

Developing a Mini Food E-additive Database for Mobile Telephones

Andras Sik
*Prolan Zrt., Budakalasz,
Hungary*

andras.sik@gmail.com

Gergely Sik
*Budapest University of
Technology and Economics,
Budapest, Hungary*

sikgeri@gmail.com

Cecilia Sik Lanyi
University of Pannonia, Veszprem, Hungary

lanyi@almos.uni-pannon.hu

Abstract

Our project has developed an application for mobile telephones to provide a catalogue of E-numbers and E-names: the food additives widely known as “E-numbers”. Our application, the *Mini Food E-Additive Database (MIFED)* provides a user interface that is friendly and easy to use. To achieve our main goal of platform independence we applied the JAVA programming language to implement our software. Thus, based on JAVA our software can run on a very wide range of mobile devices. The development process of the software is outlined in this paper.

Keywords: mobile technology, database, food additive

Introduction

Almost everyone owns a mobile phone nowadays, and while these phones vary in both their cost and capabilities, they do offer some common features, and the new intelligent mobile phones have advanced capabilities, often with PC-like functionality. The goal of our project was to develop useful, user-friendly software using mobile technology to help consumers to make informed choices about food. After all, what options does the average consumer have at a supermarket if they want to know what an E-number or food additive actually is? Our software was developed to resolve this problem, a development motivated both by the food scandals so frequently reported in the media, and by a question: what are E-numbers?

In fact, E-numbers are simply a systematic way of identifying different food additives. Before a

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

food additive is given an ‘E-number’ it must be cleared by the European Food Safety Authority (EFSA) to show that it can be used safely in foods. An ‘E-number’ is only given to an additive when it has passed all the required safety checks (Current EU approved additives and their E Numbers, n.d.). So **E numbers** are number codes for food additives that have been assessed as safe for use within the European Union.

They are also encountered on food labelling in other jurisdictions, including Australia, New Zealand and Israel although the 'E' prefix is omitted in Australia and New Zealand. They are increasingly (albeit still infrequently) found on the packaging of imported European products in North America and especially Canada ("E_number," n.d.).

E-numbers cover an enormous range of additives, ranging, for example, from E100 - the natural spice and yellow food colorant turmeric - to E 110 (also known as 'sunset yellow'), a chemical that has been implicated in child behavior disorders, and as such its use is restricted or banned in at least some food groups in 3 European countries although not yet in all. Mizsei (2008) stated that in Hungary the number of the children affected by autism has increased from twenty thousand to eighty thousand. A fourfold increase such as this cannot be ignored neglected! Mizsei explained this fact by the various organic chemicals which accumulate in the bodies of mothers by consuming low quality food and water, for example, food additives.

Human beings have a right to consume healthy food! Currently, the potentially hazardous additives mixed into our daily food are frequently a topic of discussion ("E-Numbers," n.d.). People living in urban areas are largely obliged to consume what they are able to obtain from retail outlets, and unfortunately such food contains many additives. Our goal in creating this application was not to judge, however, but rather to help people to shop in a purposeful way. There are hundreds of E-chemicals, far too many to memorise, and given that it is almost impossible to avoid the use of food additives nowadays, present as they are in almost every kind of food sourced through industrial production, MIFED presents us with an opportunity to consume the food that we would like to, food that does not tax the body excessively, if at all. We find it outrageous that innocent people can be poisoned by companies for their own financial and business gain in such an irresponsible way.

As it is, to put it simply "we are fed" with all kinds of things that others want us to eat, which is why the acronymic project and product title is "MIFED" (pronounced: **Am I fed?** – with E food additives, that is). This perfectly reflects the fact that nowadays people rarely have any idea of what they are eating. One of the principle concepts behind the development of our user-friendly software for our **MI**ni **F**ood **E**-Additive **D**atabase (**MIFED**). was the notion that the complexity of handling a mobile device is likely to reduce its ease of use and thus its perceived usefulness to the user (Ziefle et al., 2006; Ziefle & Bay, 2004).

