A(H1N1) Suspects Management Application

DAN ADRIAN MARIOR, RADU ZGLIMBEA, CONSTANTIN CÂRCIUMARU
Department of Automation and Department of Computer Science
University of Craiova
B-dul Decebal, nr. 107, 200440, Craiova
ROMANIA
marior@automation.ucv.ro http://www.ace.ucv.ro
radu@automation.ucv.ro http://www.ace.ucv.ro
ccircium@ford.com http://www.ford.com

Abstract: This paper deals with the differential diagnostics process behind the verdict of AH1N1 infection or not for the swine flu suspects. The application designed as a support for the doctors in an Infectious Institute in Romania was built with ASP.NET 3.5 technology, the broad spectrum development platform, and it enabled us to create a web application that could be accessible from anywhere on the internet, assist the doctor in the diagnostics process, filter out false positives, manage the patients, and also generate statistical charts.

Key-Words: A(H1N1), application, diagnostic, management, charts

1 Introduction
During the past year we have seen an outbreak of swine flu in many countries of the world, and in Romania those who are suspected of the disease can put their hope for reliable diagnostic and treatment in the hands of the capable doctors from the “Matei Balș Infectious Diseases Institute” in Bucharest.

Transmission of the swine origin influenza virus is thought to occur in the same way as seasonal influenza. Human-to-human transmission occurs by inhalation of large infectious droplets as well as by direct contact with secretions or aerosols. At present, there is no evidence of spread of infection by eating pork, or through water, thus the psychological effect is greater than reality and the name of the disease is not so accurate [1].

In Romania the pandemic psychological impact effect has been quite large, and family doctors could not efficiently manage it, so it was and still is up to the “Matei Balș Infectious Diseases Institute” to impose appropriate measures and reliably diagnose and keep statistics of the persons suspected of contracting the virus. The false positives (those who contracted the common influenza or some other infection) are very hard to detect and filter out, as the symptoms are similar (fever, chills, nausea, vomiting, body aches, lethargy, and fatigue, which usually appear in rapid succession). As it has already been noted from practice, the new flu can lead to death by respiratory failure and other causes like sepsis.

2 Problem Formulation
The doctors entrusted with the diagnostics and treatment of AH1N1 influenza in Bucharest are overwhelmed with the number of patients accusing the symptoms mentioned earlier in the paper, thus an application that can help the doctors organize, filter out the unnecessary and drawing statistical and management conclusions would be most helpful.

Management is in general a complex problem, but in this situation, given the fact that it was a national health problem, the application would have to be fairly reliable, easy to use, highly available and dependable, the requirements of all critical applications.

Security measures had to be imposed by design, as no other person besides the doctors and the administrator of the application should have the right to access the resources the application was supposed to store and process. Programmatically, this can be done by specific permissions on files and folders, or user based access, without restrictions of the operating system the application runs on. Further in the paper we indicate how these security measures were imposed in this case.

3 Problem Solution
The ASP.NET 3.5 platform was chosen in order to offer a solution to the inherent problem, being the most appropriate for the facilities it offers, such as independence from hardware architecture (the application runs in a browser on the client machine, this being the only compulsory requirement to run
it), very good database connectivity (Active Data Objects, ADO.NET), membership and role management, personalization, site navigation, themes, master pages and other advantages; a positive aspect of implementing such a solution is that the application can be accessed via a browser from anywhere in the world, through the Internet, and if security measures are correctly imposed, the great advantage over the alternative solutions is clear.

Other positive aspects of the ASP.NET choice are: reduced amount of necessary code for writing large applications, Windows built in security and per-application configuration (making applications safer and more secure), better performance due to early binding, just-in-time compilation, native optimization, and caching services, language independence, simplicity, pure server side technology, process monitoring and others.

The architecture for the application was naturally chosen to be client/server. Client/server architectures divide applications into two or more components. The client portion uses the functions of the server; in most cases, separate hardware systems are used for clients and servers. “The distribution of the application load across several computers linked in a network keeps the individual units relatively favourable” [3].

For reasons of scalability and optimal distribution of computing resources, the three layer resources distribution approach (business logic layer, database layer and presentation layer) has prevailed as a viable foundation for companies of all dimensions; the former phrase is summarizing the revolution that made central mainframe computing abilities available to more organizations, no matter how wealthy.

![Main application page (after login)](image)

---

**Fig. 1 Main application page (after login)**
Fig. 1 Monthly evolution

Fig. 2 Patient number evolution
Fig. 3 Differential diagnostics
The first page of the application is the login page, where the doctor can insert his credentials and after being authenticated (his access to resources is determined, for example he can have administrative rights, or, for the sake of the integrity of the data base, a smaller amount of control could be assigned).

The application was designed as user friendly as possible, with explanations at every moment of the execution. The next step for the doctor is to insert the patients in the database with their presumed diagnostic – it is worth noting that a diagnostic is confirmed only after thorough clinical examination and paraclinical examinations (HNN analysis, which can point accurately to the virus strain, whether it is A(H1N1) or another).

A complex logic (business logic layer) to determine, of course with the help of the doctor (who decides on the need for certain laboratory tests), whether the patient is suffering from A(H1N1) flu or the case is a false positive (any other condition). This is accomplished by continuing to the page where the application collects all the data from the patient (age, symptoms in the beginning, actual manifestations, whether it is an infection or not).

The main advantage of the approach is that by including the possibility for the doctor to correct data inserted, the integrity and reliability of the data are increased, not to mention obtaining a more natural implementation.

The main symptom which makes the doctor consider a possible infection is fever. In the case of the common flu, the fever is equal or greater than 39°C, while in the case of the A(H1N1) flu the fever is between 37.5°C and 38.5°C. The diagnostic algorithm considers also the case of the common flu, as the symptoms are quite similar as we have already mentioned and there is a need to disseminate by further investigations the false positives from the more threatening AH1N1 infections.

In the cases when fever is at least 39°C the doctor must ask the patient whether he/she has any pain, in which case it might indicate a respiratory virosis should other symptoms occur (such as chills, altered state, watery nose secretions, sore throat, etc.). When there is additional muscle or eye pain we have a case of seasonal flu.

More importantly, the case when the temperature is in the interval mentioned before, is indicating the need for a further investigation; should any of the symptoms cough, discomfort, watery nose secretions, or throat discomfort a HNN analysis is in order (AH1N1 infection is highly possible).

Returning to the main page of the application, there are a few functionalities that need attention: searching and chart generating. In order to perform the search, the instructions in the header of the page must be read; before in the paper we presented samples of the charts generated by the Zed Graphical library (.dll) integrated with the application (figures 2 and 3). The chart engine needs to have as input the period on which it has to draw the charts, and the output is the monthly view of the situation and also the patient number statistics, in case the general management needs such reporting. The Zed Graphics library was chosen for its .Net platform integration oriented design.

4.Conclusion
As we have seen from the paper, the vast .NET platform (containing quite a lot of libraries from a broad range of development areas) can be employed even in the medical diagnostics process, in order to aid the doctors decide upon the most accurate possible verdict and decisions; the application can be considered a decision support system, as it makes the data collected about the patients available at all times for the doctor to review the decisions and eventually make corrections.

The choice of platform was optimal due to the reasons mentioned earlier in the paper and provides the best approach to the problem.

Acknowledgement
This work was supported by CNCSIS –UEFISCSU, project number PNII – IDEI 786/200).

References: