

CLINICAL CONTENT TRACKING SYSTEM – AN
EFFICIENT REQUEST TRACKING VIA A GRAPHICAL
USER INTERFACE

A Thesis presented to
the Faculty of the Graduate School
at the University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree

Master of Science

by
SAIF KHAIRAT

Dr. Chi-Ren Shyu, Thesis Supervisor

DECEMBER, 2007

The undersigned, appointed by the dean of the Graduate School, have examined this thesis entitled

CLINICAL CONTENT TRACKING SYSTEM- AN EFFICIENT
REQUEST TRACKING VIA A GRAPHICAL USER
INTERFACE

presented by Saif Khairat,

a candidate for the degree of Master of Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Chi-Ren Shyu

Professor Ye Duan

Professor Guilherme N. DeSouza

To my parents who support me since the day I stepped a foot to this world. If I write a book expressing my gratitude, appreciation and admiration to my parents I don't think I will give them what they deserve. I dedicate my little success to them and thank them for helping me become who I am today.

To my sister, Salma Khairat, for her unconditional love.

To all my friends and peers.

ACKNOWLEDGMENTS

Since I was an undergraduate, my advisor Dr. Chi-Ren Shyu has truly been an inspiration to me. His continuous support and utmost care about my career has been a key for my success in various aspects of life. Working as a student under his supervision, I can say I was among the luckiest students who had the honor of working with him and learned a lot from his past experiences which naturally lead to looking up to him as a role model in many areas. To me, Dr. Shyu is not just an advisor in graduate school but he is a great friend outside school and sometimes an older brother when it comes to giving advices. His professionalism in work and humbleness during discussions made my graduate studies a memorable experience in my life and for that I am very thankful to him.

Also, thanks to Dr. Ye Duan and Dr. Guilherme DeSouza for their service as my thesis committee members. I give many thanks to Catherine K. Craven for her tremendous effort and collaboration for almost two years to make CCTS happen. Finally, I would like to send many thanks to my peers at the MedBio lab for all their help and support during stressful times. Also, many thanks to Radwa Aly for her influential input and for proofreading my Thesis in a timely manner.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	I
LIST OF TABLE	IV
LIST OF FIGURES.....	V
ABSTRACT	VI
CHAPTER	1
1. INTRODUCTION.....	1
1.1 MOTIVATIONS.....	1
1.2 NEED FOR CLINICAL CONTENT TRACKING SYSTEM (CCTS).....	4
1.3 INTRODUCTION LITERATURE REVIEW.....	7
1.4 PROPOSED SYSTEM	10
2. SYSTEM ARCHITECTURE	13
2.1 GROUPS/ROLES ASSIGNED TO THE USERS IN THE DATABASE MANAGEMENT SYSTEM (DBMS).....	17
2.2 SYSTEM MODULES	19
2.2.1 Modules Description.....	20
2.3 DATA FLOW	25
3. IMPLEMENTATION - PROGRAMMING	32
3.1 PROGRAMMING MECHANISMS.....	32
3.1.1 Languages used.....	32
3.2 CODE SNIPPETS.....	32
3.2.1 Functionalities	32
3.2.1.1 Pseudo code	32

3.2.1.2 GUI Design.....	45
3.2.1.3 PHP Hypertext Preprocessor (PHP)	47
4. DATABASE MANAGEMENT	48
4.1 ENTITY RELATIONSHIP DIAGRAM.....	48
4.2 DATABASE ARCHITECTURE.....	50
4.2.1 Database Optimization and Tuning	50
4.2.2 Database Tables	52
4.2.2.1 Tables Description	52
4.2.2.2 Data Definition Language (DDL).....	55
4.2.2.3 SQL Statements	58
4.2.3 Security settings.....	63
4.3 OUTLINE OF THE DATABASE MANAGEMENT SYSTEM PROCESS.....	64
5. CONCLUSION, LIMITATIONS AND FUTURE WORK.....	69
5.1 CONCLUSION	69
5.2 SYSTEM LIMITATIONS.....	70
5.3 FUTURE WORK.....	71
REFERENCE	73
VITA	88

List of Table

TABLE 1 LIMITATIONS OF CURRENT CPOE SYSTEMS	9
TABLE 2 EXAMPLE OF A TEMPLATE CONSTRUCTED BY THE FACILITATOR.....	16
TABLE 3 OUTLINE OF THE DATABASE MANAGEMENT SYSTEM PROCESS	64

List of Figures

Figure	Page
FIGURE 1 DEMONSTRATES THE ARCHITECTURE OF OUR CCTS.	14
FIGURE 2 DIAGRAM ILLUSTRATING THE HIERARCHY OF MEMBERS IN THE SYSTEM.	15
FIGURE 3 DATA FLOW CHART	25
FIGURE 4 ENTITY RELATIONSHIP DIAGRAM	48
FIGURE 5 ENTITY RELATIONSHIP DIAGRAM 2	49
FIGURE 6 1-M RELATIONSHIP.....	52
FIGURE 7 M-N RELATIONSHIP	52
FIGURE 8 ADD NEW USERS TO THE SYSTEM.	76
FIGURE 9 CHANGING ROLES AND PERMISSIONS FOR USERS.....	77
FIGURE 10 SUBMIT A REQUEST TO THE SYSTEM.....	78
FIGURE 11 TRACK SUBMITTED REQUESTS	79
FIGURE 12 CONTENT BOARD PROCESSING FORM FOR APPROVED REQUEST.	80
FIGURE 13 UPLOAD LITERATURE RELEVANT FOR SPECIFIC REQUEST.....	81
FIGURE 14 PASS THE REQUEST TO FACILITATOR	82
FIGURE 15 MAKE A TEMPLATE FOR A GIVEN REQUEST USING ATTACHED DOCUMENTS.....	83
FIGURE 16 APPROVING TEMPLATE FOR A REQUEST.	84
FIGURE 17 REJECTED TEMPLATES AT CONTENT SPECIALIST.....	85
FIGURE 18 CONVERT TEMPLATE TO WORKING PRODUCT	86
FIGURE 19 TRACK A SPECIFIC REQUEST	87

CLINICAL CONTENT TRACKING SYSTEM – AN EFFICIENT REQUEST TRACKING VIA A GRAPHICAL USER INTERFACE

Saif Khairat

Dr. Chi-Ren Shyu, Thesis Supervisor

ABSTRACT

Between 44,000 to 98,000 Americans die each year due to medical errors, and about 1 million people are injured. (Institute of Medicine, 1999) Many hospitals use a paper-based system to make requests for Clinical Contents and to get approval from the different divisions at the hospital. Like any system, efficiency is the most important component. However, with advancement of today's information technology, paper-based systems could be greatly improved. Some limitations with the current system used in most hospitals are 1) difficult to track order entries made by users at each step, 2) time consuming because each entry has to get handed to the next department by hand or mail, 3) an archive of the elements used by the physicians can not be created, 4) if the order-entry fails at any stage the whole process has to start from the beginning.

To remove all these difficulties and increase efficiency; a new web-based software called Clinical Content Tracking system (CCTS) was developed to replace the current paper-based system. CCTS is a tool that helps physicians build clinical contents, in an electronic format, that will be then used in the CPOE system. This system allows users to track their orders, at which stage the order stands, and the current status of the order. This system records every possible action, approval, and rejection made to any order made by the physician. Moreover, a working product can be

made of a request in a relatively faster pace. The new system introduces the concept of archiving all Modules and Line Items that have been approved in the system. The ultimate goal for this research is to develop a general-purpose framework that can handle various logic flow, approval constraints, template creation, a library of the building units of a template and a tracking system. The tracking system will record every action made on a given order-entry, its date and status. The ability to track order entries will facilitate managing and processing patient's records. Moreover by implementing our CCTS, the ability of gathering, computing and analyzing data with just a few clicks is very feasible. This will facilitate gathering information and transform data in the system to statistics and hence, maximize efficiency on the managerial level.

