(061-BCT-504)

Jaya Ram Sedai(061-BCT-519)

Rajan Kumar Kharel(061-BCT-534)

ABSTRACT

In today's modern world, almost all of the product are tagged with the encoded visual information in the form of bar. Among them the most used in the commercial products like books, food materials and other are tagged with the EAN13 symbology. Especially in the books, the EAN13 number is transferred into ISBN number, the EAN13 with 978 is for book products. So the hidden information in the book in encoded form is to be decoded in to original number to know the products information. The decoded number has been assigned for that particular book uniquely, so after finding the unique id of that book, we can find the information (updated) from the database or wherever else. With the use of, say, a laser barcode scanner, information about a product such as description and price can be quickly obtained. Another common item in many parts of the world is mobile phone. No doubt the use of mobile phone is becoming increasingly widespread, and its features also are rapidly growing. We can use mobile phone to access the Internet, and take pictures and videos. We certainly can also use it as a barcode reader. Consumers can capture an image of a barcode label using their camera phone (i.e. mobile phone with built-in camera). The phone, programmed to interpret barcode images, will generate the barcode value which is used to identify the product. By connecting online, consumers can then get access to a wealth of information about the product. This includes not only product description and price but also product review, price comparison, location of retailers, etc.

Our project will resolve the virtual information to physical domain using mobile client, Bluetooth, and information server; the computer using J2ME, J2SE and Bluetooth Dongle for communication between them. The increasing availability of camera phones, i.e., mobile phones with an integrated digital camera, has made us to use its functionality with the images it takes directly to the program for processing and also to use Bluetooth, the wireless technology and J2ME for wireless devices. The main contribution of this project is 1D bar code EAN 13 recognition software that is intended to facilitate the creation of novel applications. In this project, we have performed scan line based technique to recognize and resolve EAN-13 barcodes from images captured by digital cameras. The ultimate aim of our approach is to enable electronic devices with cameras such as mobile phones and Personal Digital Assistants (PDAs) to act as a barcode reader.

TABLE OF CONTENTS

ABBREVIATIONS

1. CLDC: Connected Limited Device Configuration
2. DB: Database
3. EAN: European Article Number
4. ISBN: International Standard Book Number
5. J2ME: Java 2 Micro Edition
6. J2SE: Java 2 Standard Edition
7. JSR: Java Specification Request
8. MIDP: Mobile Information Device Profile
9. PC: Personal Computer
10. PDA: Personal Digital Assistant
11. RFCOM: Radio Frequency Communication
12. UUID: Universally Unique Identifier

INTRODUCTION

Motivation:

Automatic identification technology such as RFID promises to connect physical objects with virtual representations or even computational capabilities. However, even though RFID tags are continuously falling in price, their widespread use on consumer items is still several years away. Much more ubiquitous are printed bar codes, yet so far their recognition required either specialized scanner equipment, custom-tailored bar codes or costly commercial licenses – all equally significant deployment hurdles. We have developed an EAN-13 bar code recognition and information system that is both lightweight and fast enough for the use on camera-equipped mobile phones, thus significantly lowering the barrier for large-scale, real-world testing of novel information and interaction applications based on “connected” physical objects. We hope that this “low tech” version of bridging the gap will allow the community to quickly develop and try out more realistic and widespread applications, and thus gain real-world experiences for better jump-starting the future internet of things, today.

Aim:

In this project we are trying to unify the latest technologies in the single implementation. This is the age of mobile technology, latest mobile devices with high facilities and functionalities are evolving day per day. For example the quality of camera of these devices is increasing to auto focus and higher resolutions. So we tried to adapt the functionality of camera of latest mobile phones for the further processing of the image.

Similarly the trend of the world now is moving towards the wireless Bluetooth technology with high speed data transfer. Bluetooth is used for data communication between mobile devices such as phones, PDAs etc and also to and from PC to them. Our aim is to use this technology and apply the Bluetooth communication between our server and client. These two technologies are used to resolve the next technology i.e., barcode technology, which is the most for any products in these modern world. So we are sure that our aim to connect the physical things to the virtual information and vice versa is fulfilled with the camera based image processing technique. To implement this technology we used the Java technology for coding the client and server. The J2ME client and the J2SE server are designed to decode the barcode and give the information of that code.

So, the aim of our project is to combine the four technologies; mobile technology, Bluetooth technology, barcode technology and java technology for image processing and pattern recognition and resolving in the camera phones.

Today's Role of Barcode Recognition and Related Works (A Survey in Barcode Recognition)

The idea of linking real-world products with virtual information has been around for quite some time. In 1998, Barrett and Maglio already described a system for attaching information to real-world objects, while 1999 Want et al. expanded upon the idea and linked arbitrary items through the use of RFID tags with both information services and actions. Since then, a number of research projects have continued to explore this concept of "bridging the gap", i.e., the automatic identification of individually tagged real-world products in order to quickly look up information or initiate a specific action. With the increasing mobility of powerful computing systems, e.g., mobile phones or handheld PDAs, this bridging can even be done right when we need it, where we need it. While RFID potentially offers an unprecedented user experience due to its detailed means for identification (i.e., on a per item basis) and the lack of a line-of-sight requirement for reading, most industry analysts agree that an item-level rollout (e.g., having an RFID tag on every single supermarket product) is still several years away. In contrast, the printed bar codes are practically ubiquitous: Virtually every item sold today carries an internationally standardized bar code on its packaging, enabling not only checkout registers to quickly sum up one's shopping items, but also to identify a product and look up a wealth of related information. Obviously, using bar codes for linking real-world objects to virtual information has a number of drawbacks when compared to an RFID-enabled future with corresponding mobile RFID readers, such as NFC-enabled1 mobile phones. Due to their sensitivity to soiling, ripping, and lighting conditions, optical bar code recognition can be difficult. Until recently, reading a conventional (i.e., 1D) bar code inevitably required a separate laser scanner or a corresponding mobile phone scanner attachment. The increasing availability of camera phones, i.e., mobile phones with an integrated digital camera, has begun to simplify this process, however. After 2D bar codes have been successfully recognized by most consumer-grade camera phones for quite some time, the continuously increasing quality of both the camera resolution and the employed lenses have finally made it feasible to directly read 1D bar codes with such cameras, without the need for special attachments or handheld lasers. This significantly changes the attractiveness of using barcodes for the above physical-to-digital linkage: Instead of waiting several years for a comprehensive item-level roll out of RFID tags, or forcing people to carry around specific scanner attachments for their mobile phones, the support of 1D bar code recognition on any camera phone immediately allows anybody to interact with almost any commercially available product – all it takes is a small application download. The main contribution of our project is 1D bar code recognition software that is intended to facilitate the creation of novel applications and services. We believe that the adequate performance of our recognition software.

Prior work on using printed bar codes for linking real-world objects with virtual information has often used two dimensional bar codes, which do not use bars of varying widths but instead blocky rectangles that lend themselves much better to low resolutions or misalignments. There is a wide variety of code symbologies available, such as Semacodes, Spot codes, the Japanese QR-System2, or Rohs' Visual Codes. All of these systems were specifically designed to simplify camera-based recognition. However, while they offer both improved detection rates as well as additional services such as range and alignment detection, none of these codes enjoys widespread use, let alone comes close to the billions of products carrying EAN-13 bar codes today. Also, none of these codes is linked to a wealth of EAN-13-indexed information available in online databases today. A number of algorithms have already been implemented for the visual decoding of 1D bar codes on desktop computers3. Most of these are based on the transformation of the original image information into a decoding domain that simplifies bar code identification, like approaches based on the Fourier transformation or the Hough transformation as proposed by Muniz et al. These approaches are often used in professional image recognition software, as the offer very good recognition rates. However, their requirements in terms of system resources can be too demanding for typical mobile devices. While both Ohbuchi et al. and Chai and Hock have presented algorithms intended for mobile devices, these algorithms so far have not been implemented or tested on actual mobile camera phones. As an alternative to costly domain transformation, a much simpler approach is based on so-called scanlines, which try to detect the bar code along a particular line through the image. As such algorithms need much fewer computing resources; they are specifically relevant for the use on mobile camera phones. Their drawbacks, however, lie in their often poor recognition rates when dealing with dirty surfaces, reflections or shadows, or slight misalignments and their need of detecting the bar code in the image first (in order to properly align the scanline). We improved on this by making extensive use of multiple scanlines. Note that given the commercial potential of the 1D barcode recognition on mobile phones, it is not surprising that a number of commercial solutions exist. Scanbuy offers an application called ScanBuy Decoder, which is capable of recognizing 1D barcodes. Similar applications can be bought from PaperClick, Gavitec, and

MediaStick, to name but a few. While informal trials with some freely available beta programs from the above vendors showed a comparable, sometimes even superior performance of our system, we explicitly abstained from conducting formal comparisons, as improving the recognition rate or speed is not our primary goal. Instead, we are trying to create an easily usable and robust barcode recognition system for mobile phones, together with an resolving framework. The currently available commercial systems, in contrast limit barcode resolving to vendor applications and/or a fixed set of lookup services.