Development and Architecture of the Software

During the implementation of the software we considered platform independence to be our ultimate goal, and consequently the implemented software will be able to run on a wide range of mobile devices, and thus aid as many people as possible. With this goal in mind, and having considered the many other less-widely recognised and used programming languages, the Java environment seemed the perfect choice because it is itself a platform-independent programming language. The Java programming language is currently available for almost every mobile telephone device, and needless to say its basic usage and programming is very easy to learn. Platform independence itself means that implemented Java source code is guaranteed to run in the same way on every mobile device with a Java interpreter.

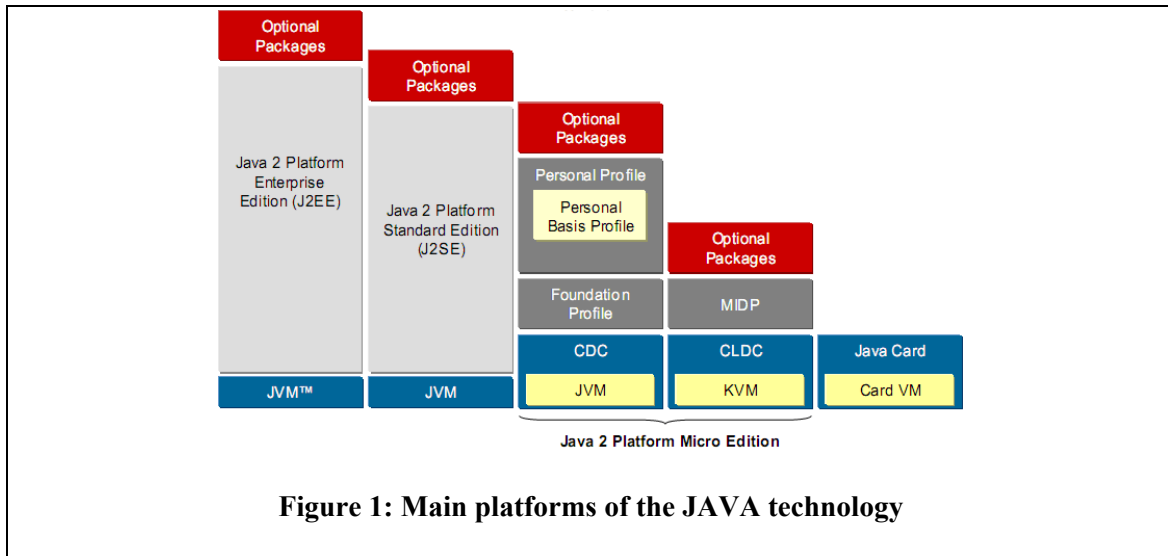
The Java platform itself can be divided into four smaller platforms: JEE¹, JSE², JME³ and Java Card⁴. Of these four platforms JME (formerly known as J2ME) was most important to us in

1 Java Platform, Enterprise Edition or Java EE is a widely used platform for server programming in the Java programming language.

2 Java Platform, Standard Edition or Java SE is a widely used platform for programming in the Java language.

providing a platform for programming mobile devices with limited resources (small screen, limited memory and power supply). The main platforms of the Java technology are demonstrated in Figure 1.

The diversity of the JME platform—and our goal of creating an application that would be usable on most mobile devices—led us to opt to use those parts of the JME runtime environment (JRE⁵) that are available on most Java enabled devices. Thus the application was created on the base of the CLDC⁶ 1.0 (<http://java.sun.com/javame/reference/apis/jsr030/>) configuration and the MIDP⁷ 2.0 (<http://java.sun.com/javame/reference/apis/jsr118/>) profile, leaving other optional packages (so-called JSR⁸-s) out of the loop.



Consequently, the program runs in mobile environment and one of our main goals was to make it as resource economic as possible by using algorithms with the shortest available runtime.

The way in which the necessary data would be represented was something else that had to be considered. Storing the data on additives anywhere other than in the mobile units was not an option for us because the memory of the mobile device is constantly accessible, generates no network traffic when used, costs nothing and provides excellent off-line storage. Given our goal of creating a program with the highest possible compatibility among all known devices, storing the used data in files separate from the software was not feasible because it would have meant

3 Part of the Java platform which provides variants of the various Java Application Programming Interfaces (API-s) implemented for devices with limited resources.

