ISBN: 978-989-8704-11-5 © 2014

THE HAEMATOLOGY TREATMENT AID PROVIDING PERONALIZED INFORMATION TO PATIENTS AND CARERS

David Paul[1, 2], Frans Henskens[1, 2], Jamie Bryant[2], Rochelle Smits[2], Robert Sanson-Fisher[2] and William Stevenson[3]

[1]*Distributed Computing Research Group, University of Newcastle, Callaghan, NSW 2308, Australia*
[2]*Health Behaviour Research Group, University of Newcastle, Callaghan, NSW 2308, Australia*
[3]*Department of Haematology, Royal North Shore Hospital, St. Leonards, NSW 2065, Australia*

ABSTRACT

This paper describes a Web-based tool to provide information and support to patients and carers of people diagnosed with haematological cancers. The Web site provides customized information relevant to the type of cancer the individual has, the treatment options available to the patient at the current stage of their illness, and the support services located near the patient. The application has been designed to allow new information to be entered into the system, including new levels of customization, with minimal extra development. This allows required changes to be implemented quickly and easily without compromising the quality of the system.

KEYWORDS

Customization, Web, Cancer.

1. INTRODUCTION

It is always difficult for a patient when faced with a diagnosis of a serious disease such as cancer (Baile et al., 2000). The shock means that a lot of information given to the patient immediately after diagnosis is forgotten (Kessels, 2003) and, as the patient comes to terms with the diagnosis, he or she may wish to know more details about particular aspects of his or her disease and treatment.

To help patients find the information they desire, a Web-based tool, called the Haematology Treatment Aid, has been created to allow patients, their family and friends, and their medical staff to view information relevant to the particular patient's condition, the treatment options available to the patient, and the hospital where the treatment occurs. This includes videos explaining various aspects of the supported diseases (initially four kinds of haematological cancer), and more detailed information, incorporating text and images, that are only displayed if relevant for each patient. Each user is free to explore in as much detail as they desire and can revisit the site as often as they wish.

The system is also being used by researchers who are examining user interaction with the system to determine which information is most important to different types of users. This will aid in communication with future patients by ensuring the most relevant information is readily provided in an accessible way.

This paper introduces the Haematology Treatment Aid. Section 2 describes the functionality of the system. Section 3 details the design used to implement the required functionality, and Section 4 describes how that design allows the system to be enhanced with new information and cancer types.

2. FUNCTIONALITY

The Haematology Treatment Aid is being evaluated using a randomized control trial. Thus, when a physician first diagnoses a patient, a randomization schedule is used to determine whether that patient will be included in the intervention group. If the patient is to be included, the physician logs in to an administrative interface and registers the new patient details. Section 2.1 describes the functionality of this administration interface.

Once registered in the system, a patient is given a unique URL, which can be shared with family and friends, to allow them to log onto the system. The Haematology Treatment Aid then provides information tailored to the particular patient. This functionality is detailed in Section 2.2.

As part of the study, the Haematology Treatment Aid is also being used to examine which information patients find most useful. The collection of the data required for such analysis is provided by the functionality described in Section 2.3.

2.1 Administrator Functionality

When first logging in to the system, an administrative user is presented with a list of their existing patients, including options to edit an individual patient's information, or to disable the patient's account. There is also an option to search their patient list, or to register a new patient.

The interface to edit a patient profile allows the physician to enter the patient's first and last name, date of birth, gender, treating hospital, diagnosis, and available treatment options. This information can be modified as circumstances change (for example, a treatment option may be removed if it is no longer relevant).

When information is entered for a new user, a unique identifier based on the patient's name and date of birth is generated. This identifier is used to create a unique username which can be used by the patient to access information tailored to their personal details. The username can also be added as a URL parameter, so a patient can simply be given a link without having to remember their username.

2.2 Patient Functionality

In order to access the information content, a user must enter a valid username, and his or her relationship with the patient ("I am the patient", "Nominated significant other", "Family member", "Friend", "GP", "Haematologist", or "Other"). For most of the relationships, a date of birth must also be entered. This is required for user tracking, as described in Section 2.3.