OBJECTIVES AND GOALS

This is our research project in the field of barcode recognition using the camera phones and the main approach of our project is to deal with the image captured by the camera of the mobile phone. Besides the image processing we are also intended to the interaction of image and the information it contains which may be updated time to time. So our project is designed for digital camera based image processing and use of the simple scan line based and less complicated algorithm. The main objectives of our project are:

1) To make the simple and complete software to read EAN-13 barcode.
2) To implement the camera based image processing and simple algorithm.
3) To take new step in mobile technology.
4) To link the physical world and virtual information.
5) To implement J2ME for mobile device and J2SE for the PC server.
6) To replace complex hardware technique from most used camera phones and java technology to read barcode.
7) To implement wireless communication between J2ME client (mobile phone) and information server (computer) using Bluetooth technology within the program .

TECHNOLOGY

Barcode:

In general, a barcode is a graphical representation of the information of the products. There are a lot of representation techniques for product information in barcode like UPC-A, EAN-13 etc. but EAN-13 is used as internationally. As explained above we are programming for EAN-13 as international requirement.

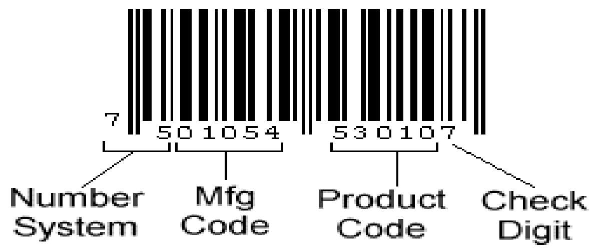


Fig: EAN-13 barcode

A bar code is one of the most widely used automatic identification technologies. Bar coding provides automatic entry of information into computer systems. This, in turn, greatly enhances control over products, materials, paperwork and personnel through applications in inventory, work in progress, tracking, and cost accounting. Bar codes themselves are a machine readable pattern of alternating parallel bars and spaces representing numbers and other characters .They can represent a product ID number, an order number, or any other information that must be entered into a computer system. The various bar code symbologies, like languages, encode information differently.

EAN 13 Barcode:

A European Articles Number (EAN) is a bar-coding standard which is a superset of the original 12-digit Universal Product Code (UPC). The *EAN-13* is defined by standards organization GS1. EAN is also called Japanese Article Number (JAN) in Japan. The *EAN-13* barcodes are used worldwide for marking retail goods.EAN-13, based upon the UPC-A standard, was implemented by the International Article Numbering Association (EAN) in Europe. This standard was implemented mostly because the UPC-A standard was not well designed for international use.

EAN-13 is a superset of UPC-A. This means that any software or hardware capable of reading an EAN-13 symbol will automatically be able to read an UPC-A symbol. The only difference between EAN-13 and UPC-A is that the number system code in UPC-A is a single digit from 0 through 9 whereas an EAN-13 number system code consists of two digits ranging from 00 through 99, which is essentially a country code. Each country has a numbering authority which assigns manufacturer codes to companies within its jurisdiction. The manufacturer code is still five digits long, as is the product code, and the check digit is calculated in exactly the same way. In *EAN-13* the symbol encodes 13 numerals divided into four parts:

1) *System Code*: The first two or three digits, usually identifying the country in which the manufacturer is registered (not necessarily where the product is actually made). When the EAN-13 barcode is a conversion of an 10-digit ISBN or ISSN code, the system code will be 978 or 977 for ISBNs (Borland), or 977 for ISSNs.The first three digits of the barcode of any product represents the country. Note that EAN codes beginning with 0 are rarely used, as this is just an addition to a 12-digit UPC. Since most scanners and registers worldwide can read both equally, most manufacturers in North America still only use UPC.

2) *Manufacturer code:* Consisting of four, five or six digits depending on the length of the system or country code.

3) *Product code*: Consisting of five digits.

4) *Check digit*: A single checksum digit. The check digit is computed modulo 10, where the weights in the checksum calculation alternate 1 and 3. In particular, since the weights are relatively prime to 10 the EAN system will detect all single digit errors. But since the difference of consecutive weights is even, the EAN system does not detect all adjacent transposition errors.

There are various ways to encoding and decoding the barcode. Encoding means breaking the EAN Barcode image into the corresponding string which implies their information related to the string. Similarly, Decoding is the reverse process of encoding.

Encoding:

To encode an EAN-13 barcode, the digits are first split into 3 groups, the first digit, the first group of 6 and the last group of 6. The first group of six is encoded using a scheme whereby each digit has two possible encodings, one of which has even parity and one of which has odd parity. The first digit is encoded by selecting a pattern of choices between these two encodings for the next six digits, according to the table below. (Unlike the other digits, the first digit is not represented directly by a pattern of bars.) All digits in the last group of six digits are encoded using a single set of patterns which are the same patterns used for UPC.

If the first digit is zero, all digits in the first group of six are encoded using the patterns used for UPC, hence a UPC barcode is also an EAN-13 barcode with the first digit set to zero. In encoding process the white gap (or space is represented by 0) and black gap represented by 1.

First digit	First group of 6 digits	Last group of 6 digits
0	LLLLLL	RRRRRR
1	LLGLGG	RRRRRR
2	LLGGLG	RRRRRR
3	LLGGGL	RRRRRR
4	LGLLGG	RRRRRR
5	LGGLLG	RRRRRR
6	LGGGLL	RRRRRR
7	LGLGLG	RRRRRR
8	LGLGGL	RRRRRR
9	LGGLGL	RRRRRR

Fig: Encoding the Barcode Image

Digit	L-code	G-code	R-code
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000

9	0001011	0010111	1110100

Table: Encoding Of Barcode

Bluetooth:

Bluetooth is a low cost, low power, short range radio technology intended to replace cable connection between cell phones, PDAs and other portable devices .It can clean up desk considerably, making wires between workstation, mouse, laptop computer etc. obsolete.

Bluetooth networking transmits data via low-power radio waves. It communicates on a frequency of 2.45 gigahertz (actually between 2.402 GHz and 2.480 GHz, to be exact). This frequency band has been set aside by international agreement for the use of industrial, scientific and medical devices (ISM).

Fig: Bluetooth Dongle

A number of devices that may be already used take advantage of this same radio-frequency band. Baby monitors, garage-door openers and the newest generation of cordless phones all make use of frequencies in the ISM band. Making sure that Bluetooth and these other devices don't interfere with one another has been a crucial part of the design process.

One of the ways Bluetooth devices avoid interfering with other systems is by sending out very weak signals of about 1 mill watt. By comparison, the most powerful cell phones can transmit a signal of 3 watts. The low power limits the range of a Bluetooth device to about 10 meters (32 feet), cutting the chances of interference between our computer system and our portable telephone or television. Even with the low power, Bluetooth doesn't require line of sight between communicating devices. The walls in our house won't stop a Bluetooth signal, making the standard useful for controlling several devices in different rooms.