CHAPTER

1. Introduction

1.1 Motivations

A computerized physician order entry, (CPOE), is a process of electronic entry where physicians enter their instructions to treat patients under their care. These entries are stored in a database and accessible by all users, including stakeholders, boards and the different departments that have to approve the entry through a computer network. CPOE has been in high demands recently due to the many benefits it brings along. Some of these benefits are reduction in potential errors; decrease in process time; error-checking for duplicate, incorrect doses, or tests; and easy access to all related records remotely.

In a paper-based order entry system “physicians and nurses rely mostly on verbal communications to coordinate their actions.” (Beuscart-Zéphir, Pelayo *et al*, 2005) On the other hand, a CPOE system allows physicians and nurses to work independently while intercommunication is still occurring. Such a system is the key component that coordinates their actions. This feature reduces potential errors and increases efficiency since a CPOE system will provide functionalities that coordinate all members in the system.

A study conducted on Physician Order Entry (POE) systems in hospitals shows that the revenue from both inpatients and outpatients increased after the introduction of POE. (Park, Kim *et al*, 2003) If a paper-based system can increase revenue in a hospital then this indicates that the use of CPOE in

hospitals will increase revenue to the hospital, at least in the same amount as POE, which as a result increases profit. A live example can be found in Brigham and Women's Hospital (BWH) in Boston. BWH spent \$11.8 million to develop, implement, and operate CPOE. Over ten years, the system saved BWH \$28.5 million for cumulative net savings of \$16.7 million and net operating budget savings of \$9.5 million given the institutional 80% prospective reimbursement rate. (Kaushal, Ashish *et al*, 2006)

Another reason why CPOE is one of the key components in a modern health informatics system is users can have remote access to records in the system from multiple departments simultaneously. Also, CPOE systems facilitate safer data and improve patient data confidentiality by allowing only authorized users to view files. In addition, it is well recognized that a CPOE system can minimize required paper storage as well as losing entry records.

CPOE systems are usually designed to serve as a time-saving interface that allows clinicians to access the data they need without having to wade through enormous paper files. The quick navigation system reduces fetching time, which, in turn, will speed up process time and will lead to faster and more efficient care delivery. This will be of great benefit to patients with critical conditions require special attention and who need to be helped in a timely manner. Implementing a CPOE system will have a significant effect on the efficiency of workflow for critically ill patients. (Ali, Mekhjian *et al*, 2005)

Nearly 85% of hospitals and an even greater percentage of ambulatory clinics have yet to implement electronic health records (EHR) and interlinked

computer provider order entry (CPOE) systems (Ash, Bates, 2005; Chaudhry, Wang, Wu *et al.*, 2006). This means that most hospitals still use paper-based systems to handle their requests. Paper-based systems, especially in the Health Informatics field, tend to be error prone and inefficient. On the other hand, computer-based systems tend to provide reduction of costs, time, redundant medical services and, ideally, the duration of treatment. (Schabetsberger, Ammenwerth, 2005)

EHR has been in high demand recently because it provides health institutes and organizations with cutting-edge technology. This results in reducing errors and increasing productivity thus, an enhanced quality of work is produced. The high quality of clinical contents, such as order entries for patients and instruction files from clinical specialists as well as, the minimal time consumption will be two major benefits. “The greatest challenge in the new world of integrated healthcare delivery is to provide comprehensive, reliable, relevant, accessible, and timely patient information to each member of the healthcare team.” (Schloeffel *et al.*, 2001)

Furthermore, CPOE systems represent a great potential for cost savings and using workplace efficiently instead of having piles of folders. “No longer are paper-based record systems fulfilling the needs of clinicians, and related healthcare workers” (Koeller 1). For that very reason, computer-based software that can resolve current problems has to be encouraged and given a chance to prove its efficiency.

1.2 Need for Clinical Content Tracking System (CCTS)

Despite the desirability of implementing a new CPOE, only 9.6% of U.S. hospitals presently have CPOE completely available. (Ash, Gorman, 2002) Therefore, about 90% of hospitals in the U.S. still use paper-based order entry system for various reasons. This means that the need for a CPOE system that is efficient and can be implemented in any hospital, regardless of its size, is in very high demand.

A study conducted in 2004 suggested that implementing a CPOE system will result “in a significant reduction in medication turn-around times and medication errors for selected drugs”. (Cordero, Kuehn, 2004) We developed a pre-component of a CPOE system called Clinical Content Tracking System (CCTS). CPOE is a general name for all electronic order-entry systems, where as CCTS is the generic name for our new system that acts as a mediator between paper-based systems and CPOE systems. CCTS is a tool that helps physicians build clinical contents, in an electronic format, that will be then used in the CPOE system.

The reason behind naming the system CCTS is that, to our knowledge, this is the first system that provides most of the functionalities in a typical pre-CPOE system along with a detailed tracking system for any given order-entry.

The order can be tracked step-by-step from the time it enters the system until it reaches its final stage.

Consequently, CCTS will increase clinical content accuracy and reduce common errors in the current system which will result in providing CPOE systems with higher quality order entries. The computer-based system will increase efficiency to all its users; requests will be handled and managed in a more organized process and in a timely manner. A study shows that if physicians are required to navigate across multiple different systems to care for their patients, their willingness to participate will be lower. (Subramanian, Hoover *et al*, 2007) CCTS is designed so that members have their own work station in the system where they can handle all patient related work without navigating through multiple interfaces or systems.

The CPOE system does not differ from the currently used paper-based system when it comes to overall costs. “The significant cultural and workflow changes that accompany the implementation of POE did not adversely affect acuity-adjusted length of stay or total cost”. (Hagop, Rajee, 2002) So, in comparison, the CPOE will provide many more benefits to users than paper-based systems and yet will not significantly affect the total cost of the system which will indeed be preferred by Health Informatics Specialists.

With today’s system, managerial boards cannot easily produce statistics on large amounts of data in their system, let alone conduct research. By implementing a CCTS, the ability of gathering, computing and analyzing data with just a few clicks is highly feasible. Furthermore, users can display

information in a simple yet flexible layout and order that set of information based on the different fields.

1.3 Introduction Literature Review

A study conducted in 2004 suggests that there are five types of hospitals willing to implement a CPOE (Poon, Blumenthal, 2004): (1) hospital that fully implements a CPOE with no complications; (2) hospital committed to full implementation: full implementation not achieved, but budget and personnel committed to it; (3) hospital considering full implementation: full implementation not achieved, no resources committed to that goal, but planning to do so; (4) hospital attempted but abandoned: full implementation attempted, but not achieved and no longer under consideration; and (5) hospital never considered implementation. With such a division it is hard to implement a CCTS that can be used by every hospital in those five categories.

The CPOE system has not yet proven its ability to fully eliminate medication errors. Some systems have appeared to have a strong impact in eliminating medication prescribing errors (MPE) and rule violations (RV). However, when it comes to adverse drug events (ADEs) – a serious type of medical error; the same CPOE system shows less dramatic effects. (Potts AL, Barr FE *et al*, 2004) A year later, another study conducted showed high rates of ADEs may continue to occur after implementation of CPOE and related computerized medication systems (Nebeker, Hoffman *et al*, 2005)

More data has been reported from the Brigham and Women's Hospital (BWH) where a CPOE system was implemented in 1993. A study conducted seven years later shows that the frequency of serious medication errors in particular and medication error rate in general has substantially decreased.

(Kuperman, Teich et al, 2001) This significant success attributed by the CPOE system indicates that the door for improvement in reducing medication errors is still open with the traceable functionality to keep up-to-date progress of an order entry.

