Management Database System Using the Cell Phone

Alharith Alkafije/Department of Computer Science University of Babylon

Abstract:

The database systems are considered as one of the basic elements of modern programming which are the results of decades of development and diligent search. Historically, these systems are among the oldest systems of service that have been developed, so it is a paved road for many of the techniques in the design of systems in order to increase usability, development and credibility in using them in other environments. Although many of the algorithms used in data management systems are material for systematic texts there is little coverage of opinions of design systems that make the data systems work. This research submits an application to the database system and manage by using mobile to control the database system for a school, for example, but this is not limited to the example we chosen but to implement this program in all areas.

It was noticed during conducting this research ,the ease and speed with which the operations of the system were carried out by using mobile-handled devices that are compatible with Windows . This research was carried out by using ….

إدارة قواعد بيانات باستخدام الموبايل

م.م الحارث عبد الكريم عبد الله
قسم علوم الحاسبات /كلية العلوم/جامعة بابل

المستخلص:

تعتبر أنظمه قواعد البيانات من العناصر الاساسيه للبرمجة ألحديثه وهي نتائج لعقود من التطور والبحث المثابر. من الناحية التاريخية ، هذه الانظمه من بين أقدم الانظمه الخدمية التي جرى تطويرها ، وهكذا فإنها تعتبر ممهد طريق لكثير من التقنيات في تصميم الانظمه من اجل زيادة قابليه تطويرها ومصداقيتها في الاستعمال في البيئات الأخرى . بالرغم من ان الكثير من الخوارزميات المستعملة في انظمه إدارة البيانات هي مواد لنصوص منهجيه فان هناك تغطيه ضئيلة أو قليله في أراء تصميم الانظمه التي تجعل انظمه البيانات تعمل .

يقدم هذا البحث تطبيقاً لنظام قواعد البيانات وإدارتها باستعمال الموبايل حيث ندير ونسيطر على نظام قاعدة البيانات لمدرسة على سبيل المثال وليس الحصر ونختاره كمثال لتطبيق هذا البرنامج في كل المناطق .

لوحظ عند تنفيذ هذا البحث سهوله وسرعه عمليات النظام وتحديثها باستخدام أجهزة الموبايل الكفية والتي تعمل مع نظام الوندوز Windows CE، البحث نفذ باستخدام آخر إصدار لفيجول بيسك (فيجول بيسك دوت نت ٢٠٠٨).

1. Introduction

A DBMS is a set of software programs that controls the organization, storage, management, and retrieval of data in a database. DBMSs are categorized according to their data structures or types. The DBMS accepts requests for data from an application program and instructs the operating system to transfer the appropriate data. The queries and responses must be submitted and received according to a format that conforms to one or more applicable protocols. When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system.

Database servers are computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with generous memory and RAID disk arrays used for stable storage. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large volume transaction processing environments. DBMSs are found at the heart of most database applications. DBMSs may be built around a custom multitasking kernel with built-in networking support, but modern DBMSs typically rely on a standard operating system to provide these functions.[1]
The objectives of this project are using Database Management System in the smart mobile where we used the visual basic dot net 2008 that supports the all the applications for the smart phone that has Windows CE.

The system is run on the database for schools where we used the mobile as a server and the other mobiles client and send and receive

the information by the SMS and by the Bluetooth and we noticed the speed updating the system and this is the focal point of the project.

2. Literature Review

Databases have been in use since the earliest days of electronic computing. Unlike modern systems which can be applied to widely different databases and needs, the vast majority of older systems were tightly linked to the custom databases in order to gain speed at the expense of flexibility. Originally DBMSs were found only in large organizations with the computer hardware needed to support large data sets.

1960s Navigational DBMS

As computers grew in speed and capability, a number of general-purpose database systems emerged; by the mid-1960s there were a number of such systems in commercial use. Interest in a standard began to grow, and Charles Bachman, author of one such product, Integrated Data Store (IDS), founded the "Database Task Group" within CODASYL, the group responsible for the creation and standardization of COBOL. In 1971 they delivered their standard, which generally became known as the "Codasyl approach", and soon there were a number of commercial products based on it available.