Bluetooth can connect up to eight devices simultaneously. With all of those devices in the same 10-meter (32-foot) radius, we might think they'd interfere with one another, but it's unlikely. Bluetooth uses a technique called spread-spectrum frequency hopping that makes it rare for more than one device to be transmitting on the same frequency at the same time. In this technique, a device will use 79 individual, randomly chosen frequencies within a designated range, changing from one to another on a regular basis. In the case of Bluetooth, the transmitters change frequencies 1,600 times every second, meaning that more devices can make full use

Java Database Connectivity:

JDBC (Java database connectivity) is an API for the Java programming language that defines how a client may access a database provides methods for querying and updating data in a database. JDBC is oriented towards relational databases.

Overview:

JDBC has been part of the Java Standard Edition since the release of JDK 1.1. The JDBC classes are contained in the Java package java.sql. Starting with version 3.0, JDBC has been developed under the Java Community Process. JSR 54 specifies JDBC 3.0 (included in J2SE 1.4), JSR 114 specifies the JDBC Rowset additions, and JSR 221 is the specification of JDBC 4.0 (included in Java SE 6).

HOW OUR SYSTEM WORKS

Our EAN-13 bar code recognition and resolution system contains two parts:

i) The barcode recognition component running entirely on J2ME enabled mobile phones that support the MMAPI9 (Mobile Media API extension) and

ii) The Java based information server component, which is located on a separate server, to which the detected product code is transmitted via a Bluetooth connection.

The provided client provides functionality to recognize an EAN13 code, communicate with the server and display the results. The information server provides the information from the database source for the incoming EAN13 number .The Incoming EAN 13 number is first converted into the ISBN and then database is searched for the information for that particular number and the information is transmitted client to display. Although this process could also be located on the phone itself, performing them on an external server provides us with greater extendibility, higher flexibility and better performance.

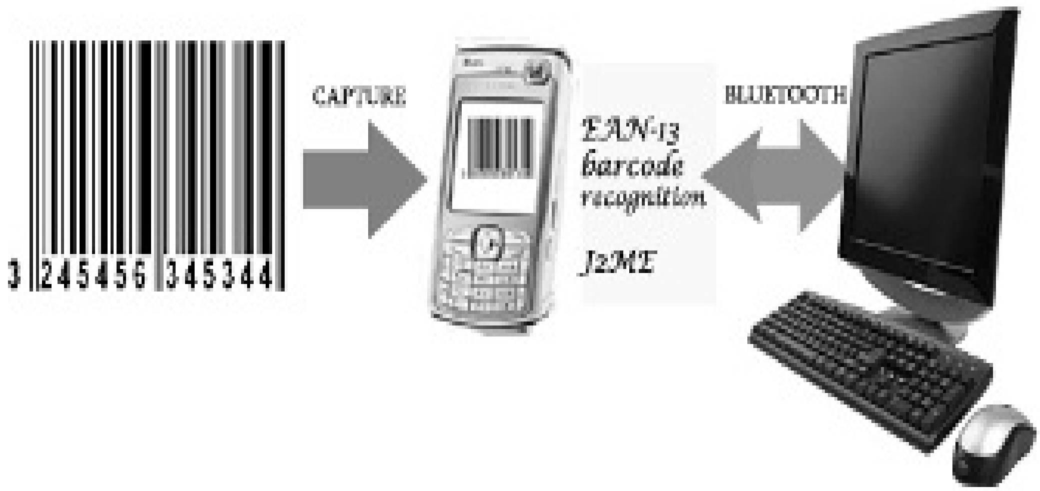


Fig: System block diagram Of Barcode Reader

REOGNITION ALGORITHM

In general, our recognition algorithm is scanline based. In order to improve robustness, we decided to not only use a single scanline, but a set of multiple, potentially arbitrarily oriented scanlines . If multiple scanlines cross the bar code, each with a different sensitivity, we can increase the chances that at least one of them will result in a properly recognized code. Also, multiple scanlines can be combined in a majority-voting fashion, were inaccuracies due to dirt or reflections on one line can be compensated by two or more correct identifications on other lines. By applying slightly different recognition parameters along each individual scanline (i.e., the binarization threshold that categorizes pixels into either black or white), the overall recognition accuracy can also be improved. Last not least, by using a variable amount of scanlines, we have a simple mechanism to adapt our algorithm to the processing power of the individual phone it is running on: The more computational capabilities available, the more scanlines and orientations we can try. Since the algorithm is scanline based, it cannot cope as well with image distortions as transformation-based algorithms. However, as the analysis below will show, our implementations sufficiently robust even for lower image resolutions. Also, it is quite fast, has very little memory requirements, and can be implemented relatively easy.

PROGRAMMING LANGUAGE

J2ME:

In J2ME, the Java runtime environment is adapted for constrained devices - devices that have limitations on what they can do when compared to standard desktop or server computers. For low-end devices, the constraints are fairly obvious: extremely limited memory, small screen sizes, alternative input methods, and slow processors. High-end devices have few, if any, of these constraints, but they can still benefit from the optimized environments and new programming interfaces that J2ME defines.

The Connected Limited Device Configuration (CLDC)

At the core of Java 2 Micro Edition (J2ME) are the *configurations*, the specifications that define the minimal feature set of a complete Java runtime environment. J2ME currently defines two configurations. In this article we look at the first of these, the Connected Limited Device Configuration, or CLDC for short.

The Java APIs for Bluetooth Wireless Technology

The Java 2 Platform, Micro Edition (J2ME) and Bluetooth technology are two of the most exciting offerings in the wireless industry today. J2ME, most compact of the three Java platforms, is inherently portable because it shares the Java "write once run anywhere" philosophy and thus enhances developer productivity. Bluetooth is a short-range universal wireless connectivity standard for electronic appliances and mobile devices.

Imagine being able to use our Bluetooth-enabled mobile phone to lock and unlock our car, operate our garage door, and control our TV, VCR, DVD player, and other consumer appliances. If we want to make that kind of control available to our users, we'll need to be able to write Bluetooth applications that customize these appliances, and deploy them in a way that lets users download them, to a cell phone for example. Bluetooth and J2ME can work together to achieve this unified vision. Bluetooth allows devices to communicate wirelessly and J2ME allows we to write custom applications and deploy them on mobile devices.

Mobile Media API (MMAPI):

The Mobile Media API, JSR 135 in the Java Community Process (JCP), extends the functionality of the J2ME platform by providing audio, video, and other time-based multimedia support to resource-constrained devices. As a simple and lightweight optional package, it gives Java developers access to native multimedia services available on a given device.

The MMAPI is an optional package within the J2ME platform. While the main emphasis is on devices that implement profiles based on the Connected Limited Device Configuration (CLDC), the API design also aims at supporting devices that implement the Connected Device Configuration (CDC) and the profiles based on CDC.

J2SE:

Java Platform, Standard Edition or Java SE is a widely used platform for programming in the Java language. It is the Java Platform used to deploy portable applications for general use.

In practical terms, Java SE consists of a virtual machine, which must be used to run Java programs, together with a set of libraries (or *packages*) needed to allow the use of file systems, networks, graphical interfaces, and so on, from within those programs.

It must be noted that the expressions such as *super*, *this* or the return type *void* and the method *main()* are not part of the class hierarchy. Instead they are implemented in the JVM architecture.

USED API and PACKAGES

javax.microedition.lcdui.Graphics*;

Provides simple 2D geometric rendering capability.

javax.microedition.lcdui.Canvas*;

The Canvas class is a base class for writing applications that need to handle low-level events and to issue graphics calls for drawing to the display.

javax.microedition.lcdui.Commandlistener*;

This interface is used by applications which need to receive high-level events from the implementation.

javax.microedition.lcdui.Choice;

Choice defines an API for a user interface components implementing selection from predefined number of choices

javax.microedition.lcdui.Displayable*;

An object that has the capability of being placed on the display.

javax.microedition.lcdui.Screen*;

The common superclass of all high-level user interface classes.

javax.microedition.media.Control*;

A Control object is used to control some media processing functions. The set of operations are usually functionally related. Thus a Control object provides a logical grouping of media processing functions.