Currently there are not many commercial CPOE systems developed for hospitals. However, some of the systems available on the market have some potential problems that prevent them from performing with maximal efficiency. In a recent study (Guilherme, Roberto, 2005), researchers found that most CPOE systems do not use programmable decision-support infrastructures, and most health-care institutions do not have centralized and encoded clinical databases. For a CPOE system to be fully functional, a built-in decision-support system has to be included as well as a database that controls the decision making process since it has all the decision-making protocols and procedures stored in it.

A major limitation of most CPOE systems is the high cost and the lack of capital. Studies suggest that the cost of CPOE ranges from \$3 million to \$10 million, depending of hospital size and how well-built the IT infrastructure is. (Advisory Board Company, 2004) Many small to medium size hospitals will not be able to afford such a system. Thus, CPOE systems with such costs will only be implemented in certain hospitals with large funds. Evidently, this is not a practical solution.

In addition, hospital officials who have tried CPOE systems in the past complained about poorly designed user interfaces and unacceptable processing time. While vendor products are definitely improving over times, only some of

them were able to develop a system that can be implemented in more than one hospital. “Several chief information officers (CIOs) accused vendors of selling ‘vaporware,’ referring to software functionality that was promised but never delivered.” (Poon, Blumenthal, 2004) If CPOE systems are expensive and can not deliver a fully functional and independent system, then the market has no commercial product that satisfies users.

Finally, a survey conducted in 25 US hospitals that have experience with CPOE systems states the concern about physician rebellion and difficulty training physicians, particularly at community hospitals. (Poon, Blumenthal, 2005) Complicated and hard-to-learn systems discourage Physicians to learn how to use the software and thus, negatively criticize the system.

Table 1 Summary of limitations in current CPOE systems and solutions to those problems

Studies discussing CPOE systems:	Limitations	Our solution in CCTS
Guilherme, Roberto, 2005	No built-in decision support system	CCTS has a fully functional, built-in decision-support system backed by a Database Management System.
Advisory Board Company, 2004	High cost	Low cost
Poon, Blumenthal, 2004	Poorly designed user interfaces and unacceptable processing time	User friendly interface and short processing time
Poon, Blumenthal, 2005	hard-to-learn systems discourage physicians to learn	Designed under the Supervision of clinical content specialists to facilitate training

Table 1 shows the four studies conducted in recent years showing the limitations and shortcoming of some current CPOE systems. CCTS solutions to those limitations are shown in the last column.

1.4 Proposed System

We developed a CCTS which is designed to meet most requirements for successful, standalone software. This system is designed carefully to help users avoid most kinds of medical errors and to avoid many constraints that exist in other similar systems. Some current CPOE systems may lead to medication errors as a result of choosing the wrong drug or patient. CCTS minimizes the use of free text by users to help eliminate errors. (Shane, 2002) In sections where physicians fill an order or form, CCTS enforces two safety checks on data entered by the user. Moreover, the system provides a backtracking algorithm in the case of finding an error. These functionalities provide an improvement over some of the current limitations in CPOE systems in respect to errors accidentally caused by users.

Furthermore, the main motivation behind this research is to develop an adaptive system that can be configured or customized to meet the needs of various health institutes. We believe that this functionality is not available in other similar systems. Also, CCTS develops new electronic order sets by modify existing paper order sets through a governed structure. An order set is the grouping of patient orders for a specific diagnosis or condition.

Some current systems have a serious problem, which is not encouraging users to use the system. It is believed that the reason behind this is the absence in high creativity in the interface and a slow processing time. (Poon, Blumenthal,

2005) In order for a system to be successful, users have to feel comfortable using the interface. CCTS resolves these problems by having a web interface that is both consistent and easy to navigate through. The interface is divided into departments and each department has its own menu of options. As for the processing time, the system has a minimal waiting and processing time that sometimes can be non-noticeable.

We propose a system that will cost hospitals less compared to current CPOE systems being that the major factor stopping hospitals from implementing a CPOE system is the overall high cost. This research emphasized the development of a system that can be used by a large number of hospitals, regardless of their size or funds.

Common limitations in CPOE systems are caused by user unfriendly interfaces. The interface in a CCTS system is dynamic and user-friendly because of the standard self-explanatory menus and forms. This easy to navigate, user-friendly Graphical Interface encourages Physicians to learn the system rather than deter them which, in turn, will, reduce computer illiteracy. The processing time that Health Informatics specialists complain from in current CPOE systems has been a main goal in our development phase. That way data can be easily stored and retrieved from the Database regardless to the amount of data stored.

Furthermore, the CCTS has a decision-support system built within to ensure that the workflow is compatible with the original workflow in the hospitals. CCTS is designed to meet the needs and requirements of any hospitals. This is

accomplished by giving the administrator the ability to add or remove modules from the system as needed.

2. System Architecture

In this section we discuss the logic behind the design of CCTS and the components of the system. Our system has eight modules and an administrator role. The eight modules manage an order-entry in a sequential yet independent fashion. Each module has its own role and functionalities which are preset by the administrator who has the privileges to eliminate or add roles to the system. We foresee the importance of communication between certain modules in CCTS and therefore, a built-in communication method is supported.

Our system's architecture is divided into two partitions, a server-side component and a client-side component. The server side handles all requests made by clients such as requests to load a specific page. Requests made by clients can be divided into two categories, a request that can be handled by the server only and a request that requires interaction between the server and the database. A request to view all order entries in the system is a perfect example of a request that requires work from both the server and database. All requests made by the clients go to the server first, and then the server analyzes the request, determines how to gather the required information and then sends a response back to the client.

On the client side, there are two different types of clients, the administrator and regular users with no administrative privileges. Both groups send requests to the server and get a response, however, most requests sent by the administrator target the database more than the server where as regular users submit a combination of requests depending on their needs. The administrator would

typically manage new users, and make updates and changes to the various user related information, but does not have access to clinical contents. The database has a secluded space allocated for all administrative requests. All other requests made by non-administrator users are handled primarily by the server and the database when needed.

