Acta Scientiarum
http://periodicos.uem.br/ojs/acta
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v41i1.39483

A web-based application for knowledge sharing in the emergency unit of a university hospital

João Luiz Ramalheira de Almeida[1*], Max Naegeler Roecker[1], Vinicius Paes de Camargo[1], Heloise Manica Paris Teixeira[1] and Renato Balancieri[1,2]

[1]Departamento de Informática, Universidade Estadual de Maringá, Avenida Colombo, 5790, 87020-900, Maringá, Paraná, Brazil. [2]Universidade Estadual do Paraná, Apucarana, Paraná, Brazil *Author for correspondence. E-mail: jlramalheira@gmail.com

ABSTRACT. This paper presents the development of the Alvar Architecture, which aims to manage learning materials (images, videos, and others medias) shared by professors and the health care team from the emergency unit of a University Hospital. The main objective was to model a web-based architecture for knowledge sharing, which enables precise documents search in weak-connectivity conditions from mobile computing devices. To demonstrate and validate the architecture, a prototype was implemented, named Alvar System, using free and open source software. Early results suggest that the implemented prototype can be successfully applied in the emergency unit to promote knowledge sharing by expert professors and their students.

Keywords: health informatics; web-based systems; mobile devices.

Received on September 14, 2017.
Accepted on December 7, 2018

Introduction

Medical staff that work in emergency units needs to have access to comprehensive information about medicines, diagnoses and treatments as fast as possible. A wide range of knowledge-based information is available on the Web, including on-line databases, digital libraries, medical textbooks, best practices and clinical practice guidelines. Health care is an information-based science. A clinical practice involves gathering, synthesizing and acting on information. Nevertheless, in the emergency department of a hospital, the waste of time to search for information before making a decision is very unwelcome.

This scenario creates challenges for knowledge access in the medical urgency and emergency unit. However, the recent technological advances in mobile computing make it feasible to consider the development of systems to be employed at the point of care to provide knowledge regarding patient healthcare. With the growth of on-line information resources, applications to use them through a mobile device (tablet, smart-phones or personal digital assistants - PDAs) have been developed. In a typical scenario, emergency staffs, using a portable device, could review a clinical procedure for a complex situation. Mobile devices offer a wide range of features in a compact way, allowing mobility and making it possible to access texts, videos, tools and applications to aid in everyday tasks of medical staff in hospitals.

The development and introduction of new applications in the emergency care context requires an accurate system modeling process. It is highly recommended that medical schools collaborate with each other in order to share learning resources and materials (Greenhalgh, 2001; Davis et al., 2001). The peculiar characteristics of the emergency unit, leads to the development of specific medical documents and search mechanisms.

In this paper, the Alvar System is presented, a prototype modeled and implemented based on a public University Hospital of Maringá (UHM). The motivation for this research is that UHM has over three hundred students engaged in a supervised clinical experience in hospital and community settings. Furthermore, public university hospital faces the issue of limited technological resources in the practice setting that supports clinical learning.

The main objective of this research was to model and develop a custom web-based architecture for knowledge sharing, which enables precise documents search (reports, clinical guidelines and clinical procedures) for the emergency unit staff. The system should interact with existing hospital information systems, run on mobile computing devices and be suitable for operation in weak-connectivity conditions and unstable connections. To reach this objective, the proposed architecture was developed based on a study of the emergency unit of UHM to be integrated with an electronic whiteboard system.

This paper is organized as follows: The next sections present related work concerning the use of mobile device technology for educational purposes, some existing web-based systems to search medical information and usability requirements for mobile applications. In the sequence, the methods and the proposed web-based architecture and its main components and techniques are described. To demonstrate the proposed architecture and its validation, the prototype and a user evaluation are presented. Finally, the main conclusions and some directions for future work are described.

Related work

Mobile devices such as cell phones and PDAs are becoming widely used in hospitals not only for medical assistance but also for educational purposes. In consequence, there is an extensive range of research based on mobile devices to support clinical decision making, electronic documentation, and patient tracking and enhance learning among physicians and medical trainees (Lindquist, Johansson, Petersson, Saveman, & Nilsson, 2008). In addition, many studies have looked at why medical and nursing schools include PDAs in their education and training program (Tempelhof, 2009; Putzer & Park, 2010).

A mobile technology use in medical education study was presented in Luanrattana, Win, Fulcher, and Iverson (2012). Boruff and Storie (2014) investigated the extent to which students, residents, and faculty members in medical faculties use mobile devices to find medical information. The results of this study shows that barriers to access and lack of awareness might keep the students from using reliable, library-licensed resources.