4 Java Card refers to a technology that allows small Java-based applications (applets) to be run securely on smart cards and similar small memory footprint devices. Java Card is the tiniest of Java targeted for embedded devices.

5 The Java Runtime Environment (JRE), also known as Java Runtime which provides the minimum requirements for executing a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files.

6 The Connected Limited Device Configuration (CLDC) is a specification of a framework for Java ME applications targeted at devices with very limited resources such as pagers and mobile phones.

7 Mobile Information Device Profile (MIDP) is a specification published for the use of Java on embedded devices such as mobile phones and PDA-s which sits on top of (CLDC).

8 The Java Specification Requests are formal documents that describe proposed specifications and technologies for adding to the Java platform.

utilising the optional JME File Connection API (http://www.forum.nokia.com/document/Java_Developers_Library_v2/) package which evidently we could not do.

Clearly then, the only possible form of data representation was that of data storage within the program itself. Based on the limited resources of the various mobile units we felt that the quickest possible implementation of data access would be very important. Java implemented hash tables fulfilled this requirement by providing a constant $O(1)$ time for data access, the quickest means of retrieving data. We built up two hash tables, one of E-numbers and another of E-names as keys in order to guarantee access to all the information based on either the numbers or the names of the additives.

Unfortunately, the implementation of the hash tables did not include the feature of storing data in an ordered way, although such a display of information was a basic requirement of the application. So, from the outset of the program we applied a Quicksort ordering algorithm for the keys of the hash tables (which were also stored separately in arrays) in order to arrange them in an ascending order. This algorithm itself required $O(n \log n)$ time to order the arrays in which n is the number of components to order, thus it is one of the fastest ordering algorithms. We used these ordered arrays to fill up the choices, and with that browsing of the catalogue could begin. The WPA database provided various pieces of redundant information that were stored only once and were displayed at every appropriate food additive, using an appropriate programming logic in order to consume less storage space.

A search module was an important addition since there were a great many E-numbers and E-names, which made browsing among the choices very awkward (especially for additives at the bottom of the list). Once a search has been performed the search module determines the type of the current search, which means the type of key used for searching (E-number or E-name). When searching for an additive number the search key is used as a key in the corresponding hash table to retrieve information which can be completed in $O(1)$ time. When the search is for additive names, then they are first mapped to numbers using the corresponding hash table (requiring $O(1)$ time) and then the obtained key is used to perform a substring based search among the keys of the other hash table, which search then builds a result set. This kind of search requires $O(n)$ time to be executed, which is the absolute minimum to iterate over the keys.

The application consists of three javax.microedition.lcdui Form elements, which implement the display of the software. These three Form elements are mainForm, infoForm and searchForm. All three forms described above use simple JME GUI elements such as StringItem, TextField and ChoiceGroup [9].

The mainForm serves as a display screen for the main menu, the searchForm for the search results, and the infoForm for the additive information.

After starting up, the user sees the software's main menu (Figure 2.) from which he or she can choose between the three main functions of the application: to browse data based on E-numbers, to browse based on E-names, and to search among the information stored in the program.

Under the **E-numbers** drop-down menu (Figure 3) the E-numbers themselves can be browsed. Once a number is selected, all the information that is stored about a specific additive in the database is presented.