Once a user has successfully logged in, they are presented with a video explaining the purpose and structure of the system. The user is then free to explore the system, using navigational links customized based on the information entered by the administrator for the patient associated with the current user name.

The information in the Haematology Treatment Aid is presented through a series of tabs that split the information into sections such as "My cancer and its causes", or "What support is available?". When one of these tabs is selected, the user is automatically presented with the first information item (which may be in video, text, or hypermedia format) from that section, and a navigation menu on the left-hand side of the screen allows the user to view further information if desired.

The user may choose to enter keywords into the search box available on every page to find pages of interest. Further, there is an information button at the bottom of each page, which allows the user to request more information either by calling an available helpline, or by emailing a cancer nurse. The system also includes a simple discussion forum that allows users to interact with each other.

2.3 Researcher Functionality

One of the aims of the Haematology Treatment Aid is to discover the information most useful to patients and their carers. The system collects information for this purpose, which researchers can download from the administrator interface, in two different ways: user interaction (via requests for feedback), and user tracking.

After viewing each piece of information content, there is a chance a user may be presented with a feedback dialog. This dialog asks the user to rank the information content as being "very helpful", "a little helpful", or "not helpful", and enter any desired comments. To ensure a user is not overburdened, the number of feedback requests per session is limited to four. Further, to ensure as many information items as possible have feedback, a limit is placed on the number of times feedback is requested for each item.

The system also tracks user interactions for each session within the system by storing details including which information items are viewed, and how long the user spends viewing each item. This allows the analysis of user actions to determine the items that are of most interest to particular types of users.

3. SYSTEM DESIGN

This Section provides an overview of how the functionality from Section 2 was implemented for the Haematology Treatment Aid. Section 3.1 details the data model for the system, and Section 3.2 describes the software design to use that data.

3.1 Data Model

Initially, the information content for the project was provided as a set of videos, a single Microsoft Word document containing text and images for each information page, and a description of which pages should be restricted to particular diagnoses, treatment options, or patient hospitals. To allow this information to be displayed in as wide a range of Web browsers as possible, it was necessary to convert it into a more Web-friendly format.

To ensure videos could be viewed by as many users of the system as possible, each video was converted to both h.264 and WebM formats. Most browsers only support one of these formats, so both are required. MediaElement.js (Dyer, 2013) was used to ensure a consistent experience regardless of the browser used.

Two options were considered for converting the Microsoft Word document to a set of HTML files. The first option was to use automatic conversion. The other option was to manually convert the content to HTML.

While it initially seemed that it would be faster to use automatic conversion to generate the hundreds of pages of content to HTML, it was ultimately decided that manual conversion was preferable. One reason for this choice was that the HTML generated by automatic tools was not compatible with the HTML5 standards. Further, while each of the pages had a very similar style, the automatic tools regenerated style information for every page. By performing the conversion manually, HTML5 compatibility was ensured and style information was written only once, and then imported into each content page.

Since the information content is predominately static data that will rarely change, it was decided this information should be stored in the file system external to any database. However, information about which pages should be shown for particular patient conditions would be stored in a database with references to the content on the file system.