Callegari, Jersak, and Costa (2013) present the state of the art from the available literature on mobile health care. The study was performed by means of a systematic review and the analyses (since 2010) show development in ten application areas, ongoing trends and technical challenges on the subject. The application areas include patient monitoring, infrastructure, software architecture, modeling, framework, security, notifications, multimedia, mobile cloud computing, and literature reviews on the topic. The most relevant challenges include the low battery life of devices, multiplatform development, data transmission and security. This study has consolidated important findings in the field and serves as a resource guide for future research planning and development.

Bellini, Nesi, Simoncini, and Tibo (2014) introduce a general architecture of a Mobile Emergency Solution. The mobile application is available on the iPhone for personal support. The main functionalities are: possibilities of communicating events (maintenance and/or emergency), monitoring events/emergency getting information from the server, collaborating with other colleagues via mobile communication, navigating in/out getting information from the Mobile Medicine Best Practice Network.

Despite the area of use, usability is an important concept when developing software. Since Nielsen started researching heuristic usability in 90's (Nielsen 1994; Nielsen, 1999) until now, many discoveries happened and many theories were purposed. Today, usability is defined as the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use (International Standardization Organization [ISO], 2018).

In the context of mobile application, the heuristic usability is a bit different from the web application. Nayebi, Desharnais, and Abran (2012) stated that is important to consider aspects such as the touch gesture, since the user may have some paralysis and size of buttons, since the screen is generally small. However, that year, the application ecosystem was not developed as it is today, so the paper stated that usability had no well addressed scientific research.

Currently, about 70% of software usability question present in questionnaires are cross platform, focusing on desktop, mobile and web (Assila & Ezzedine, 2016). However, this rate are not enough to cover the most important question when evaluating a mobile application: 'How difficult is to user access the main functionality of the application?'. Joyce, Lilley, Barker, and Jefferies (2016) developed and evaluated a mobile-specific questionnaire that addresses this question and other mobile-specific questions and stated that this kind of questionnaire is more suitable when evaluating mobile applications, getting a better evaluation when compared as traditional desktop-based or web-based questionnaires.

Related works often are proposed for iPhone/iPod/iPad and PDA as Windows Mobile based phones (Luanrattana et al., 2012; Bellini et al., 2014). The solution presented in this paper gathers contributions from the related work available in the literature. Unlike previous researches, it was based on free software components, so that new findings and future development may arise.

Caracterization of search

This research can be characterized by four perspectives: nature, approach to the problem, goals and adopted procedures (Gil, 2002):

- Nature: the research can be considered as applied, since it generates knowledge for practical application directed to the solution of specific problems.
- Approach of the problem: the research is qualitative-quantitative. Since it involves the analysis of data in an inductive way, considered qualitative. In addition, quantitatively, then opinions and information will be translated in quantifiable form.
- Objectives: The research can be considered as exploratory, because it seeks a greater familiarity with the problem, in order to make it explicit and to allow the formulation of hypotheses.
- Procedures adopted: the research is initially supported by a bibliographical research and then by a case study, involving a thorough and exhaustive study of the object, so as to allow it sample and detailed knowledge.

Material and methods

In order to identify the requirements for developing the web-based application architecture, the scenario regarding the emergency conditions have been formalized and depicted by using standard Unified Modeling Language (UML) models. The system requirements were identified and formalized by analyzing the scenarios, reviewing the literature and interviewing emergency experts and professors that works in the hospital.

The business-modeling phase defined the activities that Alvar System should perform and the actors responsible for these activities. The Astah Community modeling tool supported this step. Then, the analysis phase aimed to understand the system requirements and to detail the software requirements, that is, refine the system requirements for software requirements. The Object Oriented Approach was used to represent this understanding. Afterwards, in the design phase, it was defined the software architecture. The chosen architecture was the model layers. In this activity, some design decisions were taken to attend nonfunctional requirements, such as performance, reliability and maintainability. The standard Boolean Model was used in the information retrieval (Lancaster & Fayen, 1974), which is based on Boolean logic and Set Theory in that resources and users term are conceived of sets of terms. The terms of each resource are consisted of the title, description and a list of tags. A search query is a Boolean expression consisted of unique terms and produces a union of results for each term. The search query results are ordered by the rate of the resource. The users can review the resource and rate it from zero to five stars, the best rated resources will come on top of the search results.

The next phase was the effective coding for the development of a prototype. The prototype was designed using MySQL, Node.js, and Web technologies to create a dynamic and platform-independent application. The MySQL as used for storage and retrieve data; the Node.js was used to create a server that interfaces for resource management; and the client application was built with Web technologies. Other technologies and its interaction are presented in the related technologies section.