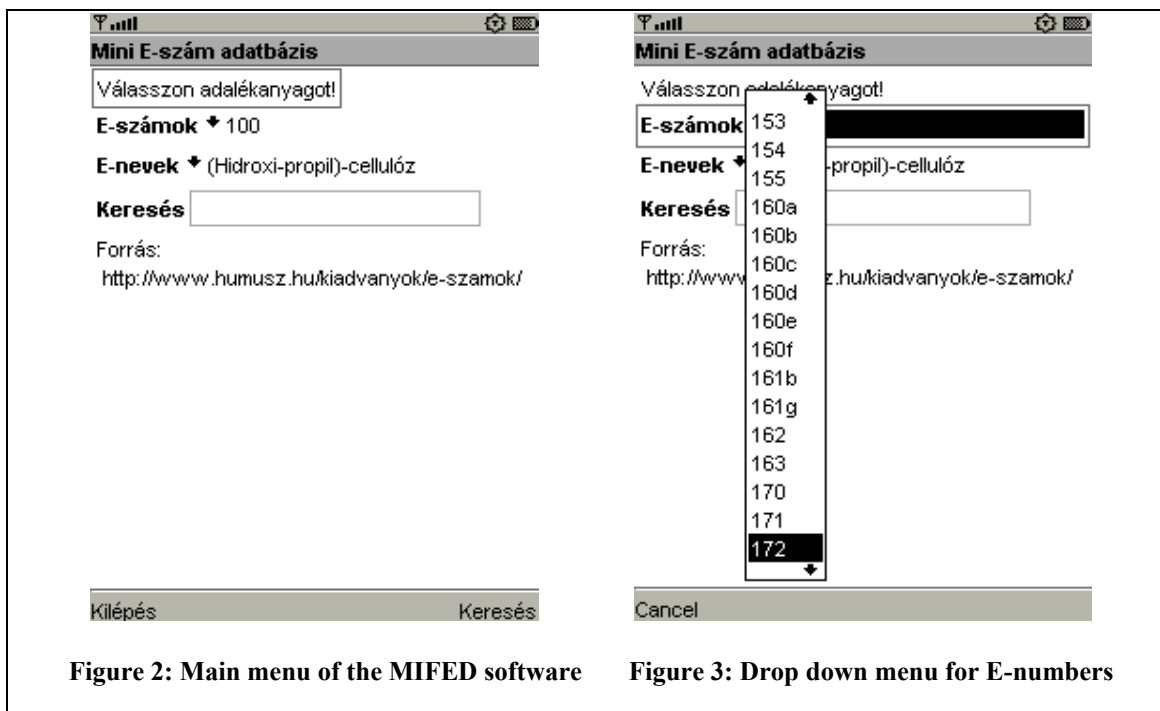


Figure 2: Main menu of the MIFED software

Figure 3: Drop down menu for E-numbers

The same is true for **E-names** (Figure 4.) except in that while we are able to browse the E-names themselves the information accessible is the same as in the previous instance.