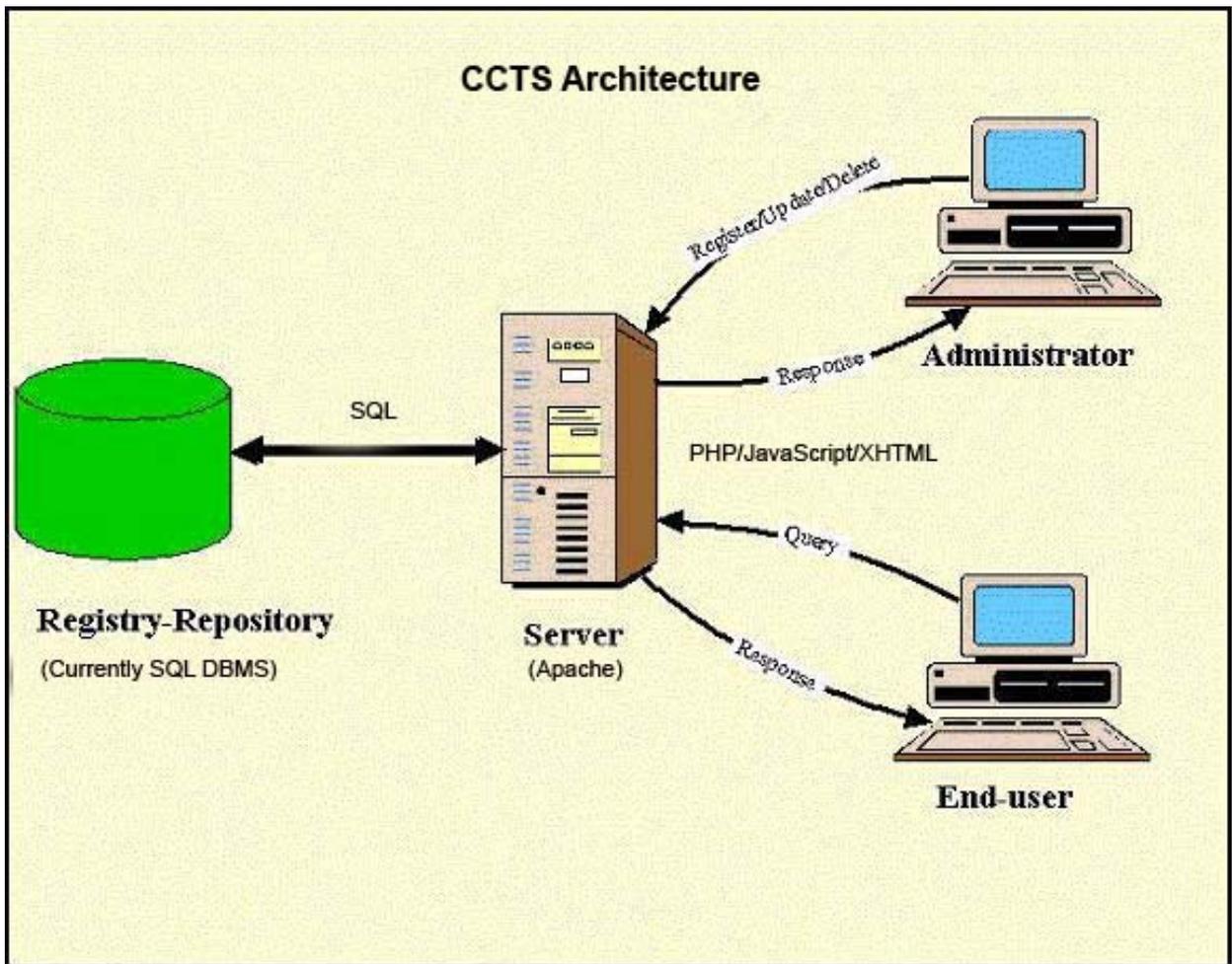


Figure 1 Architecture of our CCTS.

In order for an order entry to reach the last step in the system, the order entry has to go through our Electronic Decision Making process (EDM). This process can be defined as a list of sequential steps where approval of an order

entry at its current step is dependent on its approval in previous steps. The process starts when an order is submitted to the system and ends when the final product is created in the system. Moreover, during this EDM process every action made is recorded so that there is a live report on the progress of the order. In each process, there are eight stages, as shown in Figure2, that involve a certain group of people to provide feedbacks, comments, suggestions, and decisions. In this thesis, we use stage and module interchangeably. We also define people who participate in each module/stage as members of the module/stage.

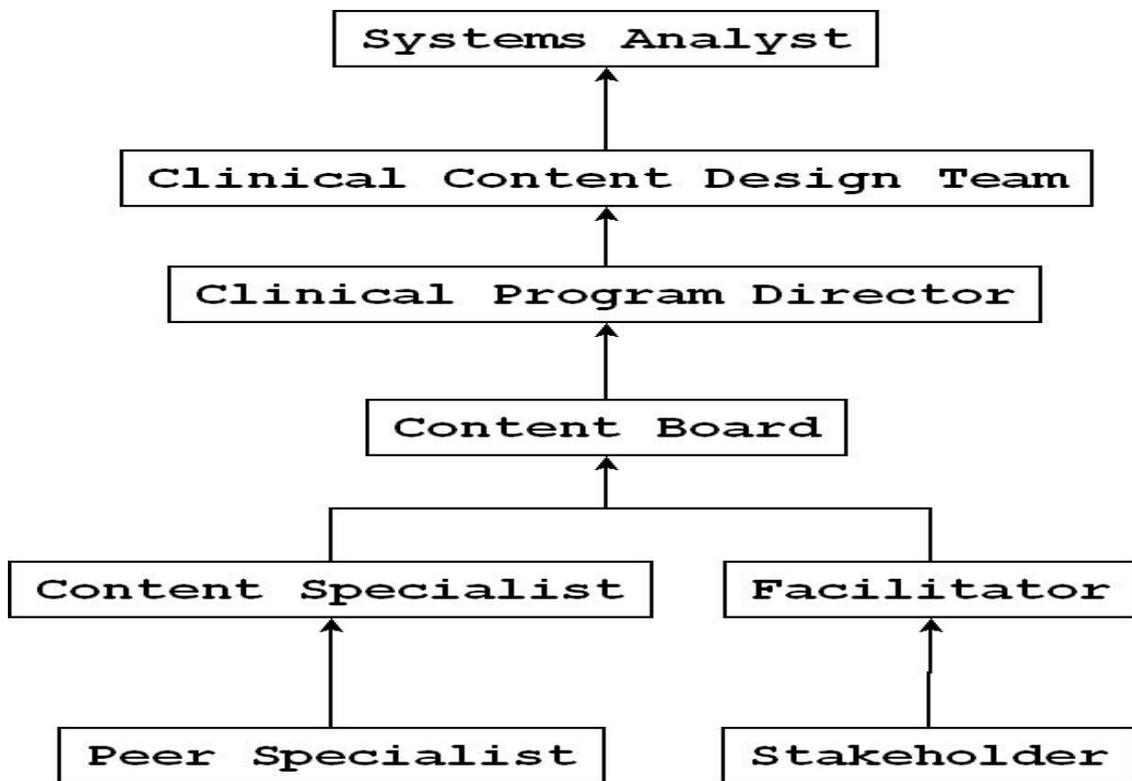


Figure 2 Diagram illustrating the hierarchy of members in the system. Peer specialist and stakeholder are the last people in the list while the systems analyst is the system member with the highest authority.

We will discuss the modules of the system using the following scenario:

A physician, or a stakeholder, submits an order-entry named “special surgery: diabetes surgery”. After submission, the content board receives the request electronically. The board meets and decides whether the request is necessary and should be approved or not. Upon approval, the content specialist creates an instruction file, with assistance from a peer specialist, which includes the elements required to fulfill the needs of this request. Then, the facilitator creates a template from the instruction file made by the content specialist. A portion of an example template is listed in Table 2.

Table 2 Example of a template constructed by the facilitator.

Formulary Oral Antidiabetic agents				
	Order	Dose	Route	Frequency
<input type="checkbox"/>	Metformin (Glucophage)	500mg	PO	tid
<input type="checkbox"/>	Metformin (Glucophage)	850mg	PO	tid
<input type="checkbox"/>	Micronized Glyburide (Glynase)	3 mg	PO	tid
<input type="checkbox"/>	Glyburide (Micronase)	2.5mg	PO	tid
<input type="checkbox"/>	Glyburide (Micronase)	5 mg	PO	tid
<input type="checkbox"/>	Glypizide (Glucotrol)	5mg	PO	tid
<input type="checkbox"/>	Glypizide (Glucotrol)	10mg	PO	tid
<input type="checkbox"/>	Glypizide Extended Release (Glucotrol XL)	2.5mg	PO	tid
<input type="checkbox"/>	Glypizide Extended Release (Glucotrol XL)	5mg	PO	tid
<input type="checkbox"/>	Glypizide Extended Release (Glucotrol XL)	10mg	PO	tid

Once the template is created the *Clinical Program Director* and the *Clinical Content Design Team (CCDT)* reviews the template to either approve or decline it. Then, the *systems analyst* will sign off on the template to approve that it is now

a valid working product in the system. Finally, the working product has to go through a final round of approval before the product can be used.