The list of tags of a resource used in search terms are provided by DeCS (Portuguese for 'Health Sciences Descriptors'), a trilingual and structured vocabulary that is used for medical document indexing. DeCS (Centro Latino-Americano e do Caribe de Informação em Ciências da Saúde [Bireme], 1992) was developed from the MeSH (Medical Subject Headings of the U.S. National Library of Medicine) with the purpose of permitting the use of a common terminology for searching in the three languages (Portuguese, Spanish and English).

To be successful in clinical practice and fit into the clinical workflow, the modeling process it was iterative. Each system functionality was discussed and validated by users (healthcare professionals and students that work in the emergency unit). To demonstrate our contribution, it was designed and implemented a prototype that executes on mobile devices. Tests and validation have been performed in a controlled experiment with academic and professionals from health area.

Overall system architecture

This section addresses the proposed architecture based on a client server solution. Alvar Architecture fits the Zare Platform, which is a software platform comprised of several healthcare systems and applications developed for the University Hospital setting. This paper focuses on describing the Alvar Architecture, which aims to manage and retrieve resources previously uploaded by professors or preceptors of the emergency unit. Resource is a generic term to define structured data of the media files: photos, video, audio or documents. Figure 1 illustrates an architectural view of Zare Platform, including Alvar Architecture.

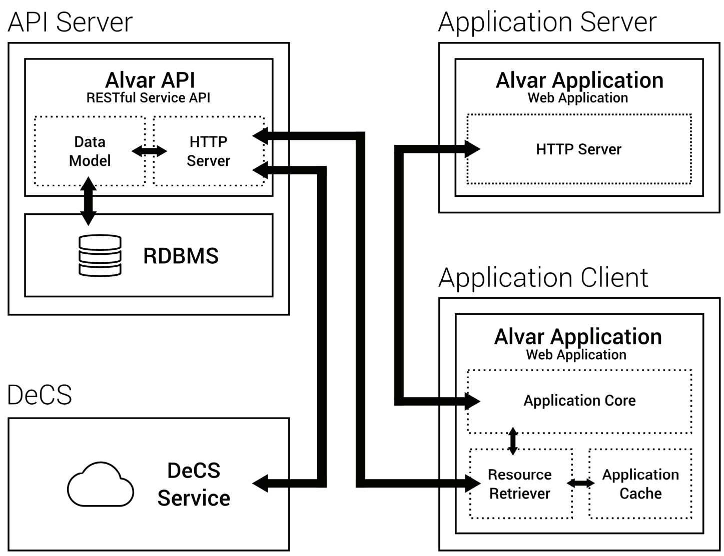

Figure 1. Architectural view of Zare Platform.

The HTTP Server is the main interface for the requests to the API Server. It is responsible for redirecting a request to the correspondent Resource Model. The Resource Model contains the business and access control rules of the API Server resources. It is also responsible for data persistence or retrieving from the Storage System, where the resource data is persisted. It can be a relational database management system or the local file system.

Client Application resources and assets are exposed in the Application Server, which is composed by two main parts: the HTTP Server and the Static files. The HTTP Server is the main interface for Application Server requests. It is responsible for redirecting the request to the correspondent Static file. The Static files are a set of files that contain the Client Application resources and assets.

The Application Client is where the application is actually run and interacts with the user. When requested, it is responsible for retrieving resources from the API Server and exposes it to the user. The Application Client can also post new resources to API Server to be later retrieved. It is also in charge of requiring other assets from Application Server in order to maintain itself in consistent run-time status. In summary, the Application Client is allowed to perform resource registration and updates, and to perform tag-based queries to retrieve resources from the API Server.

The Application Client can be executed from different electronic devices, such as cell phones, tablets or desktop computers. When using a Web Browser, the user starts a connection via URI (Uniform Resource Identifier) to the Application Server. When the Application Client files are downloaded, the application starts running.

Three main parts compose the Application Client: the Application Core, the Resource Proxy and the Application Cache. The Application Core is majorly responsible for maintaining the application running in the Client Application and responds to user interactions. The Resource Proxy module is responsible for intercepting the user's requested resources. The Application Cache is a cache system where resources are stored for a period of time. Before perform a request to API Server, the Resource Proxy hits (ask the required information for) the Application Cache. If no results are found in the cache, the Proxy requests the resource to the API Server via HTTP, and saves the response (new data/information) in the cache.