Controls are obtained from Controllable. The Player interface extends Controllable. Therefore a Player implementation can use the Control interface to extend its media processing functions. For example, a Player can expose a Volume Control to allow the volume level to be set.

Multiple Controls can be implemented by the same object. For example, an object can implement both Volume Control and Tone Control. In this case, the object can be used for controlling both the volume and tone generation.

The javax.microedition.media.control package specifies a set of pre-defined Controls.

javax.microedition.media.Manager

Manager is the access point for obtaining system dependent resources such as Players for multimedia processing. A player object used to control and render media that is specific to the content type of the data. Manager provides access to an implementation specific mechanism for constructing Players. For convenience, Manager also provides a simplified method to generate simple tones.

javax.microedition.lcdui.Display

Display represents the manager of the display and input devices of the system. It includes methods for retrieving properties of the device and for requesting that objects be displayed on the device. Other methods that deal with device attributes are primarily used with objects and are thus defined there instead of here.

There is exactly one instance of Display per Midlet and the application can get a reference to that instance by calling the getyDisplay() method. The application may call the getDisplay() method at any time during course of its execution. The Display object returned by all calls to getDisplay() will remain the same during this time.

Javax.microedition.midlet

The MIDlet package defines Mobile Information Device Profile applications and the interactions between the application and the environment in which the application runs.

Javax.microedition.media

The MIDP 2.0 Media API is a directly compatible building block of the Mobile Media API (JSR-135) specification.

CASE STUDY

When we decided to build this project, we searched in the net to collect the in formation regarding the application and building methodology of this project. During the study phase of our project, we first study how to get snapshot from the mobile using j2me we thoroughly make study over different e-book and tutorial. In other side the study of the barcode pattern is also necessary so we involved on study of different study of the different types of barcode system from that study we came to conclude that EAN-13 is more useful barcode system and it is used by the most manufacture company so we designed our project to recognize the EAN-13 barcode. Another issue of our project is to read the bar of te barcode so that the study of the simple image processing technique is also necessary so we also involved on that

Up to midterm , we did get output of the barcode and display it on client till that day we simply display the EAN- code on a mobile device so further work is to get the information regarding that EAN-13 code so to communicate that mobile with client we used Bluetooth communication that mobile with client we used Bluetooth communication so the study of the Bluetooth communication was also essential similarly another important issue may be creation of database of the information regarding the barcode so we studied and used JDBC.

METHODOLOGY

A) Program Flow:

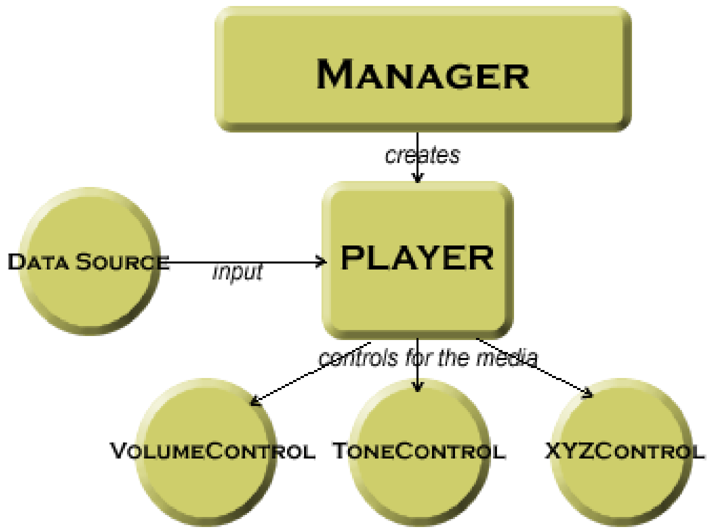
1) Capturing Barcode Image
2) Scanning the image
3) Identifying the string and displaying the string
4) Transferring string to server (Computer) through Bluetooth
5) Reading the necessary data from mysql database
6) Getting data from the server to mobile and
7) Displaying the identified data to user

1) **Capturing Barcode Image*:***

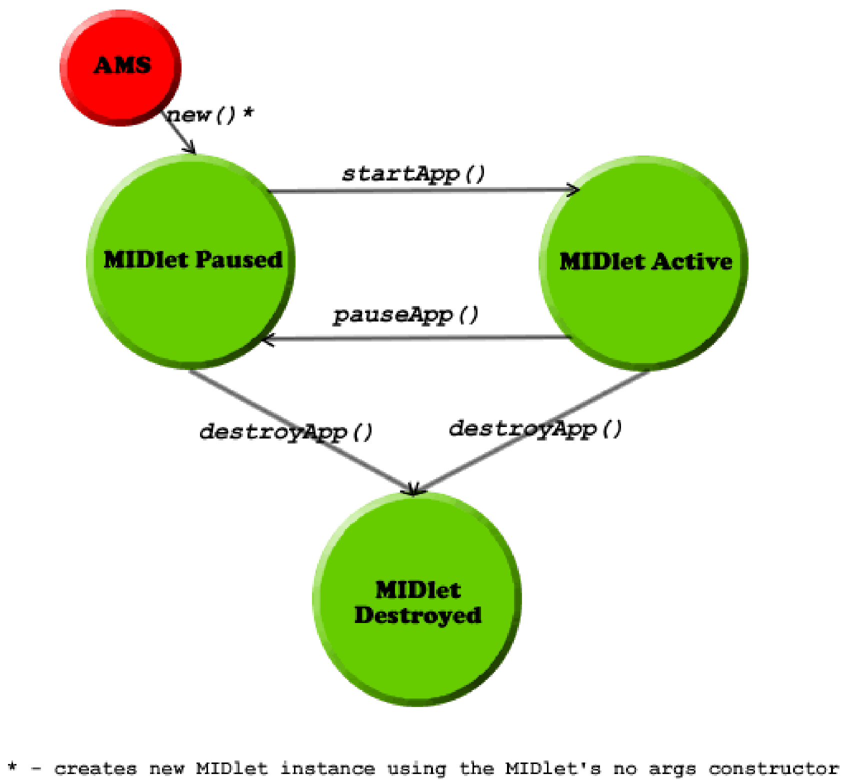
MMAPI defines the superset of the multimedia capabilities that are present in MIDP 2.0. The MMAPI is built on a high-level abstraction of all the multimedia devices that are possible in a resource-limited device. This abstraction is manifest in three classes that form the bulk of operations that we do with this API. These classes are the *Player* and *Control* interfaces, and the *Manager* class. Another class, the

DataSource abstract class, is used to locate resources, but unless we define a new way of reading data we will probably never need to use it directly.

The *Manager* class to create *Player* instances for different media by specifying *DataSource* instances. The Player instances thus created are configurable by using *Control* instances. For example, almost all Player instances would theoretically support a *VolumeControl* to control the volume of the *Player*. The figure below explain the process states :



Once a Player instance is created using the Manager class methods, it needs to go through various stages before it can be used. Upon creation, the player is in an UNREALIZED state and must be REALIZED and PREFETCHED before it can be STARTED. Realization is the process in which the player examines the source or destination media resources and has enough information to start acquiring them.



2.) Scanning the Image: The algorithm and process of extracting the string from EAN-13 barcode image has been explain in the Literature Review of Encoding Barcode. The main function and steps use to explain the string generation has been described in Methodology.

3.) Identifying the string and displaying: After identifying the valid string with the many steps used, we notify the user by vibrating the mobile and at the same time it connect to the server for transferring the string.

4.) Transferring string to server through Bluetooth: The valid string generated by our program is transferred to computer as server via Bluetooth. Here we have used the external library bluecove for this purpose.

5.) Reading the necessary data from mysql: After the EAN-13 number passed to computer server, it is converted into ISBN number. ISBN number used to represent the book information, and our program is especially targeted for reading information from EAN-13 barcode on book. So we converted the

EAN-13 barcode number to ISBN number in server side.