After submitting an order, the physician wants to know whether the request will be approved and when a final product can be delivered. Through our request tracking functionality, physicians can track the progress of their requests.

2.1 Groups/roles assigned to the users in the Database Management System (DBMS)

In Figure 2, the organizational hierarchy demonstrates the relationship among different roles in our proposed system. These roles are described in details below.

Stakeholder – anyone within the hospital that is allowed to make requests for order set reviews, amendments and new order sets. Eligibility for this role is contingent upon hospital policy. This would most likely include head physicians and nurse managers within each program or service area.

Content Board – this is a small group of senior decision-makers from various programs and service areas who oversee all clinical content reviews, revision and updates for their hospital. If the Content Board approves a request, they assign a Facilitator and Content Specialist.

Facilitator – a higher level administrative assistant who assists the Content Board by creating templates within the tool based on the instruction file passed by the content specialist.

Content Specialist – These would be knowledgeable, experienced clinicians assigned by their program or service directors to participate in the ongoing content review and updating process. Every Clinical Program Director or service area would have at least one content specialist, if not more.

Peers Specialist – These would also be knowledgeable, experienced clinicians with whom the Content Specialist would confer on clinical content questions. Every Clinical Program Director or service area would have a least one peer specialist, if not more.

Clinical Program Director - this group comprises one person, likely the director or the director's designate, from each Clinical Program Director or service area. This is the senior in charge for that area who makes the final review decision on templates for input into the CPOE Clinical Information System (CIS) and the final decision on the working product – i.e. the computerized physician order entry order set – within the new CPOE prior to CPOE go-live date.

Clinical Content Design Team - this group comprised by a small number of hospitals IT people and informatics-savvy clinicians who will work with a consultant and the CIS implementation team to oversee the CIS implementation within the hospital.

System Analyst – this would be the person from the hospital IT department. This individual is the primary liaison between the CCDT and the CIS Team. He or she could be a member of the CCDT or the hospital IT manager responsible for the CIS implementation. This would be the person who creates the working product, which is an electronic order set template that has been approved, and

delivers the working product to the CIS Team. A system analyst also closes the loop in this decision-tracking process in this database management system (DBMS) by entering a completed date when a working product has gone live in the CPOE.

CIS Team – this is not an entity or user within our DBMS. This is a separate entity consisting of members of an implementation team from the CIS vendor for the CIS that the hospital has purchased. The CIS Team will provide technical support of the CPOE system based on the working product that the System Analyst delivers.

Administrator – this is a designated hospital IT person responsible for the operation and maintenance of this DBMS tool. The Administrator is the only one with the ability to assign groups/roles to users, to change their roles and to change their passwords.

To connect the above roles with external consultants, we designed a Web-based MySQL database tool. The system will allow stakeholders to track clinical-content review decisions and facilitate the grooming of clinical content for implementation in a CPOE system within a CIS.

2.2 System Modules

The system consists of different modules where each module performs a group of unique and sequential tasks. A single request has to go through a certain path where a request will not proceed to the next unless it has been approved. In a case where a request is declined at a certain module, the system will return that request to the last module that approved it. For example, if Module R4 rejects

request 'A' then the system will automatically return request 'A' to Module R3. It is possible that a request can be returned to a module that is not immediately previous to the current stage. Any given request will have to go through all modules in the system at least once. Among the eight modules, four modules have higher authority where a request will have to be approved multiple times in order for the product to go live. These four modules, called Working Produce Approval Group (WPAG), include Content Specialist, Content Board, Clinical Program Director, and System Analyst.

2.2.1 Modules Description

The system is designed so that each module has two types of functionalities, namely general purpose and specific purpose. The general-purpose functions are the same for all modules. Those general functionalities are implemented in each module to unify the different independent body, that way the system can serve each and every module and guarantee communication between the different units.

Stakeholder

Every user in the system has the role of a Stakeholder assigned to them by default. A Stakeholder is the only module in the system that can generate a new request. A Stakeholder has to specify if the request submitted is a request for a new order set or a request to modify an existing order set. A preferred date of completion and a priority are required by the module user in order to notify the modules that follow with the level of necessity of getting the request done. After a request is being submitted by the user then the Stakeholder can monitor the

progress of the request through a tab that show how far the request has gone, percent completed as well as a status update. The status updates show which module the request is at and its status, i.e. failed, pending.

Content Board

This module consists of many members; the board will meet once every week and will review submitted requests from Stakeholders. This module will allow only one person, usually the highest in authority in the board, to sign off on any given request either by approving or by denying. The board will review a request and all the information that was submitted by the previous module. If the board decides to reject the request then the Stakeholder will be notified that the request was denied. However, if a request is approved then the board will assign a Content Specialist and a Facilitator. A new required date and priority will be given by the board based on what the Stakeholder had submitted. Once the content board signs off on the approved request then the request moves to the next module and the Content Board can monitor the progress of the request through a tab that show how far the request has gone, percent completed as well as a status update.

Another task done at this module is to view and approve the working product created in the CPOE system. Approving this product is essential for the product to pass to the next module in the cycle.

Content Specialist

This is the module where external sources are inserted into the system. A Content Specialist (CS) will receive the approved request from the previous

module. The CS uploads necessary literature that relates to the request. An electronic form has to be filled before any literature can be accepted in the system. The form includes important fields such as ISBN, author's last name and journal name. This information will then be stored in the database and could be used as tags to find articles in the future. Then the CS can upload with another literature review or consult with a Peer Specialist. Finally an instruction file is uploaded to the system. This instruction file is used by the next module to create a template.

Later in the cycle a working product will be created, this module will be the first to view the working product. The CS will then view the product and sign on to approve it.

Peer Specialist

This module primarily plays the role of an advisor of the CS. The literature review uploaded by the CS is sent to the Peer Specialist (PS) before the CS sends the request off to the next module. The PS will receive information submitted from Stakeholder and CB and review the documents used by the CS. If the PS decides that the documents used by the CS are not relevant then the PS will make suggestions to the CS and the CS will then make modifications and send the request to the next module.

Facilitator

At this point all necessary information are in the system and a new template can be made by the Facilitator. The Facilitator electronically receives the Literature documents and instruction file passed on by the CS. The instruction file contains the procedures necessary for the Facilitator to make a new template. Once the facilitator creates the template and signs off on it then it is passed on. The Facilitator's role in regards to that request is done.

Clinical Program Director

This module is the first stop for the newly created template. The Clinical Program Director (CPD) views the template that the previous module has created and the template can either be approved and moves up the ladder or rejected. If rejected the CPD has to give a reason for declining. The CPD also has a role in co-approving the product that was created in the system. This unit is the third module in the WPAG to sign off on the working product.

CCDT

The structure of this module is similar to that of Content Board. CCDT is a team to review a template. If the template is approved by the team, the team leader will sign off the template. The CCDT can view the template that has been approved by the previous module and then decide to either approve the template or fail it based on the feedback from its preceding comments. The CCDT is the only module that has the ability to upload a new instruction file with the new

modifications to the template. The instruction file will be sent to the CS along with a message explaining a reason for rejecting the template.

Systems Analyst

This is the last module in the process to view the template. At this point, it is believed that if the template has any glitches then these would have been caught by the previous modules. As a result, this module only has only one option and that is to view the electronic template then approved it. Once the SA approves of the electronic template a working product will be created in the CPOE system, at this point a new cycle will take place in order to approve the working product. Finally, once the working product has gone through all steps it comes back to the SA to approve that this working product can now go live.