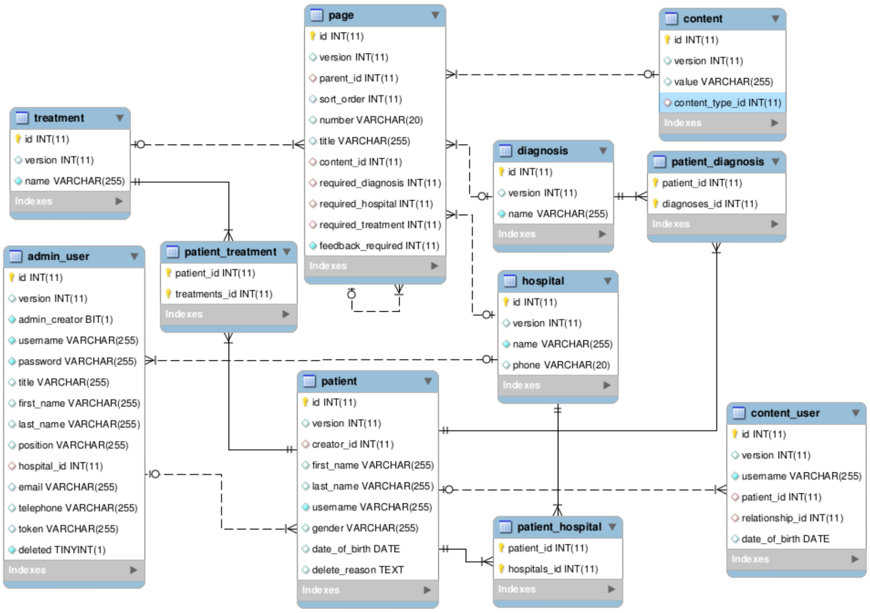

Figure 1. Extract of Entity-Relationship diagram for the Haematology Treatment Aid database.

Figure 1 is an extract of the Entity-Relationship diagram of the Haematology Treatment Aid database. Details of the supported diagnoses, treatments, and hospitals are stored in the *diagnosis*, *treatment*, and *hospital* tables, respectively. The *content* table links the database to the information stored on the file system, and the *page* table contains information specifying which conditions are required for a piece of content to be displayed to a particular user.

The *page* table also includes information specifying the order in which pages should be displayed. This ordering information is specified by creating a hierarchical structure, where each page can have another page as its parent (a page with a *NULL* parent is one of the top-level tabs). Each page also has a *sort_order* to provide an ordering for pages with the same *parent_id*.

Administrative user details are stored in the *admin_user* table, which links to the *hospital* table so physicians can only see patients from their own hospital when logged in to the administrator interface. Each patient has their details in the *patient* table, and there are tables to link diagnoses, treatments, and hospitals to each patient. The *content_user* table is also linked to the *patient* table; this table contains information entered by a general user when logging into the system (i.e. patient identifier, relationship with the patient, and date of birth), and is used for site customization and user tracking.

3.2 System Development

The Haematology Treatment Aid is implemented using Spring MVC (Johnson et al., 2013) with a MySQL (Oracle, 2014) backend. It was developed using HTML5 Web standards (Berjon et al., 2012). The design of the system follows a typical Model-View-Controller (MVC) pattern (Krasner and Pope, 1988).

The MVC pattern is particularly useful for customizing the information displayed to each user based on patient details: the Controller uses current patient information to filter the list of pages returned from the Model to ensure only relevant pages are passed to be displayed by the View. Thus customization logic is only needed in the Controller, and the Model and View need not even know the customization has occurred.

Further, using the Aspect-Oriented Programming (Kiczales et al., 1997) cut points included in Spring Security, it is possible to add the filters required for customization without modifying existing Controller code. Instead, an *AfterInvocationProvider* can be configured so any request for certain Model objects from the Controller has its result object automatically passed through the *AfterInvocationProvider* before being returned to the Controller. This is used, for example, in the Haematology Treatment Aid, by including a *FilterPageByPatientAfterInvocationProvider* that removes any page that does not match the logged-in patient's diagnosis, treating hospital, or treatment options.

4. ENHANCING THE SYSTEM

The Haematology Treatment Aid was implemented using the design in Section 3, and has been in use for over twelve months. Recently, some changes to the study required modifications of the system. These changes required the addition of a new cancer type, including new information to present if a patient is diagnosed with that condition, and the modification of some of the existing information items.

The modification of existing information items simply required replacing the old files with the new ones. To support the new cancer type, it was necessary to add a new record in the *diagnosis* table of the Haematology Treatment Aid's database. Administrative users could then start setting patient's diagnoses to the new condition. The next step was to include the content relevant to that cancer by entering it into the *content* table, and adding items to the *page* table to specify which content should only be displayed for the new diagnosis. This allowed users diagnosed with the new cancer type to log in and see the new information.

Had the system been developed with a less flexible design, these changes would have been much more difficult to implement. For example, if the content had been defined linearly, rather than with the hierarchical structure used in this system, adding the information for the new cancer type would have required all of the content to be reordered. Instead, all that was necessary was creating a new child page to the "My cancer and its causes" page, and ensuring the sort order for this new page was correct. All further pages related to the new cancer could use the new child page as their ancestor, and it was guaranteed they would not affect the ordering of any other pages.