The Codasyl approach was based on the "manual" navigation of a linked data set which was formed into a large network. When the database was first opened, the program was handed back a link to the first record in the database, which also contained pointers to other pieces of data. To find any particular record the programmer had to step through these pointers one at a time until the required record was returned. Simple queries like "find all the people in India" required the program to walk the entire data set and collect the matching results. There was, essentially, no concept of "find" or "search". This might sound like a serious limitation today, but in an era when the data was most often stored on magnetic tape such operations were too expensive to contemplate anyway.

IBM also had their own DBMS system in 1968, known as IMS. IMS was a development of software written for the Apollo program on the System/360. IMS was generally similar in concept to Codasyl, but used a strict hierarchy for its model of data navigation instead of Codasyl's network model. Both concepts later became known as navigational databases due to the way data was accessed, and

Bachman's 1973 Turing Award award presentation was The Programmer as Navigator. IMS is classified as a hierarchical database. IMS and IDMS, both CODASYL databases, as well as CINCOMs TOTAL database are classified as network databases.

1970s Relational DBMS

Edgar Codd worked at IBM in San Jose, California, in one of their offshoot offices that was primarily involved in the development of hard disk systems. He was unhappy with the navigational model of the Codasyl approach, notably the lack of a "search" facility which was becoming increasingly useful. In 1970, he wrote a number of papers that outlined a new approach to database construction that eventually culminated in the groundbreaking A Relational Model of Data for Large Shared Data Banks.[2]

In this paper, he described a new system for storing and working with large databases. Instead of records being stored in some sort of linked list of free-form records as in Codasyl, Codd's idea was to use a "table" of fixed-length records. A linked-list system would be very inefficient when storing "sparse" databases where some of the data for any one record could be left empty. The relational model solved this by splitting the data into a series of normalized tables, with optional elements being moved out of the main table to where they would take up room only if needed.

In the relational model, related records are linked together with a "key".

For instance, a common use of a database system is to track information about users, their name, login information, various addresses and phone numbers. In the navigational approach all of these data would be placed in a single record, and unused items would simply not be placed in the database. In the relational approach, the data would be normalized into a user table, an address table and a phone number table (for instance). Records would be created in these optional tables only if the address or phone numbers were actually provided.

Linking the information back together is the key to this system. In the relational model, some bit of information was used as a "key", uniquely defining a particular record. When information was being collected about a user, information stored in the optional (or related) tables would be found by searching for this key. For instance, if the login name of a user is unique, addresses and phone numbers for that

user would be recorded with the login name as its key. This "re-linking" of related data back into a single collection is something that traditional computer languages are not designed for.
Just as the navigational approach would require programs to loop in order to collect records, the relational approach would require loops to collect information about any one record. Codd's solution to the necessary looping was a set-oriented language, a suggestion that would later spawn the ubiquitous SQL. Using a branch of mathematics known as tuple calculus, he demonstrated that such a system could support all the operations of normal databases (inserting, updating etc.) as well as providing a simple system for finding and returning sets of data in a single operation.
Codd's paper was picked up by two people at the Berkeley, Eugene Wong and Michael Stonebraker. They started a project known as INGRES using funding that had already been allocated for a geographical database project, using student programmers to produce code. Beginning in 1973, INGRES delivered its first test products which were generally ready for widespread use in 1979. During this time, a number of people had moved "through" the group — perhaps as many as 30 people worked on the project, about five at a time. INGRES was similar to System R in a number of ways, including the use of a "language" for data access, known as QUEL — QUEL was in fact relational, having been based on Codd's own Alpha language, but has since been corrupted to follow SQL, thus violating much the same concepts of the relational model as SQL itself.
IBM itself did one test implementation of the relational model, PRTV, and a production one, Business System 12, both now discontinued. Honeywell did MRDS for Multics, and now there are two new implementations: Alphora Dataphor and Rel. All other DBMS implementations usually called relational are actually SQL DBMSs. In 1968, the University of Michigan began development of the Micro DBMS relational database management system. It was used to manage very large data sets by the US Department of Labor, the Environmental Protection Agency and researchers from University of Alberta, the University of Michigan and Wayne State University. It ran on mainframe computers using Michigan Terminal System. The system remained in production until 1996.
End 1970s SQL DBMS