2.3 Data Flow

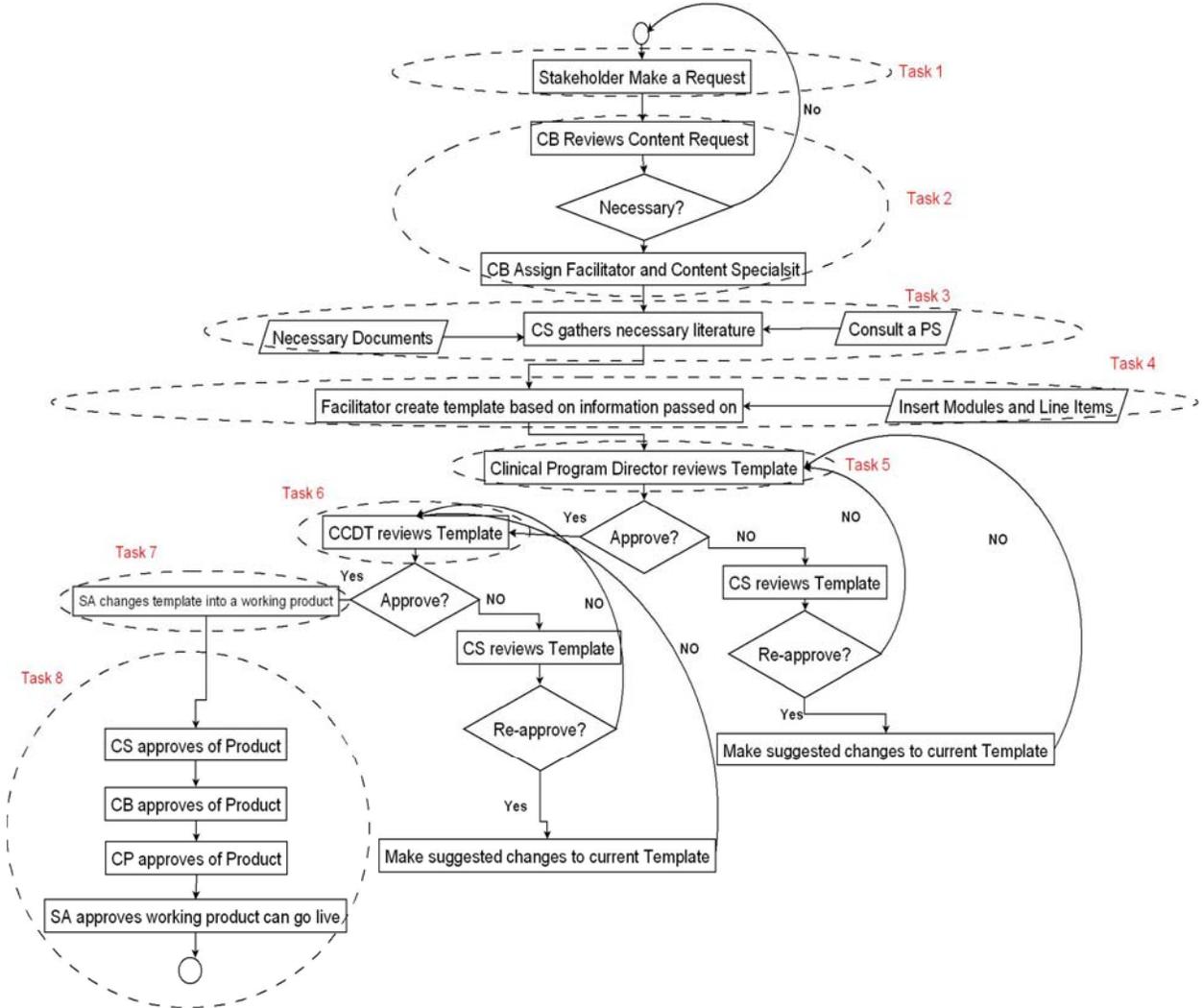


Figure 3 Data flow chart

Figure 3 shows the full cycle of any request from submissions time to when a product is created in the system. The diagram also shows all possible actions to a request for each task.

The system is designed to be flexible to be customized for various settings of CPOE systems. In this section, we discuss the possible paths where a request can travel from the beginning stage to the final stage where a working product is created. To better explain this section, we introduce the concepts of general and

specific approaches. The former is adaptable for any CPOE setting; while the latter is specific to the system we developed based on domain expert's input. As shown in Figure 3, there are eight tasks (modules) which are explained in details below.

Task 1:

General Approach:

By default each registered user in module R1 has the privileges to submit a request. After the request has been submitted, the member of R1 can track the request at any stage of this EDM process.

Specific Approach:

In order for a stakeholder to submit a request a form has to be filled out. The stakeholder will give a name, a brief description, a preferred completion date and a priority for the request. The name given by the stakeholder will be the id used for this request throughout the rest of the cycle. Once the request is submitted the stakeholder will be able to track the request, status, which module is currently working on the request.

Task 2:

General Approach:

Members in R2 receive all requests submitted by members from R1. It is the duty of R2 members to determine whether this request should proceed in the system or not. If the request will proceed then a user from modules R3 and R4 will be assigned to the specific request.

Specific Approach:

The request goes to the Content Board (CB) who will review all the information from the Stakeholder. The CB has two choices, first it can decline the request and in this case the request will no longer be functional and the Stakeholder will be notified that the CB has declined their request. If the CB thinks the request is valid then the board will approve the request. Upon approving the request, the board will be asked by the system to assign a Content Specialist and Facilitator to the request, give a preferred completion date and priority to the request. The preferred date and priority given by the board could be different than that of the Stakeholder and therefore, since the board has a higher authority than the system will use the CB information for the rest of the cycle. The next step in the cycle is the Content Specialist (CS).

Task 3:

General Approach:

At this step a unique member in module R3 is responsible to gather all the necessary information about the request information such as background information, literature review and which modules and line items are needed by the module R4. R3 member consults a specialist about the literature information gathered before the request can be submitted to the next module.

Specific Approach:

The assigned CS will electronically receive the request with the necessary information from the CB. The CS will have to do three steps before the request can move to the next module. The first is to add literature, such as journals or

papers that relate to the request. A form has to be filled by the CS and the document can be uploaded into the Database. Secondly, an instruction file that contains instructions to the facilitator is required by the CS, multiple documents can be uploaded to the Database. Finally, the request and its documents will be sent to a Peer Specialist to give their opinion on whether the uploaded documents are necessary or changes should be made by the facilitator. After the three steps are completed then the request can move to next level in the cycle which is the Facilitator.

Task 4:

General Approach:

Module R4 will translate the documents uploaded by members in R3 into a template, as shown in Table 2 that has a name, modules, and line items as the building blocks.

Specific Approach:

The facilitator will view the instruction file passed on by the CS and according to the contents of the file, a template will be created in the system. After this step the request will be replaced by the newly created template. In other words,

Task 5:

General Approach:

The members of Module R5 will view the template that was designed and then created by R3 and R4, respectively. Members of R5 can either fail the template and send it back to R3 or approve it and submit it to the next level.

Specific Approach:

The template will be reviewed by the Clinical Program Director (CPD), if the contents of the template are incomplete or the CP thinks that the template should be modified then the CPD will write a reason for their decision and fail the template. The template will then go back to the CS, who will receive the message from the CP. The CS can either re-approve the template and at this point it will go to the CP, or the CS can do the modifications and then send it back to the CP. If the CP approves the template then it will go to the Clinical Content Design Team (CCDT).

Task 6:

General Approach:

Module R6 reviews the approved template and decides whether it is valid to become a working product or not. If the template is invalid, the template is returned to the member of R3 who designed the template for modifications then comes back to R6. After validating the template, the member submits the template to the next module.

Specific Approach:

The CCDT will review the template, if the contents of the template are incomplete or the CCDT thinks that the template should be modified, and then the CCDT will write a reason for their decision and fail the template. The template will then go back to the CS, who will receive the message from the CCDT. The CS can either re-approve the template and at this point it will go to the CCDT or the CS can do the modifications and then send it back to the CCDT. If the CCDT approves the template then it will go to the Systems Analyst (SA).