6.) Getting data from server to mobile: The information from the mysql database is extracted with the key value of converted ISBN number. This will extract the book information from database with the field values like name, price, country etc.

7.) Displaying the identifying data in mobile: After all the data extracted from database was transferred to mobile with the help of Bluetooth connection.

B) Scanning the barcode: After capturing the barcode image the following steps generate and identified the valid string:

1. Transferred the captured image into bmp format
2. The converted bmp is transferred to 2D array RGB values.
3. The RGB value transferred to pixel containing B/W only under two steps :
    1. Grayscale conversion
    2. B/W value by Adaptive Threshold algorithm

4. B/W pixels to Fields of B/W

5. Generate the 13 digit barcode number for one scanline

6. Repeat the process for less than or equal to number of scanline

c) Getting Information from server:

1. Connection to server and client
2. Converting EAN-13 to ISBN
3. Extracting data from server and transferring to client
4. Displaying the information on mobile

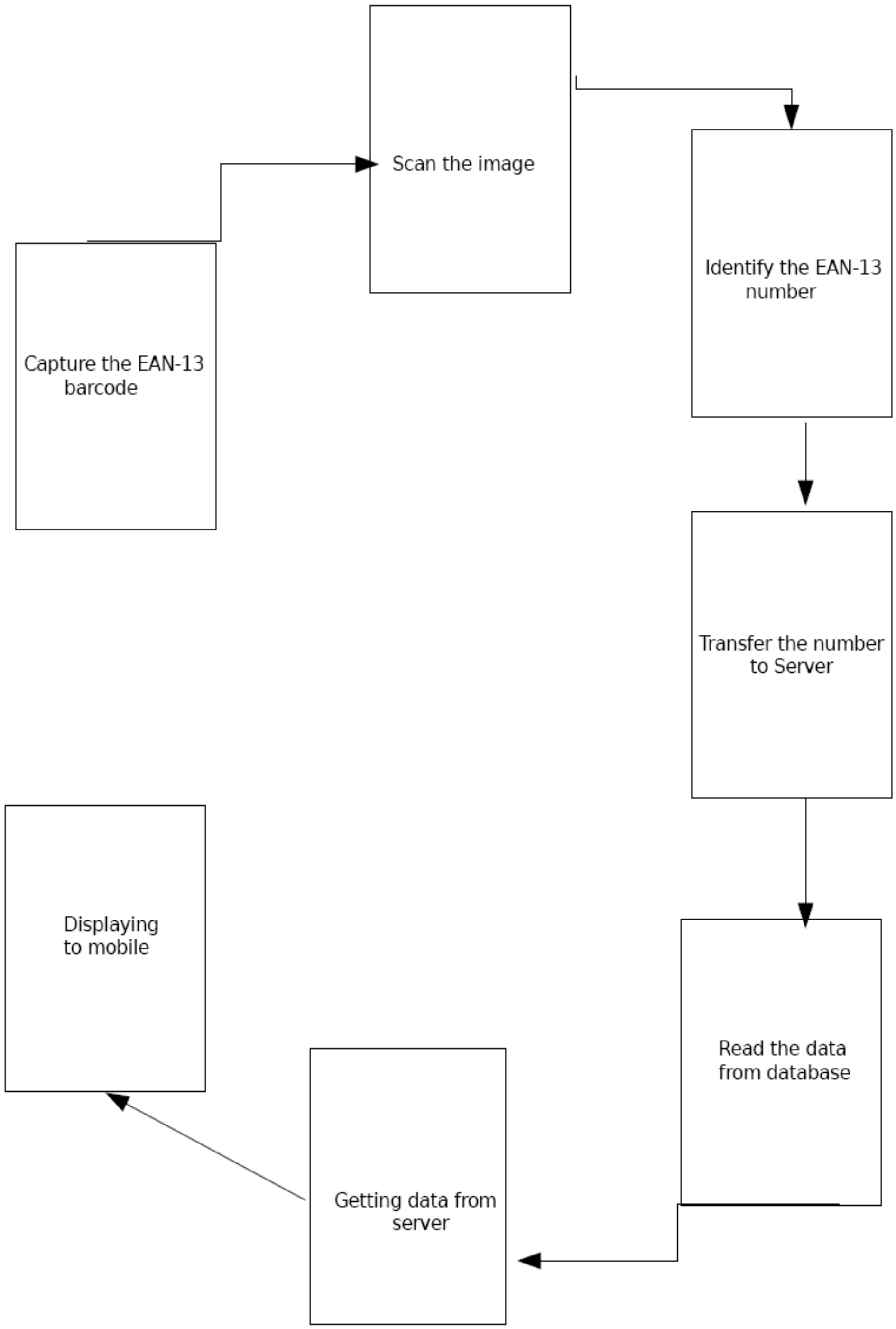


Fig : Process diagram

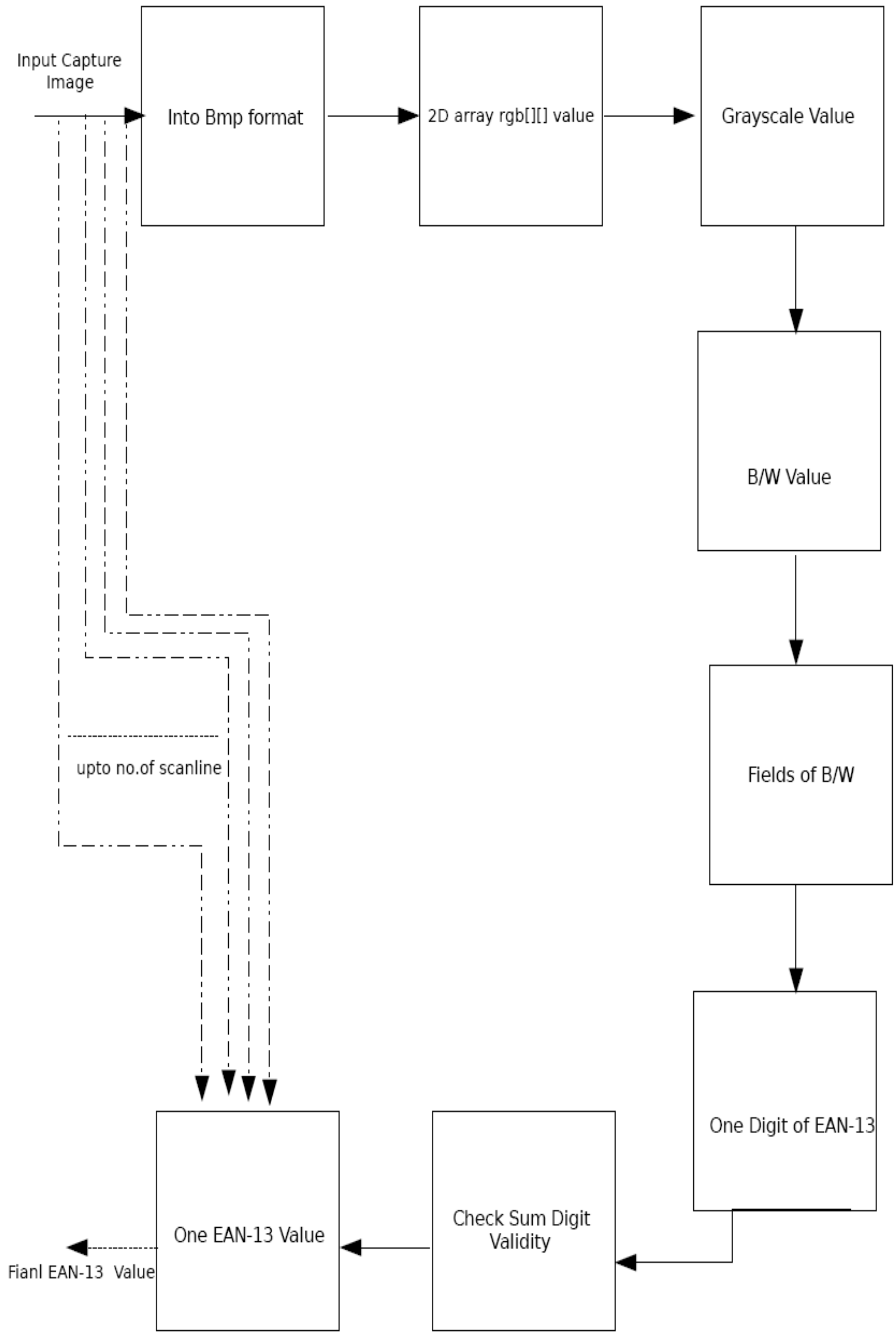


Fig : Process Under Client Side (Generating EAN-13)