Mobile devices such as cell phones and PDA are well known for offering limited computing capabilities. They have battery limitations and reduced computational performance compared to desktop computers. Moreover, frequent disconnections and low bit rate occurs in HUM's wireless network, what makes it difficult to provide information for mobile users. In addition, as Michalowski et al. (2005) says, the wireless availability in a hospital cannot be safely considered because it can interfere with medical equipment. In this context, data caching plays a key role in data management to improve system performance and availability in case of disconnections. Further, it can help save battery power in a client-server communication model and mobile users are still able to work using the cached data when the network is unreachable.

The mobile and network challenges described above are addressed in this research considering a cache model usage. The Application Client has a component responsible for cache management. Functionalities on the Client Application include a local data storage, which contains information that was collected by previous queries executed by the users. The first action is to verify whether a full (or partial) answer for a query exists inside the local cache. Only if necessary the system forwards the query to the server.

The API Server communicates with DeCS Web Service, that is an ontology domain for the healthcare area. The structured vocabulary is used to link keywords with documents (reports, clinical guidelines and clinical procedures, for example) by using the appropriate emergency vocabulary. The concepts that characterize the DeCS vocabulary are organized in a tree structure allowing a search on broader or narrower terms or on all terms from the same tree within the hierarchical structure (Bireme, 1992). In the Alvar System, when uploading a new file, professors should know about related descriptors or import them from DeCS.

Proposed prototype and evaluation

This section presents a prototype designed and implemented to demonstrate and validate the Alvar Architecture in practice. The following sections describe the technologies used to the prototype implementation and illustrate a scenario conceived to demonstrate the knowledge sharing among expert professors and their mentees. Finally, a user experience evaluation and preliminary results are described.

Related technologies

For the API Server implementation a full JavaScript stack based on Node.js was used. The HTTP Server was built in Express.js[1], a minimalistic open-source web framework that supports standard HTTP requests and responses. The Resource Model was defined with LoopBack[2], an open-source framework to development of RESTful APIs.

Resources with structured data are stored in the relational database system MySQL[3]. Otherwise, media file resources are stored in the Server's local file system. The API Server only supports requests with JSON (JavaScript Object Notation) media type. On responses, resources are also serialized to a JSON representation.

The Application Client uses three main standards with high interoperability between platforms and devices: HTML (Hypertext Markup Language), CSS (Cascade Style Sheets) and JavaScript. The HTML is used to structure and organize the information on a document page. The CSS is responsible for presenting and stylizing the information organized by the HTML.

To meet the evolution of web applications, the Application Client follows the specifications and APIs of HTML5, highlighting, the Web components. Web components are a new feature in web platform, consisting of four fundamental specifications: Shadow DOM (Document Object Model), Custom Elements, HTML Imports and Template element. The Shadow DOM specification describes a method of combining multiple DOM trees into one hierarchy and how these trees interact with each other within a document, thus enabling better DOM composition. The Custom Elements specification describes the method to define and uses new types of DOM elements in a document. The HTML Imports specification is a way to include and reuse HTML documents in other HTML documents. The Template element is used to declare HTML fragments that can be cloned and inserted in the document by script.

The Alvar Application Client uses the Polymer library[4] to create Web components and manipulate events and interactions with the interface. The Application Cache module implementation was built using IndexedDB, a simple server-less key-value database, supported in most modern Web browsers. In this research, the cache management is implemented according to LRU policy.

[1] http://expressjs.com/
[2] https://loopback.io/
[3] https://www.mysql.com/
[4] https://www.polymer-project.org/

System functionalities

The Alvar System prototype is a free and open source software and can be accessed from any electronic device with a web browser. To demonstrate and validate the proposed solution in practice, the implemented prototype was performed on a mobile device. The experimental environment consisted of a server connected to the client with a wireless network. The experiments were based on simulation trials on a Motorola Moto G phone.

The user interface was designed to be easy to use and show the needed information in a meaningful way. First, the usability was tested and second, a real trial was performed in order to verify its operation in a real device in the hospital network. The client application interface is organized in two contexts that are called Retrieve Context and Management Context.

The Retrieve Context provides the interface (based on search-key queries) for query information about the resources. It also allows the user to download selected files and to store it on the local cache. This context has public access and no authentication is required. It can be used by any person that works in the hospital, allowed to search and to grade resources. To perform a search for resources, the user has to input one or more search key. If the query returns any result, the documents found will be exhibited, sorted by user rate.

To exemplify the Retrieve Context interface, let us consider a student user, interest about early management of patients with cerebral stroke. Then, he inserts any descriptor or tag in the search bar, for instance, ‘Stroke’, as seen in Figure 2. If a professor previously tagged a document about this subject using DeCS descriptors, all documents about cerebral stroke and its synonym (Apoplexy, Cerebrovascular Accident, Cerebrovascular Apoplexy, Vascular Accident or Brain) will be returned as a result. Lastly, the user is requested to rate all accessed document.