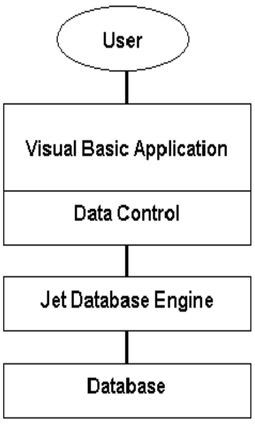
IBM started working on a prototype system loosely based on Codd's concepts as System R in the early 1970s. The first version was ready in 1974/5, and work then started on multi-table systems in which the data could be split so that all of the data for a record (much of which is often optional) did not have to be stored in a single large "chunk". Subsequent multi-user versions were tested by customers in 1978 and 1979, by which time a standardized query language, SQL, had been added. Codd's ideas were establishing themselves as both workable and superior to Codasyl, pushing IBM to develop a true production version of System R, known as SQL/DS, and, later, Database 2 (DB2). Many of the people involved with INGRES became convinced of the future commercial success of such systems, and formed their own companies to commercialize the work but with an SQL interface. Sybase, Informix, NonStop SQL and eventually Ingres itself were all being sold as offshoots to the original INGRES product in the 1980s. Even Microsoft SQL Server is actually a re-built version of Sybase, and thus, INGRES. Only Larry Ellison's Oracle started from a different chain, based on IBM's papers on System R, and beat IBM to market when the first version was released in 1978.
Stonebraker went on to apply the lessons from INGRES to develop a new database, Postgres, which is now known as PostgreSQL. PostgreSQL is often used for global mission critical applications (the .org and .info domain name registries use it as their primary data store, as do many large companies and financial institutions).
In Sweden, Codd's paper was also read and Mimer SQL was developed from the mid-70s at Uppsala University. In 1984, this project was consolidated into an independent enterprise. In the early 1980s, Mimer introduced transaction handling for high robustness in applications, an idea that was subsequently implemented on most other DBMS.
1980s Object Oriented Databases
The 1980s, along with a rise in object oriented programming, saw a growth in how data in various databases were handled. Programmers and designers began to treat the data in their databases as objects. That is to say that if a person's data were in a database, that person's attributes, such as their address, phone number, and age, were now considered to belong to that person instead of being extraneous data. This allows for relationships between data to be relation to objects and their attributes and not to individual fields. [3]

Another big game changer for databases in the 1980s was the focus on increasing reliability and access speeds. In 1989, two professors from the University of Michigan at Madison[l], published an article at an ACM associated conference outlining their methods on increasing database performance. The idea was to replicate specific important, and often queried information, and store it in a smaller temporary database that linked these key features back to the main database. This meant that a query could search the smaller database much quicker, rather than search the entire dataset. [4] This eventually leads to the practice of indexing, which is used by almost every operating system from Windows to the system that operates Apple iPod devices.