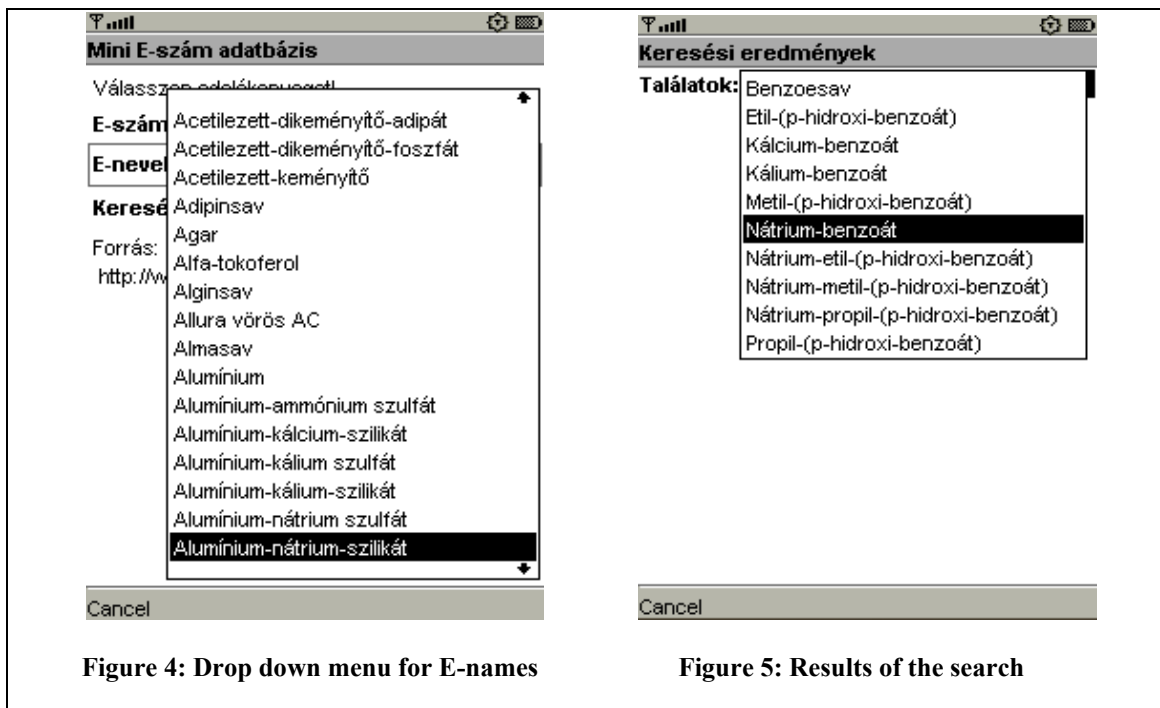
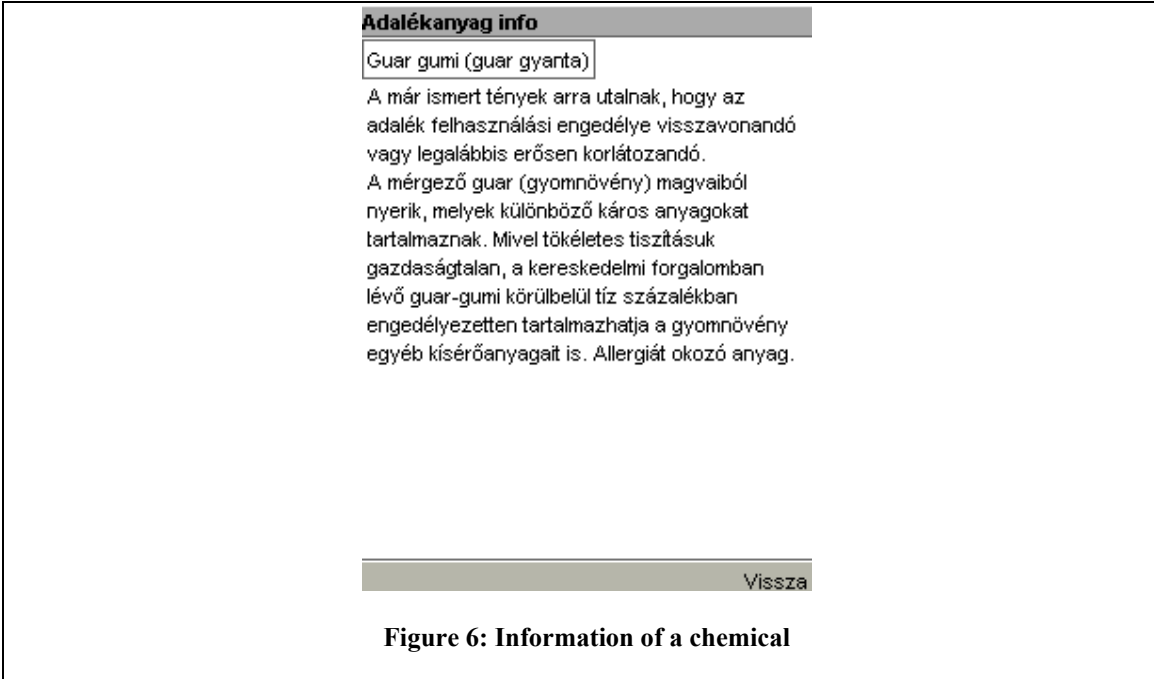


Figure 4: Drop down menu for E-names

Figure 5: Results of the search

The **Search** function can access any information that fits the search conditions in two particular cases. If the search is for an E-number then the program tries to show the user the information on the specific additive (identified by the number keyed in).

If the search is based on the name of an additive, then the program performs a substring based search.