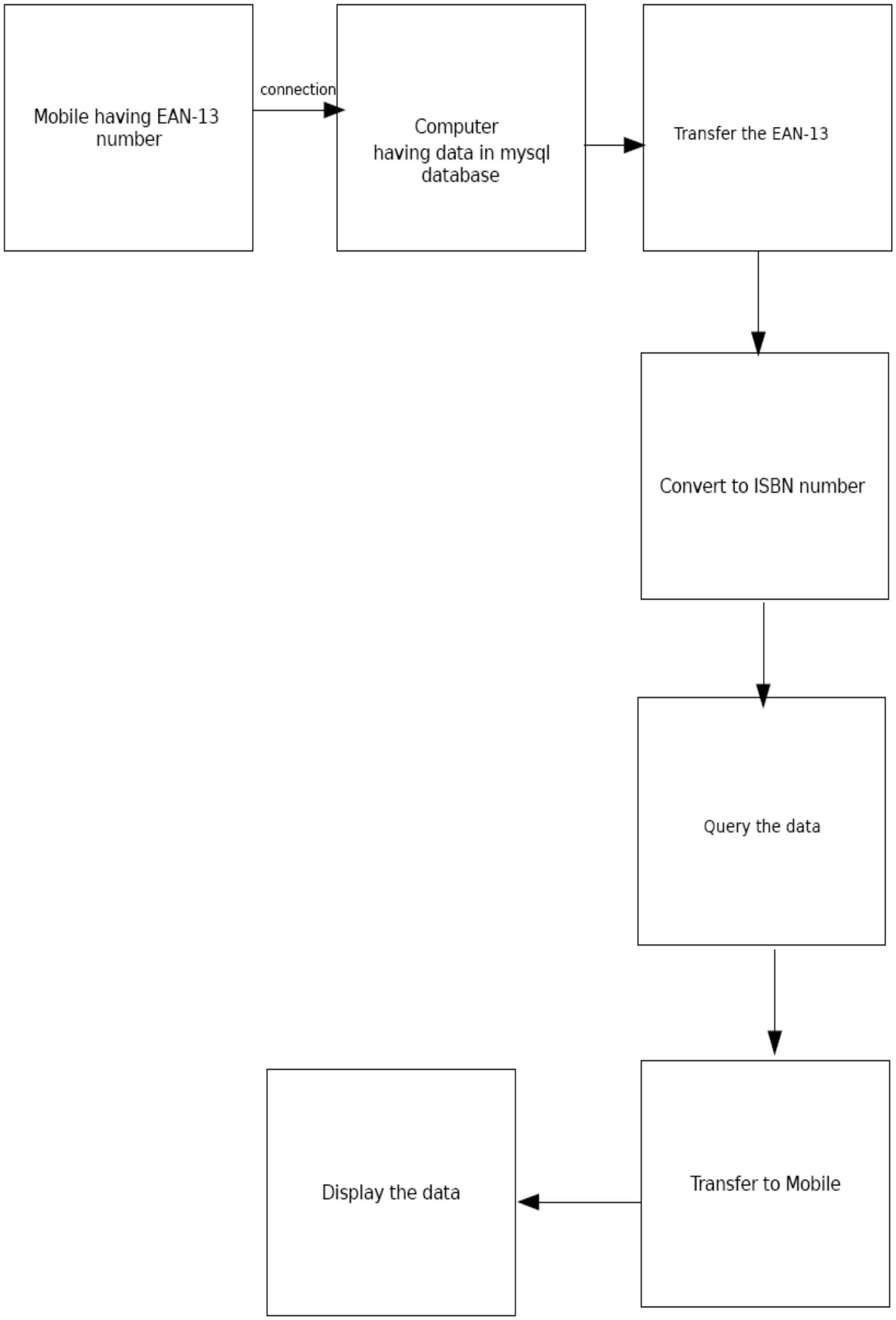


Fig : Process on server side

INSTALLATION PROCESS

Here is basic information for implementing our program. For the implantation in the windows environment .First place the Bluecove 2.0.1 in the library of the server program and transfer and install the executable jar file of client side program in the mobile device.

1) Start the wampserver
2) Run the Barcode server at the file path Dist (in our case:
   D:\mybackup\BarcodeServer\Dist) of the project as
   Java –jar BarcodeServer.jar
3) Run the main Midlet in the mobile client
4) The picture of barcode is to be enlarged if taking photo from camera without auto-
   focus functionality. Otherwise directly take a picture of barcode.

SAMPLE OUTPUTS

Here we present some of output given by the mobile client and the corresponding action and setting of the server. Basically there is no GUI output in the server part.

Our client is designed for input to be taken and output to be displayed both. Server is just for providing the information. Besides that there is some intermediate output response of client which could not be trapped in the picture as they are fast response and changing to other steps.

Here are some of the outputs of our system.