The Management Context extends the Retrieve Context features and functionalities, providing an interface to Alvar’s API Server management. To access the Management Context, the user must be authenticated and authorized by the Alvar System Administration.

The Management Context functionalities include resources upload and statistical reports about previously executed queries. To upload a new resource, the user must tag the new document with related tags. The new term can be typed by the user or imported from DeCS. If DeCS descriptors are used, its definition and synonyms are automatically imported and the user is allowed to complement or to modify this information.

To exemplify the Management Context interface, let us suppose that an authorized professor wants to upload a new document about treatment guidelines for solar burns. Firstly, the document must be registered and linked with related tags, as illustrated in Figure 3. In this example, we suppose that the user can’t find tags related with this subject in the system. In this case, he can include new tags or import from DeCS the formal descriptors: ‘queimadura solar’ (in Portuguese) or ‘sunburns’ (in English) or ‘quemadura solar’ (in Spanish).

Management Context functionality is the statistic report about previously executed queries. This report is used to verify the frequency that a document is accessed and also to identify terms that are being used as search-key with no results. Terms with no results represent areas that students are looking for, and this suggests that they are in doubt or needing more information and learning material.

Figure 4 illustrates a list of terms searched without results. These listed topics are those that students are looking for, and must to be considered by professors or preceptors.

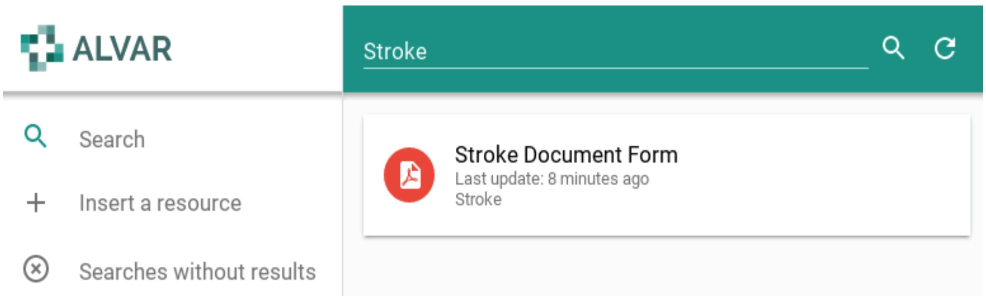

Figure 2. A tag-based query result.

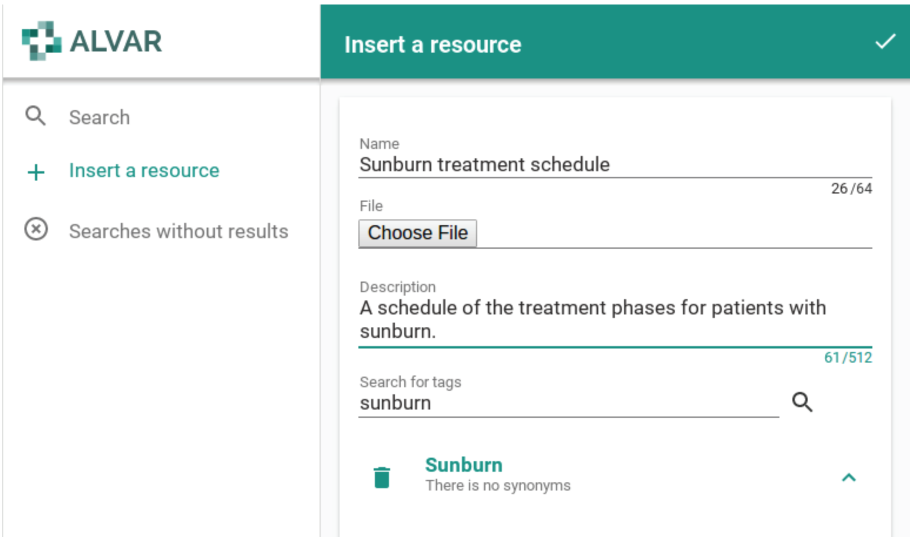

Figure 3. Searched keywords without related documents.