Task 7:

General Approach:

R7 is the final module to review the template. The template will be approved after it has been approved all the way to R7.

Specific Approach:

Health Professions believe that by the time the template has reached the SA, it is error-free. Therefore, the system only allows the SA to sign off on the template and then the template will become a working product.

Task 8:

General Approach:

To have a final round of approval before going live, the template that survives from Task 1 to Task 7 has to be reviewed again by a sequence of members in R3, R2, R5, and R7. After R7 signs off on the product, there exists a working product in the system for that specific request. The product is then handed off to the CIS group who will use it to serve their necessities.

Specific Approach:

The working product will then go through a series of steps until it goes live. The working product will be sent to the CS to approve it. If it is approved then the CB will be the next to receive the product then the CP will sign off on the product and

finally the SA will approve the working product. By signing off, the working product goes live.

3. Implementation - Programming

3.1 Programming Mechanisms

3.1.1 Languages used

Since this system is primarily designed to track Clinical Content, the system receives many inputs from external sources. Furthermore, for every request there is a lot of information to keep track of. Therefore, the Graphical User Interface (GUI), which the user interacts with, is connected to a database that was created specifically for this system.

The system infrastructure is divided into three different parts: design, user interface, and database management. The design of the website was created using Cascading Style Sheets (CSS) and HTML. Hypertext Preprocessor (PHP), HTML, JavaScript was used to build a user friendly interface. A multithreaded, multi-user SQL database management system (DBMS) called MySQL was used to access and modify the Database.

3.2 Code Snippets

In this section we discuss key components of the entire system which has 5000+ lines of source code.

3.2.1 Functionalities

3.2.1.1 Pseudo code

In this section, we show high-level description of some of the algorithms used in our system.

Add user:

0. If (required fields != NULL) then {
1. If (all fields are validated) then{
2. If (username is not in the system) then {
3. User added successfully and Roles assigned to user
4. }
5. else {
6. Username already used
7. }
8. else {
9. forms fields are not valid
10. }
11. else {
12. Please fill in all required fields
13. }

An administrator can add a user through the interface. In order to add a user the administrator has to fill out a form, give a username and password to the new user and assign all necessary roles to the user.

Submit Request:

0. If (required fields != NULL) then {
1. If (all fields are validated) then {
2. Request stored into DB , sent to the next Module
3. Request tracking is launched
4. }
5. Else {
6. Validate form fields
7. }
8. Else {
9. Fill in all required fields in form
10. }

The personnel who submits the request needs to fill out an online form. This form has required fields that have to be submitted. The form makes sure that all fields are not empty. Then, using JavaScript, the form is validated to ensure that each field contains correct information. For instance, the preferred completion date has to be after the date when it was submitted.

Module 2 approving Request:

Connect to DB

Retrieve Content Specialist and Facilitator names from DB

0. If (Content Specialist name == NULL || Facilitator name == NULL) {
1. Form cannot be processed
2. }
3. Else if (Fields are not valid) {
4. Error Message
5. }
6. Else {
7. Process form. Store necessary information in the DB.
8. Send the Request to Content Specialist first, then to
 Facilitator.
9. }

If the Content Board decides to approve the request, then a Content Specialist and Facilitator have to be chosen. Then, the form will be filled based on information coming from the request submitter. When the form is ready to be processed, all the information is captured and stored in the Database and the system then sends the request to the “assigned” CS. When the CS is done with the request, the system then sends the request to the “assigned” Facilitator.

Content Specialist:

0. Add Literature();
1. ConsultPeer();
2. UploadFile();
3. If (Add Literature != 'Completed' && Consult with Peer != 'Completed' &&
4. Add Instruction File != 'Completed) {
5. Action on request is incomplete => User unable to submit Request.}
6. Else {
7. Store Files and their necessary information in Database.
8. Record all actions made on the request and their dates.
9. Pass the request to the next user assigned to this request.
10. }
11. AddLiterature(
12. If(required fields != NULL) {
13. If(all fields are validated) {
14. If(files uploaded has the correct extension) then {
15. All information are store in the Database
16. User given permission to consult a Peer.
17. else {
18. Upload file with the correct extension
19. }
20. else {

```

21.     JavaScript command used to specify which fields need to be validated.
22. }
23. else {
24.     Message box (Java) used to specify empty fields. User types what they
        want to
25.     Enter and information directly goes to the right place.
26. }
27. ConsultPeer(){
28.     Connect to DB; retrieve all Peer Specialists in the system.
29.     If ( Peer Specialist NOT selected && Message == 'NULL') then {
30.         Error;
31.     }
32.     else{
33.         Send information to selected Peer Specialist.
34.         Store all information in DB
35.     }
36. }
37. UploadFile(){
38.     If (file == Empty) {
39.         Error;
40.     }
41.     else if (File extension is incorrect) {
42.         Error;

```

```

43.     }
44.     else {
45.         Upload file and store its information in DB.
46.     }
47. }
48. );

```

The Content Specialist (CS) has to go through three sequential processes in order to pass the request to the next user in the chain. First, the CS has to upload one or more literature reviews that are related to the request. Java and JavaScript are used to validate the form and to ensure the correct information has been entered. Once the CS submits one Literature Review they have the option to either submit more reviews or move to the next step. Secondly, the CS consults a peer about the request and the literature review. Finally, an instruction file has to be uploaded to the system; the file consists of instructions to the facilitator to help them build the template. Once all three steps are completed the CS can now submit the request and send it to the Facilitator who was assigned by the Content Board to this request.

Create a Template

```

0.     ShowFile();
1.     Lineltem();
2.     If(required fields != NULL) {
3.         If(all fields are validated) {

```

```
4.         If (action = "Add") {
5.             Insert (OrderSet Name, Module Name, Line Item name) into the
           DB }
6.         If (action = "Delete") {
7.             Delete selected information from DB
8.         }
9.         If (action = "Edit") {
10.            Update necessary information in the DB
11.        }
12.        If (action = "Save") {
13.            Save all changes made to the template and final template in the
           DB.
14.        Send Request to next Module.
15.        }
16.    else {
17.        Form Fields not valid
18.    }
19.    else {
20.        Required forms fields are missing.
21.    }
22.    ShowFile(){
23.    Get RequestID;
```

24. Fetch for last instruction file submitted for a request with the given RequestID.
25. Display Instruction file in frame.
26. }
27. LineItems(){
28. If (action = 'Add') {
29. Add new input box field for new Line Item
30. }
31. If (action = 'Remove') {
32. Delete last input box field inserted.
33. }
34. }

In order for the Facilitator to come up with a new Template they have to follow the instruction file uploaded to the system by the CS. Therefore, we have decided to divide the page into two halves. The first half includes the form from which the template will be created and the second half will include the instruction file from the CS. This will increase efficiency and decrease the facilitators' chances of making typing errors.

In the first half of the page, there is the form on the left side and a view of the current template on the right. The view of the template is used to allow users to make changes to the template and therefore, as well as to give the user a chance for trial and error. Each template is named mandatorily with at least one module and one line item. If the Facilitator tries to submit a form that violates the

structure of the template then, his/her request will not be processed. When the facilitator submits a form, a message will be displayed to the user to confirm the request, if the user thinks they made a mistake they can easily click cancel and go back a step. If the Facilitator submits a form and then decides they would like to go back and add, update or delete it then they have the options to do that through three buttons named “add”, “edit” and “delete”.

Once the user clicks on the “Save Completed Template” button, the template will be stored as-is in the DB and the template will be sent to the next module in the list.