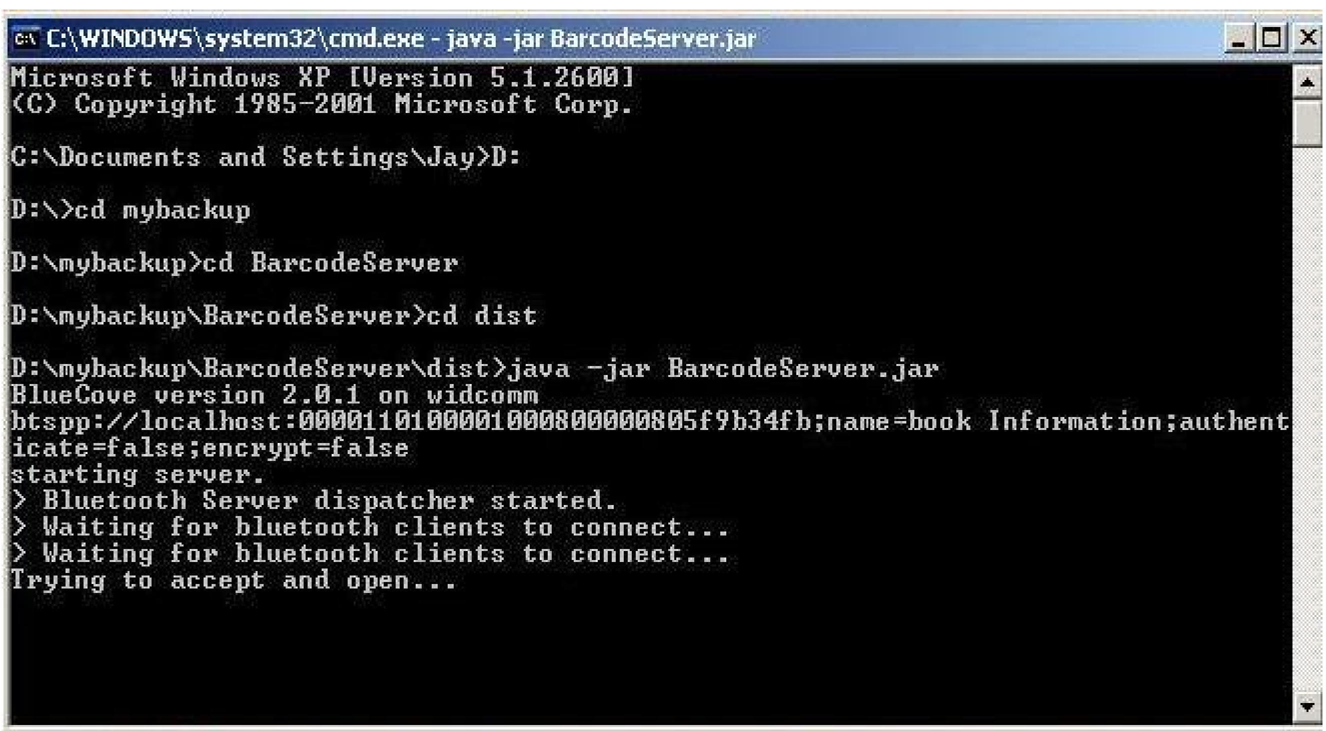


Fig: Server in command line run

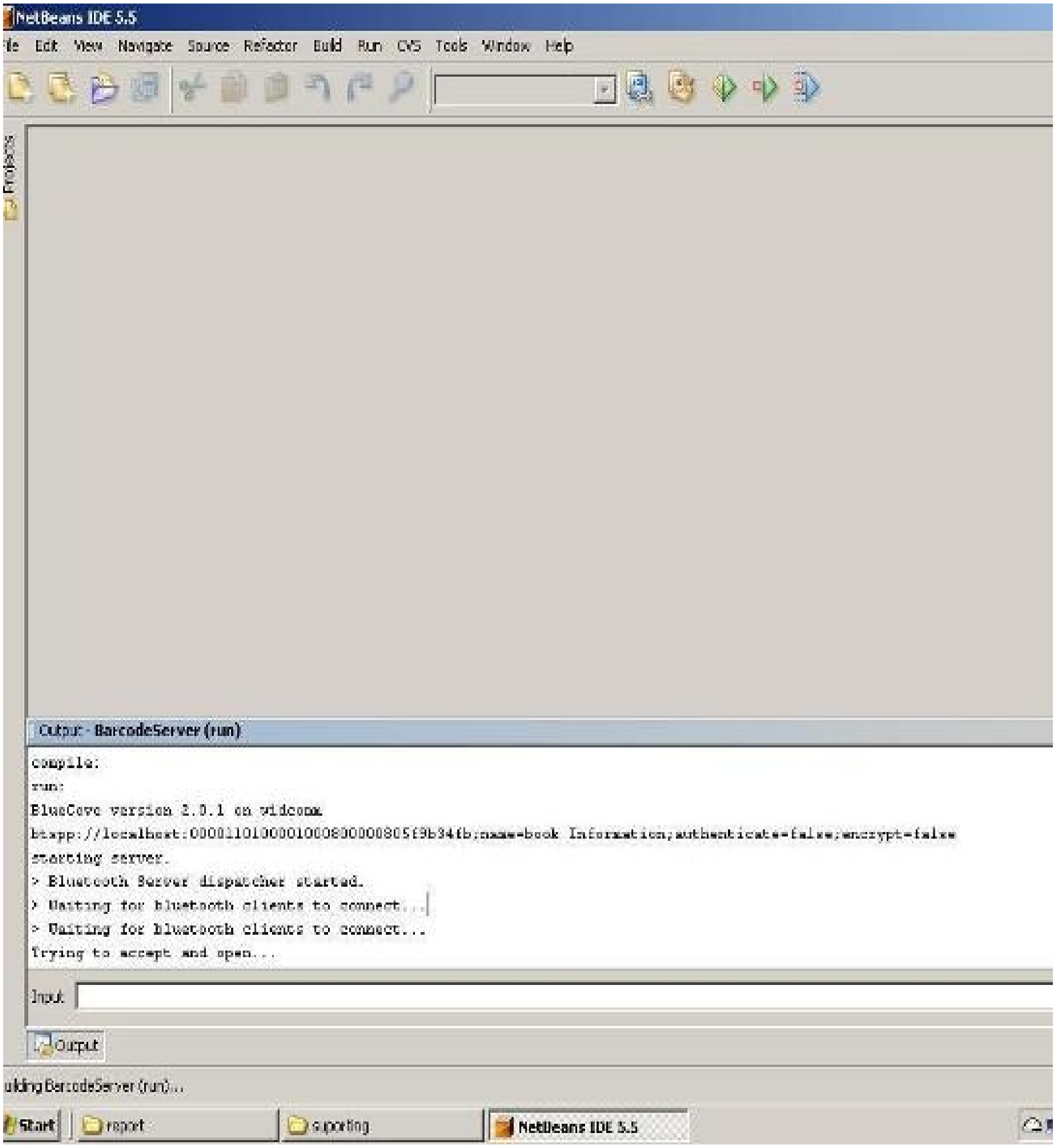
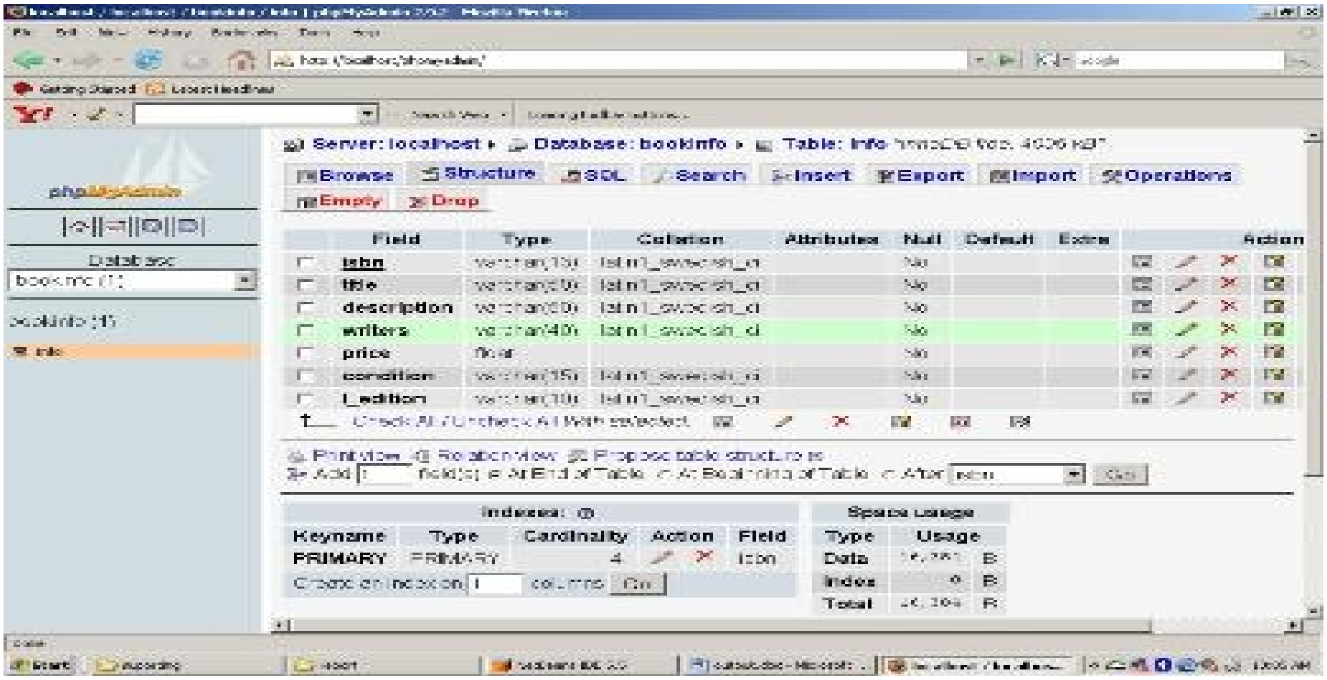


Fig: Server responding to client


Server: localhost
Database: bookinfo
Table: info
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Field
Type
Collation
Attributes
Null
Default
Extra
Action
isbn
title
description
writers
price
condition
l_edition
Indexes:
Keyname
Type
Cardinality
Action
Field
PRIMARY
Space usage
Type
Usage
Data
Index
Total