3. Methodology

Framework Dot net

For database management, we say our Visual Basic dot net 2008 application acts as a front-end to the database. This means the Visual Basic dot net application provides the interface between the user and the database. This interface allows the user to tell the database what he or she needs and allows the database to respond to the request displaying the requested information in some manner. A Visual Basic dot net application cannot directly interact with a database. There are two intermediate components between the application and the database: the data control and the database engine:



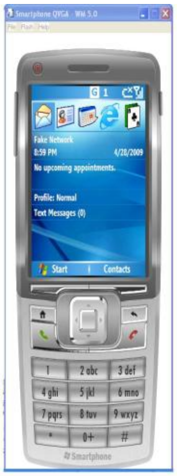
(Figure 1) shows the two intermediate components between the application and the database

The data control is a Visual Basic dot net object that connects the application to the database via the database engine. It is the conduit between the application and the engine, passing information back and forth between the two. The database engine is the heart of a Visual Basic dot net database management system. It is the actual software that does the management. Having this engine saves programmers a lot of work. The database engine native to Visual Basic dot net is known as the Jet engine. It is the same engine used by Microsoft Access for database management. Hence, it is primarily used to work with Access databases, but it can also work with others. As mentioned, the Jet engine will save us lots of work. An observation that illustrates the power of using Visual Basic as a front-end for database management systems: Using Visual Basic, it requires less code to connect to an existing database, view all information within that database, and modify any and all information within that database, than it does to add two numbers together. That's right - all the database tasks mentioned above can be done without writing one line of code! That's the power of the Jet database engine! So, if the Jet engine is so powerful and is the same engine used by Microsoft Access, why not just use Access as a DBMS instead of writing a custom Visual Basic application? There are two primary advantages to using Visual Basic as a DBMS instead of Access: Your users don't need to have Access installed on their computers or know how to use Access. By building a custom front-end, you limit what your user can do with the information within the database. Under normal operation, Access provides no such limits. So, in this course, we will look at how to build Visual Basic applications that operate as front-ends to databases. Research has shown that over half of all Visual Basic dot net applications involve working with databases. We will look at how to make our applications into complete database management systems, being able to view, search, modify, add, and/or delete database information. Before going any further, let's review the steps in building a Visual Basic dot net application and then build a simple application for practice. [5], [6].

The proposed method and practical results:

The practical part deals with a full explanation of the project which represents the management of data basis by using the mobile. The project is worked by using v.b. basic language, Net 2008 (which is considered the last one for v.b. basic). Al-Mahawel school has been used as an example of our data basis. It is also used by all scientific fields that use data basis.

The form is like a mobile during performance. The mobile screen contains the present data and time as in the following figure.

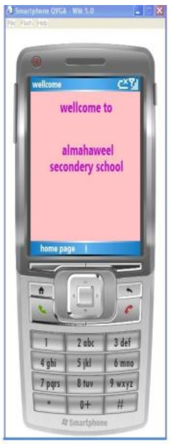


(Figure 2) shows the implementation of program - the mobile screen

When the program is starts, welcome screen will be seen containing the main page which can be got by the following code:

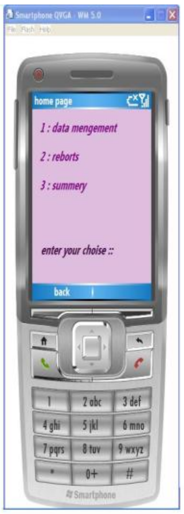
Code:

```
Dim f2 As New Form2
      Me.Hide()
       f2.Show()
```



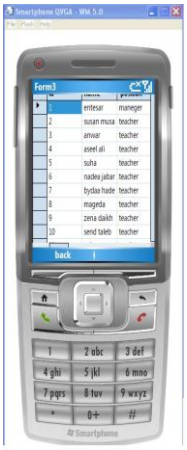
(Figure 3) shows the implementation of program - the welcome screen

By pressing the main page button, we shall get the first page containing the same information in the main page.



(Figure 4) shows the implementation of program - the main page

During selecting any of the three choices above (on the mobile screen), all you do is to press the button which contains the choice number. For instance, pressing on number1, the following page will be seen (the page contains information table and numbers of other choices).