Adalékanyag info

Guar gumi (guar gyanta)

A már ismert tények arra utalnak, hogy az adalék felhasználási engedélye visszavonandó vagy legalábbis erősen korlátozandó. A mérgező guar (gyomnövény) magvaiból nyerik, melyek különböző káros anyagokat tartalmaznak. Mivel tökéletes tisztításuk gazdaságtalan, a kereskedelmi forgalomban lévő guar-gumi körülbelül tíz százalékban engedélyezetten tartalmazhatja a gyomnövény egyéb kísérőanyagait is. Allergiát okozó anyag.

Vissza

Figure 6: Information of a chemical

Consequently, it is not necessary to know the exact name of all the additives, and it is easy to avoid any problems that might originate from different naming conventions. The type of search is automatically decided by the program. Once the search is completed a drop down menu appears that offers the possible results (name search case) or information on a single additive (number search case). If no results are found then notification of this is displayed on the screen.

Summary and Possible Features to Implement

The software is currently implemented in Hungarian, but support for other languages could easily be implemented if a detailed enough description of the E-names and E-numbers is available in the target language.

We intend to implement MIFED software in most languages and with a few minor adjustments the software would be able assist people with specific diseases (e.g. diabetic or kidney disease) in order to minimize their protein or sugar intake, for example.

Our project has implemented software to run on mobile telephone devices that provides a catalogue of E-numbers and E-names. **We hope that this project will contribute to a healthier lifestyle for all its users!**

Acknowledgements

The authors would like to thank the Hungarian Intellectual Alliance for their considerable help in the completion of this project, and also for the 3rd prize received by Gergely Sik at the 18th Young Scientist Contest in 2009, organized by the Hungarian Association for Innovation (http://www.innovacio.hu/en_index.html and http://www.innovacio.hu/3a_hu_18_vegeredmeny.php).

References

- Arning, K., & Ziefle, K., (2007). Barriers of information access in small screen device applications: The relevance of user characteristics for a transgenerational design. In C. Stephanidis & M. Pieper (Eds.). *User Interfaces For All: Universal Access in Ambient Intelligence Environments*. LNCS 4397. Berlin, Germany: Springer, 117-136.
- Current EU approved additives and their E Numbers*. (n.d.). Food Standards Agency. Retrieved from <http://www.food.gov.uk/safereating/chemsafe/additivesbranch/enumberlist>
- E-number*. (n.d.) Wikipedia. Retrieved from http://en.wikipedia.org/wiki/E_number
- E-Numbers: The good, The bad, The facts*. (n.d.). Explore E Numbers. Retrieved from <http://www.exploreenumbers.co.uk/>
- Mizsei, B. (2008). *Egyre több az autista* [There are more and more autistic people]. *Magyar Nemzet*, LXXI (308), 1-4.
- Ziefle, M., & Bay, S. (2004). Mental models of a cellular phone menu. Comparing older and younger novice users. In S. Brewster & M. Dunlop (Eds.), *Mobile human computer interaction* (pp. 25-37). Berlin, Germany: Springer.
- Ziefle, M., Bay, S., & Schwade, A. (2006). On keys' meanings and modes: The impact of different key solutions on children's efficiency using a mobile phone. *Behaviour & Information Technology*, 25(5), 413-431.

Biographies



Andras Sik studied computer science (M:S.) at the Budapest University of Technology and Economics in Budapest (2009). Now he is software engineer at the Prolan Zrt. in Budakalasz, Hungary.



Gergely Sik is BSc student at the Budapest University of Technology and Economics in Budapest.



Cecilia Sik Lanyi is associate professor at the University of Pannonia. She studied Mathematics and Computer Science (M.S.) – József Attila University, Szeged (1984), and teacher of mathematics (Berzsenyi Daniel Teacher Training College in 1988). She obtained the degree of Dr. Univ. at the University of Veszprem, Hungary in Physicalchemistry (1993), and of PhD at the University of Veszprém, Hungary in Computer Science (2000). She reads and leads laboratory studies on Multimedia, Virtual Reality, Programming the WEB, User Interface Design, Computer Graphics and Design. Her research area is in Multimedia, Virtual Reality, Human Computer Interaction, Design for All. She was the secretariat manager of EDeAN (www.edean.org) in 2009.