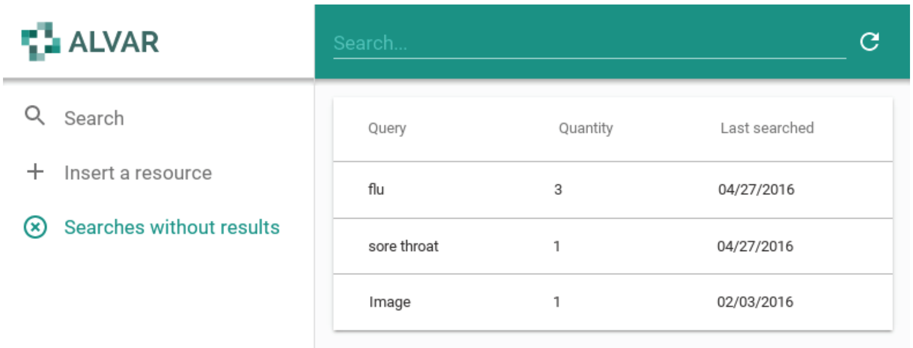

Figure 4. Searched terms without results.

User evaluation

The main objective of the user evaluation was to gather his opinions about the experience of using the prototype and to identify interaction problems and opportunities for improvement. The evaluation method used was the observation in a controlled environment. The evaluation was performed in a laboratory at the university, so that external interference does not hamper the use of the system. The laboratory was equipped with twenty-two microcomputers with Internet connection.

All questionnaires were based on Gómez, Caballero, and Sevillano (2014) which describes 13 heuristics and 230 questions to be considered when evaluating a mobile interface. In the set of 230 question, many of them are not applicable for Alvar application, so the questionnaries of Alvar have a smaller set of question, focusing on answer that will help improve Alvar in specific context of its use.

Before starting the evaluation, each participant answered a pre-test questionnaire (four questions), used to collect information about his or her profile. The application was presented to 11 health professional volunteers in the nursing area (two undergraduate, four master and five doctorate). After, the system was presented to the participating group who given the instructions about the main system functionalities and the tasks to be performed. Each participant was allowed to explore the system freely for a while, simulating the scenario of a group of students and teachers who were sharing material about different topics.

A structured form was used to support the observations and data capture. In addition, the researches took notes about relevant events, such as spontaneous speech. For this, the participants were asked to report aloud what they were thinking during the system interaction, as well as their observations - think aloud technique (Ericsson, 2006). In this way, several suggestions and opinions have been recorded from their experiences during the use of the application.

After the user completed the interaction, the post-test interview was conducted, collecting their opinion about the experience, their perceptions and interpretations during the use of the application. Figure 5 illustrates the main questions of the post-test questionnaire.