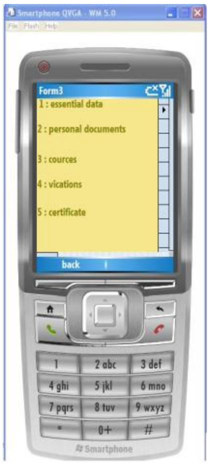


Form3

1	entesar	maneger
2	susan musa	teacher
3	anwar	teacher
4	aseel ali	teacher
5	suha	teacher
6	nadea jabar	teacher
7	bydaa hade	teacher
8	mageda	teacher
9	zena daikh	teacher
10	send taleb	teacher

back

(Figure 5) shows the implementation of program - the table information



Form3

1 : essential data

2 : personal documents

3 : cources

4 : vications

5 : certificate

back

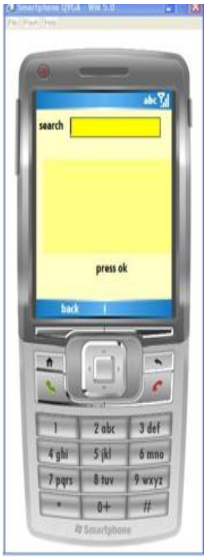
(Figure 6) shows the implementation of program - the transporting pagees

A special page will be on the screen according to the selected number. This happens by using transporting code among the pages, that is:
Code:

```
Me.Hide()

Form__.Show()
```

By using this code and particularly the form number, it can be moved between pages and so on.
Now when number 1 is selected, the following page will be seen; it contains basic data (for instance, writing a name), the especial button will be pressed for performance and information such as the name, job, and qualification will be seen.



(Figure 7) shows the implementation of program - the basic data

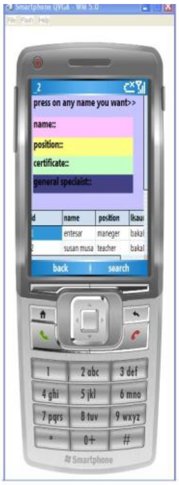
This happens by using the following code:
Code:

```
con.ConnectionString = "Data Source=\Program
Files\SmartDeviceProject2\AppDatabase1.sdf;"
con.Open()
Dim query As String = "select * from t1 where name like '%" +
TextBox1.Text.Trim + "%'"
Dim comand1 As New SqlCeCommand(query, con)
Dim reader As SqlCeDataReader = comand1.ExecuteReader
While reader.Read
Label1.Text &= TextBox1.Text + "    " + reader.GetString(2).ToString&"
"+ reader.GetString(3).ToString+ "  "
End While
```

If we want to go back to the previous page or to any page, we just press the back button which contains the following code.
Code:

```
Me.Hide()
Form__.Show()
```

That code is like a transporting code. (This is particularly used by determining a specific page number).


_2
press on any name you want>>
name::
position::
certificate::
general specialist::

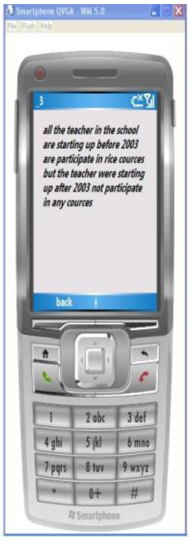
d	name	position	lkau
1	entesar	maneger	bakal
2	susan musa	teacher	bakal

(Figure 8) shows the implementation of program – Choice number two

In determining any circle of a table press "search", and the information fixed above will be filled in by using the following code:
Code:

```
Dim i As New Int16
i = DataGrid1.CurrentRowInde
ss = DataGrid1(i, 1)
Label2.Text += ss
s1 = DataGrid1(i, 2)
Label3.Text += s1
s2 = DataGrid1(i, 3)
label4.Text += s2
s3 = DataGrid1(i, 9)
Label5.Text += s3
```

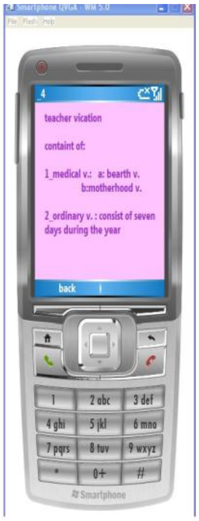
In order to display a course which teachers of the school participated in, no. 3 button will be pressed, and the information will clearly be seen in brief as in the following:



all the teacher in the school
are starting up before 2003
are participate in rice cources
but the teacher were starting
up after 2003 not participate
in any cources