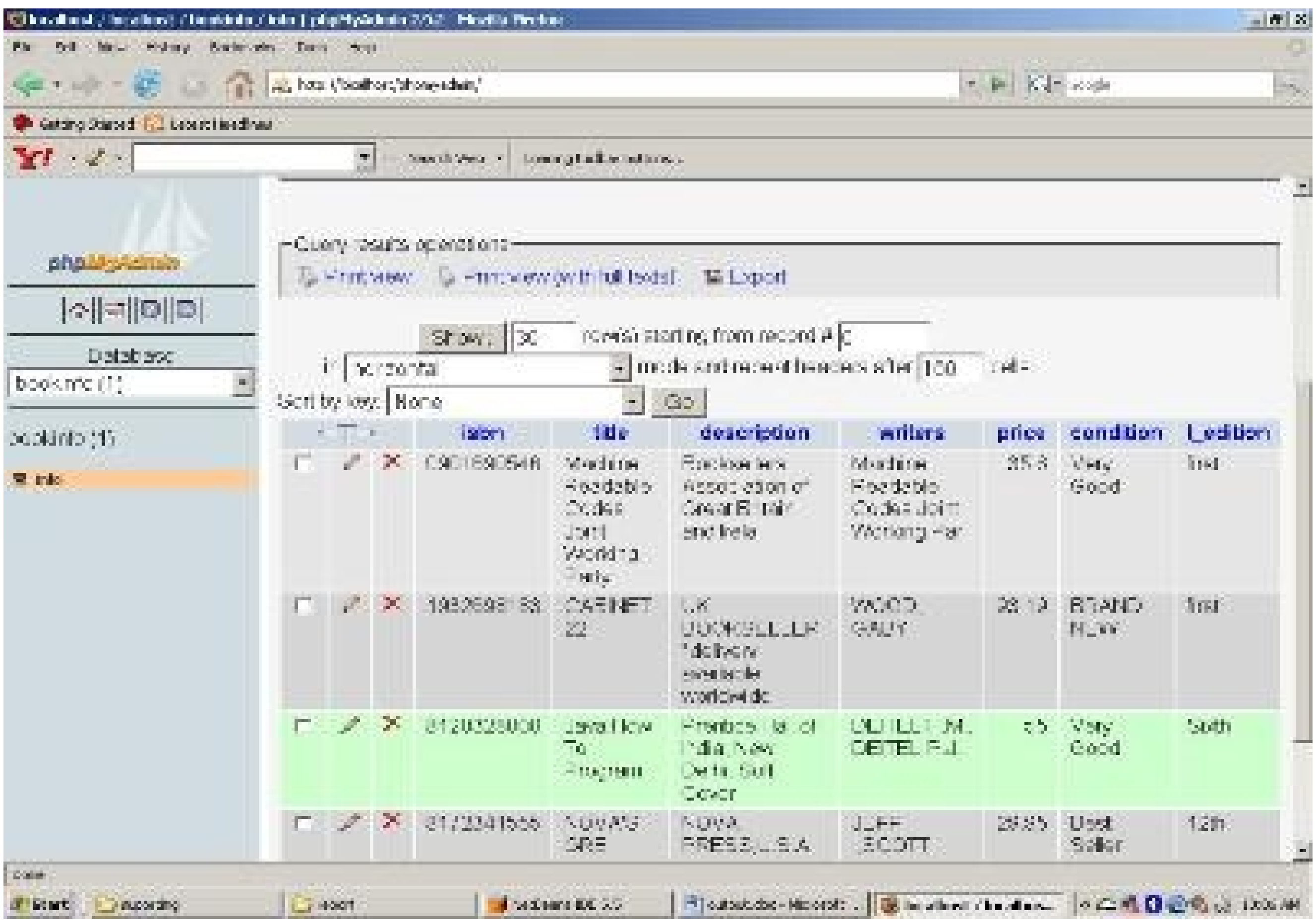

isbn
title
description
writers
price
condition
l_edition

Fig: The database in Mysql

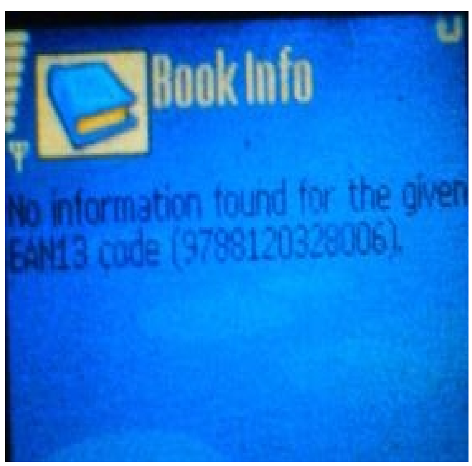


Fig: Client in different stage for giving information

PROBLEM FACED

While doing project we faced problems more than the expectation. The main problem we faced while doing is the thresholding the value of pixel and assign the appropriate white or black image.

We have felt difficulties to deal with image as we have not yet formally studied the image processing courses and research more than coding was needed.

The next problem is while connecting the mobile client and the PC server through the Bluetooth connection. We felt difficulty to define the Bluetooth stack. These are the main two difficulties we faced throughout the program and most of time we were hanged in these two difficulties.

LIMITATIONS

We have analyzed the recognition performance of our algorithm along two axes: focus and image resolution, as these are currently the two most important parameters influencing recognition accuracy on a mobile camera phone. The camera focus directly affects a picture's sharpness. Results indicate that focus remains a problem, while low camera resolutions such as 640x480 pixels are not critical. In order to allow camera phones to scan a bar code from close-up, two options are available. Increasingly, camera phones are being equipped with auto-focus lenses that have been developed over the last several years for example (e.g., the Nokia N90, SonyEricsson's K790 and W810i, or Samsungs SCH series). Ideally, our system would be deployed on auto-focus systems.

But due to unavailability of these device to test we have faced the focusing problem with the set we are using i.e., with N70. The focusing and indirectly the distance between camera and the specimen barcode are the major factor for recognition rate.

Till now our system is only limited to the some model of Nokia phones which supports bmp image format and with auto focus. But auto focus problem is not primary limitation for the demonstration of our project. This will lead the practical problem while in implementation level.

FUTURE ENHANCEMENT

Barcode scanning is the basic requirement for the commercial databases to interact with the unique number of the product i.e.; EAN13 and the barcode of the related product.So our project has the great extendibility for the practical implementation. In this project we have only shown the application example with the book information system. But in fact there is number of application after scanning and sending the output EAN13 number from mobile client to the PC server. These may be for finding the country of production of the product, to find products quality and allergic information, price of the product.By connecting online, consumers can get access to a wealth of information about the product. This includes not only product description and price but also product review, price comparison, location of retailers, etc.

Hence our future work is to use and access the freely available commercial databases in wave such as 4isbn.com, aebooks.com for book information to reduce the need of extra memory and updating mechanism of our standalone server. For this we are researching the methods to access the database of these sites with the java code (in server side).

And other thing is we are going to implement for the vendors mobile phones other than Nokia. The day to day extendibility of mobile phones characteristics themselves suits for our project, which may not till now. So future use of our project is guaranteed by the technology movement in mobile technology.

DISCUSSION AND CONCLUSION

Linking the physical world with virtual information is a powerful concept. We played with this in our project. The resulted project is we can say that robust for decoding the barcode image. Still it has some limitation as we are student and it is our study project.

Hence besides the great difficulties, we completed the project what we proposed to the Department Of Electronics and Computer Engineering, as the third year minor project.

REFERENCES:

1) H. M. DEITEL, P.J. DEITEL; Java: How to Program Sixth Edition; Prentice Hall Of India

2) PDF: Beginning J2ME from Novice to Professionals

3) PDF: Two tone image processing and recognition 4) http://forumnokia.com

5) PDF: Efficient Computation of Adaptive Threshold Surfaces for Image Binarization

6) PDF: Adaptive Thresholding for Digital Desk

7) PDF: J2ME Bluetooth Programming

8) http://java.sun.com/products/mmapi/

9) http://bluetooth .com

10) http://www.barcodeisland.com/

11) http://www.halfbakery.com/