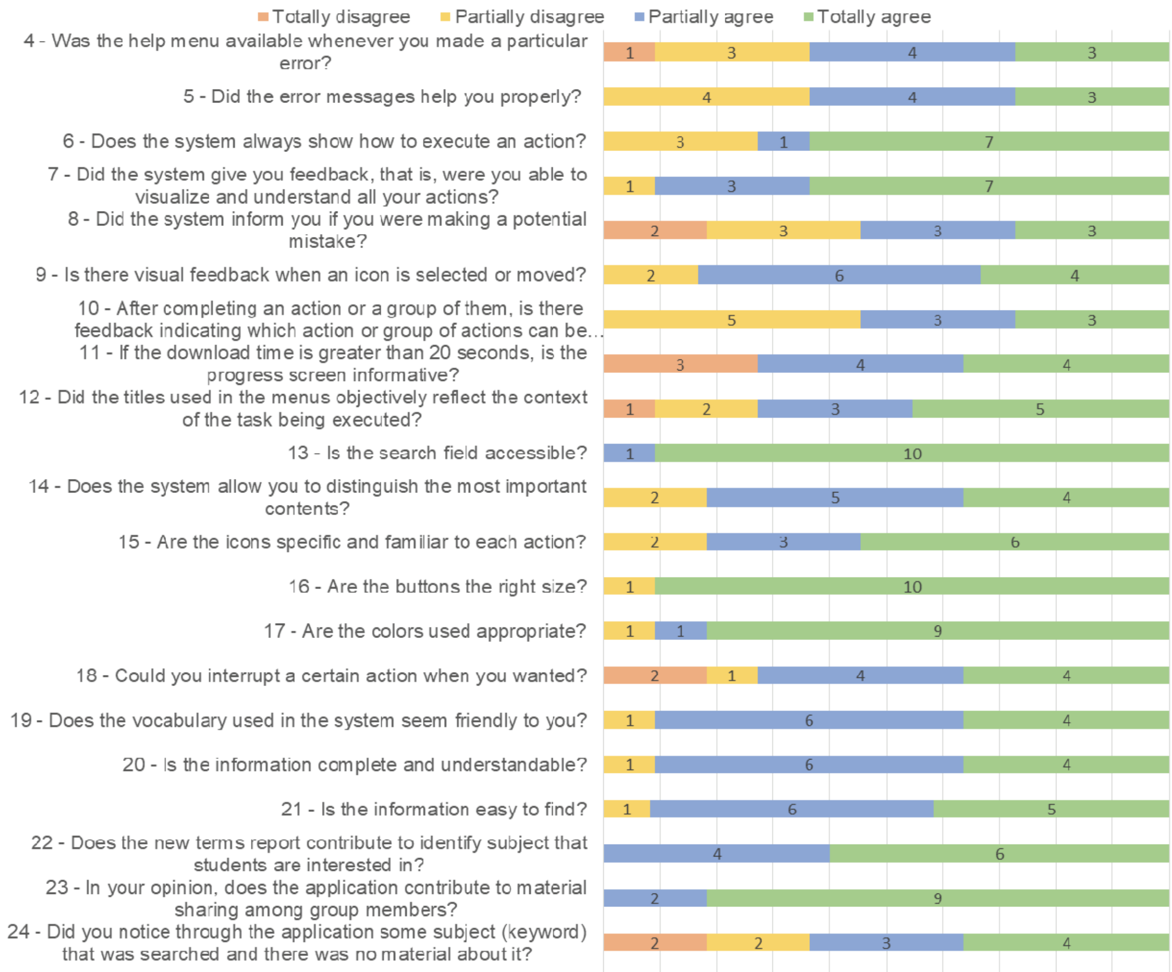

Figure 5. Post-test questionnaire.

Regarding help needed while using the application (questions 4 and 5), seven participants agreed that the system had instructions available to guide the user when some type of error had been committed and considered the error messages are appropriate. Also, seven users felt that the system was intuitive (questions 6 and 7) because it allowed visualizing and understanding an action while using the application.

Questions 8 to 11 are related to the feedback given from the system to the user in some situations, such as: when an error occurs (question 8), an icon is selected or moved (question 9), after completing an action (question 10), download time (question 11). In general, most participants agreed that the system met these requirements; however, it was observed that users felt the need for system improvements in these items.

Regarding the headings and menu names (question 12), eight users consider that the icons and menus objectively reflect the context of the task being executed. When performing a search for some document, eleven users considered the field accessible (question 13) and nine affirm that the system allows distinguishing the most important documents among the search results (question 14).

Questions 15, 16, and 17 refer to the colors, icons and sizes of the buttons used. Most participants found these items appropriate. Regarding the possibility to cancel actions (question 18), eight were able to interrupt an action when they wished.

In general, users found the vocabulary used was friendly (question 19) and the information accessed was complete, understandable (question 20) and easy to find (question 21).

Questions 22 to 24 are about the contributions and benefits of the application for academic and learning purpose. It was observed that most participants considered that the application could contribute to the sharing of learning materials among group members. Furthermore, they were able to identify subjects (key word terms) that were being extensively searched and were not returning documents in the search result.

Conclusion

The Emergency Unit of the University Hospital is the front door to the health care system and operates under increasing pressure, such as mounting patient volumes and staff shortage. In the Emergency Unit, clinicians, nurses, medical students and residents frequently receive patients that require immediate action. For effective integration into physician workflow, information must be widely available, both at the bedside and at other locations throughout the hospital.

The adoption of technology in medical education and medical practice, particularly in busy units, can help achieve the improvements in healthcare delivery with less error in clinical practice. Students have immediate access to clinical resources and information on-the-spot.

The main contribution of the proposed architecture for emergency unit is to deliver essential clinical guidelines to healthcare staff whenever are needed. The shared documents are used to guide medical students and residents to cope with emergency management and best practice determined by the institution. Furthermore, physicians and professors can be aware of the student's doubts and are able to upload documents and update relevant information with regard to their learning. Students can grade the shared learning materials, which allow the document authors (preceptors, professors or doctors) to update their materials if required.

Thus, this model provides a central resource base. In terms of future developments, the learning materials can be shared with other courses of health area within the University. This also allows health students and professors from different organizations to exchange skills, resources and ideas and collaborate on learning resources and materials.

The experiment performed with users was of great value and allowed to identify problems that can be solved with adjustments in the application, such as to include automatic messages of feedback. The preliminary evaluation was positive and suggests that the developed prototype can contribute to the document sharing not only for academic purposes but also in their professional activities. The next step is to evaluate the prototype in a real-use scenario.

Acknowledgements

The authors would like to thank the health professional volunteers that participate in the user evaluation. We also thank the Araucaria Foundation for the scholarship provided.