(Figure 9) shows the implementation of program – choice number three

To sum up, to show the school vacations for each teacher in the school, no.4 button will be pressed since the latter refers to the school vacations.



teacher vication

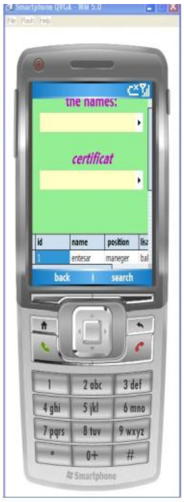
containt of:

1_medical v.: a: bearth v.
b:motherhood v.

2_ordinary v. : consist of seven days during the year

(Figure 10) shows the implementation of program – choice number foure

While pressing no.5, the following page will be seen. This page contains a special information table. It also contains two columns, so by pressing any one; rather than pressing the search button, all columns will be filled in with teachers' names with their qualifications.



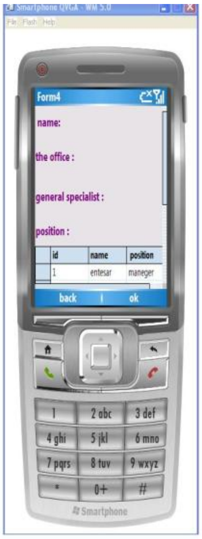
the names:

certificat

id	name	position	lis
1	entesar	maneger	bal

(Figure 11) shows the implementation of program – choice number five

If we want to return to the main page, press no.2 and the following page will be seen.



name:

the office :

general specialist :

position :

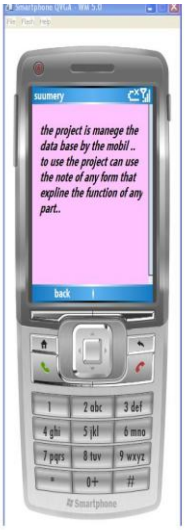
	id	name	position
	1	entesar	maneger

(Figure 12) shows the implementation of program – choice number two

In this paragraph, any column of a table is selected and by pressing its button, the column fixed above will be filled in. This is done by using the following code:
Code:

```
Dim i As Int16
i = DataGrid1.CurrentRowIndex
z = DataGrid1(i, 1)
Label1.Text += z
z1 = DataGrid1(i, 8)
label2.Text += z1
z2 = DataGrid1(i, 9)
Label3.Text += z2
z3 = DataGrid1(i, 2)
Label4.Text += z3
```

By pressing no.3 in the main page, the following page will be seen.


suumery

the project is manege the data base by the mobil .. to use the project can use the note of any form that expline the function of any part..

(Figure 13) shows the implementation of program - the project abstract

While refers to the project abstract and simply examplifies it.

4. Conclusion

By using an application developed for mobile phone devices, we are able to talk about mobility concept when monitoring a system – anytime and anywhere impediment. Limited recourses to such a device are not really an implement if we are implementing as much as possible the proceeding tasks at application server side.

5. References

1. **Taylor A., JDBC-database Programming on the Internet, Informix Press, 1997.**
2. **Codd, E.F. (1970)."A Relational Model of Data for Large Shared Data Banks". In: Communications of the ACM 13 (6): 377–387.**
3. **Development of an object-oriented DBMS; Portland, Oregon, United States; Pages: 472 - 482; 1986; ISBN: 0-89791-204-7.**
4. **Performance enhancement through replication in an object-oriented DBMS; Pages 325-336; ISBN: 0-89791-317-5.**
5. **Mathew MacDonald, Beginning ASP dot net 3.5 in VB 2008, 2008.**
6. **Bill Evjen, Scott Hanselman & Devin rader, professional ASP dot net 3.5 in C# and VB, 2008.**