Further, the use of *AfterInvocationProvider*s to filter data meant no code changes were required for the system to support the new cancer type. The provider that checks if the current patient has a diagnosis matching any page under consideration could simply load the new diagnosis id from the database, and the existing check for equality performed just as well with either the existing or the new diagnosis ids.

In fact, if so required, entirely new levels of customization could be added with minimal modification to existing code. For example, if it were decided in the future that different patients should be assigned to different cancer nurses, a table containing nurse information could be added to the database, and a link from the current *page* table to the new *nurse* table could be used to determine the required nurse for particular pages to be displayed. A *patient_nurse* table could then be added and administrative users could set the nurse just as they currently set patients' other options. Then minor changes could be made to the existing *FilterPageByPatientAfterInvocationProvider* to filter pages based on the nurse assigned to the current patient, and the Controller would only pass pages to the View if they had the correct associated nurse.

5. CONCLUSION

The Haematology Treatment Aid is a project that displays customized information to users based on patient details such as the diagnosed cancer type, and the treatment options relevant for the particular patient. Each user is able to explore the system in as much detail as they would like, at their own schedule.

Information is being collected from user interaction with the system to allow researchers to study how the information in the project is being accessed and whether the information is considered useful. This will aid in the future by determining the most important and relevant information which can then be emphasized when communicating with new patients or their carers.

The system has been designed to allow the entry of new information by adding records to the existing database that specify under which conditions the new information should be available. Further, entirely new levels of customization can be added with minimal change to the existing code base. Future use of the system by patients and researchers may yield more required changes, and it is anticipated that these will be implemented with a minimum of additional Web development.

ACKNOWLEDGEMENT

This research is supported by a New South Wales Cancer Institute Translational Health Services Research Grant. Dr. Jamie Bryant is supported by an Australian Research Council Post-Doctoral Industry Fellowship.

REFERENCES

Baile, W. F., Buckman, R., Lenzi, R., Glober, G., Beale, E. A. & Kudelka, A. P. 2000. SPIKES—a six-step protocol for delivering bad news: application to the patient with cancer. *The Oncologist,* 5**,** 302-311.

Berjon, R., Leithead, T., Navara, E. D., O'connor, E., Pfeiffer, S. & Hickson, I. 2012. *HTML5: A vocabulary and associated APIs for HTML and XHTML* [Online]. Available: http://www.w3.org/TR/2012/CR-html5-20121217/ 2014-01-10].

Dyer, J. 2013. *MediaElement.js - HTML5 video player and audio player with Flash and Silverlight shims* [Online]. Available: http://mediaelementjs.com/ 2014-01-10].

Johnson, R., Hoeller, J., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., Arendsen, A., Davison, D., Kopylenko, D., Pollack, M., Templier, T., Vervaet, E., Tung, P., Hale, B., Colyer, A., Lewis, J., Leau, C., Fisher, M., Brannen, S., Laddad, R., Poutsma, A., Beams, C., Abedrabbo, T., Clement, A., Syer, D., Gierke, O., Stoyanchev, R. & Webb, P. 2013. *Spring Framework Reference* [Online]. http://docs.spring.io/spring/docs/3.2.5.RELEASE/spring-framework-reference/pdf/spring-framework-reference.pdf. 2014-01-10].

Kessels, R. P. 2003. Patients' memory for medical information. *JRSM,* 96**,** 219-222.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M. & Irwin, J. 1997. Aspect-Oriented Programming. *Proceedings of the European Conference on Object-Oriented Programming*.

Krasner, G. E. & Pope, S. T. 1988. A cookbook for using the Model-View Controller user interface paradigm in Smalltalk-80. *J. Object Oriented Program.,* 1**,** 26-49.

Oracle. 2014. *MySQL 5.7 Reference Manual* [Online]. Available: http://dev.mysql.com/doc/refman/5.7/en/ 2014-01-10].