References

Assila, A., & Ezzedine, H. (2016). Standardized usability questionnaires: Features and quality focus. *Electronic Journal of Computer Science and Information Technology: eJCIST, 6*(1), 15-31. doi: 10.1016/B978-0-12-384968-7.00008-4

Bellini, P., Nesi, P., Simoncini, M., & Tibo, A. (2014). Maintenance and emergency management with an integrated indoor/outdoor navigation support. *Journal of Visual Languages & Computing, 25*(6), 637-649. doi: 10.1016/j.jvlc.2014.10.009

Boruff, J. T., & Storie, D. (2014). Mobile devices in medicine: A survey of how medical students, residents, and faculty use smartphones and other mobile devices to find information. *Journal of the Medical Library Association, 102*(1), 22-30. doi: 10.3163/1536-5050.102.1.006

Callegari, D. A., Jersak, L. C., & Costa, A. C. (2013). Technical trends and challenges in mobile health - a systematic review of recent available literature. In S. Hammoudi (Ed.), *Proceedings of the 15th International Conference on Enterprise Information Systems* (p. 519-525). SCITEPRESS. doi: 10.5220/0004563405190525

Centro Latino-Americano e do Caribe de Informação em Ciências da Saúde [Bireme]. (1992). *4.3 DeCS: descritores em ciências da saúde: lista permutada* (2a ed.). São Paulo, SP: Bireme/Opas/OMS.

Davis, M. H., Ben-David, M. F., Harden, R. M., Howie, P., Ker, J., McGhee, C., ... Snadden, D. (2001). Portfolio assessment in medical students' final examinations. *Medical Teacher*, 23(4), 357–366. doi: 10.1080/01421590120063349

Ericsson, A. K. (2006). Protocol analysis and expert thought: concurrent verbalizations of thinking during experts' performance on representative tasks. In K. A. Ericsson, N. Charness, P. J. Feltovich, & R. R.

Hoffman (Eds.), *The Cambridge handbook of expertise and expert performance* (p. 223-241). New York, NY: Cambridge University Press.

Gil, A. C. (2002). *Como elaborar projetos de pesquisa* (4a ed.). São Paulo, SP: Atlas.

Gómez, R. Y., Caballero, D. C., & Sevillano, J. L. (2014). Heuristic evaluation on mobile interfaces: A new checklist. *The Scientific World Journal, 2014*, 1-19. doi: 10.1155/2014/434326

Greenhalgh, T. (2001). Computer assisted learning in undergraduate medical education. *British Medical Journal*, 322(7277), 40-44. doi: 10.1136/bmj.322.7277.40

International Standardization Organization [ISO]. (2018). *Ergonomics of human-system interaction — part 11: usability: definitions and concepts* (2nd ed.). Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en

Joyce, G., Lilley, M., Barker, T., & Jefferies, A. (2016). Mobile application usability: heuristic evaluation and evaluation of heuristics. In B. Amaba (Ed.), *Advances in Intelligent Systems and Computing* (p. 77–86). Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-41935-0_8

Lancaster, F. W., & Fayen, E. G. (1974). *Information retrieval on-line*. Los Angeles, CA: Wiley-Becker & Hayes.

Lindquist, A. M., Johansson, P. E., Petersson, G. I., Saveman, B.-I., & Nilsson, G. C. (2008). The use of the Personal Digital Assistant (PDA) among personnel and students in health care: a review. *Journal of Medical Internet Research, 10*(4), e31. doi: 10.2196/jmir.1038

Luanrattana, R., Win, K. T., Fulcher, J., & Iverson, D. (2012). Mobile technology use in medical education. *Journal of Medical Systems*, 36(1), 113–122. doi: 10.1007/s10916-010-9451-x

Michalowski, W., Slowinski, R., Wilk, S., Farion, K. J., Pike, J., & Rubin, S. (2005). Design and development of a mobile system for supporting emergency triage. *Methods of Information in Medicine, 44*(1), 14-24. doi: 10.1055/s-0038-1633918

Nayebi, F., Desharnais, J. M., & Abran, A. (2012). The state of the art of mobile application usability evaluation. In *25th IEEE Canadian Conference on Electrical and Computer Engineering*. Piscataway, NJ: IEEE. doi: 10.1109/CCECE.2012.6334930

Nielsen, J. (1994). *Usability engineering*. Fremont, CA: Elsevier.

Nielsen, J. (1999). *Designing web usability: The practice of simplicity*. San Francisco, CA: New Riders Publishing.

Putzer, G. J., & Park, Y. (2010). The effects of innovation factors on smartphone adoption among nurses in community hospitals. *Perspectives in Health Information Management/AHIMA, 7*(1b), 1-18.

Tempelhof, M. W. (2009). Personal digital assistants: a review of current and potential utilization among medical residents. *Teaching and Learning in Medicine: An International Journal, 21*(2), 100-104. doi: 10.1080/10401330902791321