Approve/Fail Template

0. GetTemplate();
1. If(action = “Approve”) {
2. Date of approval and other information is stored in DB.
3. Send Template to the next module.
4. }
5. If (action = “Fail”) {
6. If (Message != ‘NULL’) {
7. If (Upload Instruction file = ‘True’) {
8. Display form.
9. Upload new Instruction file to DB.
10. }
11. else {
12. Message to CS is required

13. }
14. GetTemplate(){
15. Retrieve TemplateID for specific Request;
16. Display Template with TemplateID = 'TemplateID';
17. }

After the template has been created in the system, there are two different boards that have to approve the new template. The board will view the Template created by the facilitator, then the board can either approve or fail the template. If the template is approved then the date of approval by this board and other related information are stored in the DB, and the Template will move to the next Module.

If the Template fails then the Board has to give a reason for failing the Template. Furthermore, the board has the option of uploading a new instruction file to be used when re-creating the Template. When the Template fails, the Content Specialist who was originally responsible for the request will receive the template, message from board and the new Instruction file is applicable.

Failed Template

0. If(action = 'Edit') {
1. DisplayTemplate();
2. DisplayInstructionFile();
3. If(action = 'Delete') {
4. Delete selected Item;

```
5.    }
6.    If(action = 'Update) {
7.        Update selected Item;
8.    }
9.    If(action = 'Insert) {
10.        Insert Item;
11.    }
12.    If(action = 'Save) {
13.        Save new Template;
14.    }
15. }
16. If(action = 'Reapprove') {
17.     Reapprove current Template;
18.     Send message and Template to Module;
19. }
```

The Content Specialist (CS) who was assigned to the request receives the “Failed” Template and a message from the member at this Module who failed it.

The CS then has two options: either to re-approve the Template with no changes. (Lines 16-18) or to edit the current Template. The most recent Instruction file is retrieved from the Database so that the CS can make the changes recommended by the user. The CS can delete, update or insert items in the template. Once changes are made the Template is then sent to the Module who had failed the Template the first time.

Approve working product

0. Display working product Info
1. If (action = 'approve') {
2. If (user != last Module) {
3. Store information in DB
4. Send working product to next Module
5. Increment working product status to show progress.
6. }
7. else {
8. Display working product Info
9. working product goes Live in System
10. }
11. }

The working product (WO) has to go through multiple iterations of approvals from different Modules. If the Module signing off on the WO is not the last Module in the Cycle (lines 2-6) then, information from the template is displayed. If this Module approves the WO then it will go to the next Modules and the status of the WO is updated so that other Modules can track it. If the Module approving the WO is the last in the list then information of the Product is displayed. When it is approved, the working product goes live in the system.

3.2.1.2 GUI Design

As mentioned above, CSS was used to design the interface.

```
Body {  
    padding: 0;  
    font-family: Century Gothic;  
    font-size: 10pt;  
    background: #000066;  
    background: #FFFFFF;  
    background-color: #000066;  
}
```

```
input {  
    color:#000000;  
    background-color: chartreuse;  
    font-size:14px;  
    text-decoration:none;  
    font-weight:bold;  
    font-family: Century Gothic, verdana, arial, helvetica, sans-serif;  
    border-style:none;  
    line-height:120%;  
    cursor:hand;  
}
```

The design for the interface is divided into classes; each class is responsible for the design of certain parts of the website. In the previous code snippet, there are two classes. The first class takes care of the design of the whole page; Formatting factors such as font and background color are being defined in this class. The second class formats all input boxes used in the system. There are many forms that users have to fill out and therefore, this class is used in multiple occasions.

3.2.1.3 PHP Hypertext Preprocessor (PHP)

PHP Hypertext Preprocessor (PHP) is the programming language used to develop the majority of the CCTS web functionalities. PHP is used to validate forms, upload files to the system and improve security settings. The reason behind choosing PHP for the previously mentioned functionalities is because it provides a high speed of execution. PHP can perform one or more tasks in a relatively high speed and not slow down the rest of the machine by using its resources. Since process time is a point of interest, we thought PHP would be a good language to use due to its speed.

The CCTS stores information related to patient's records that have to be rigorously protected and kept confidential. PHP provides many levels of security that can protect the system against any malicious attacks. For that reason, PHP was chosen to serve that purpose.

Since potentially the CCTS can be used by any hospital whether it is small or large, we intend to use a programming language that is stable and available to run on almost all platforms, such as UNIX and Windows, will ease implementation for hospital with various platforms. PHP has its own resource management system which makes it highly stable. Moreover, it runs on almost all platforms and this was another reason why PHP was chosen to be used as the main programming language to develop the CCTS.

4. Database Management

Since the goal behind developing this system is to be used in hospitals and other health related institutes, we decided to use MySQL Database instead of Oracle due to the potential cost issues for small institutes. Since most hospitals do not have past experiences with a CPOE system which is new in its idea, we believe that by making this selection, we allow for the system to be available to a wide range of costumers for a minimal cost.

4.1 Entity Relationship Diagram

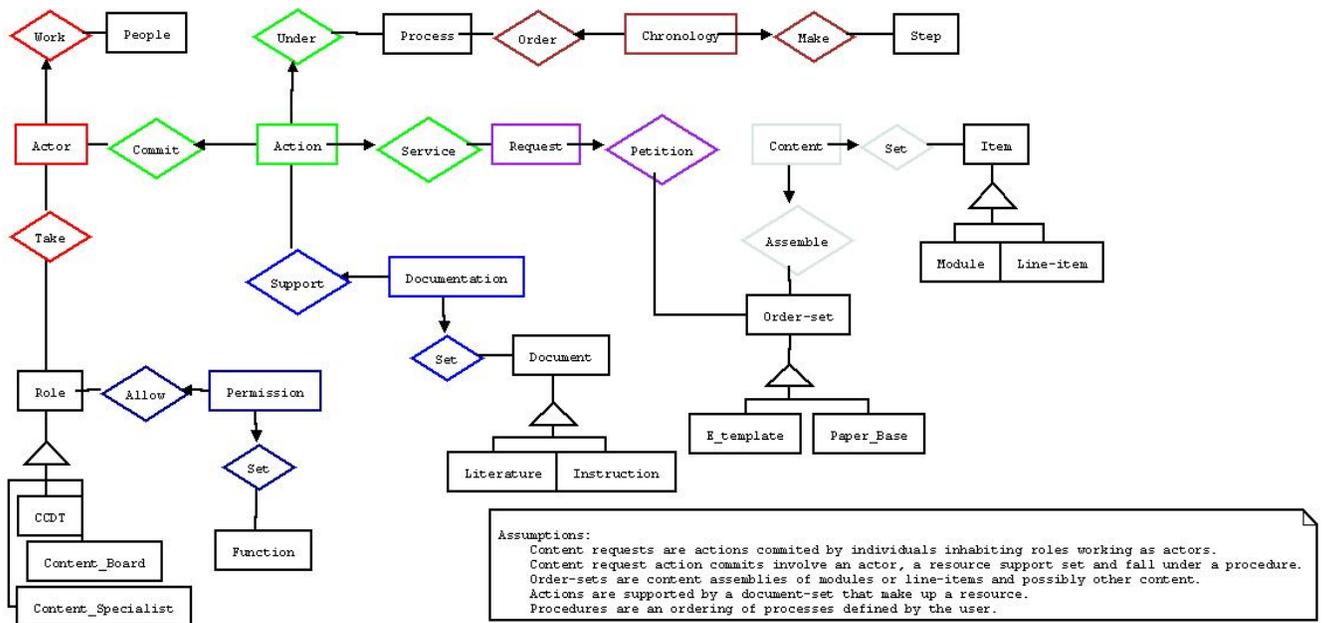


Figure 4 Entity Relationship Diagram

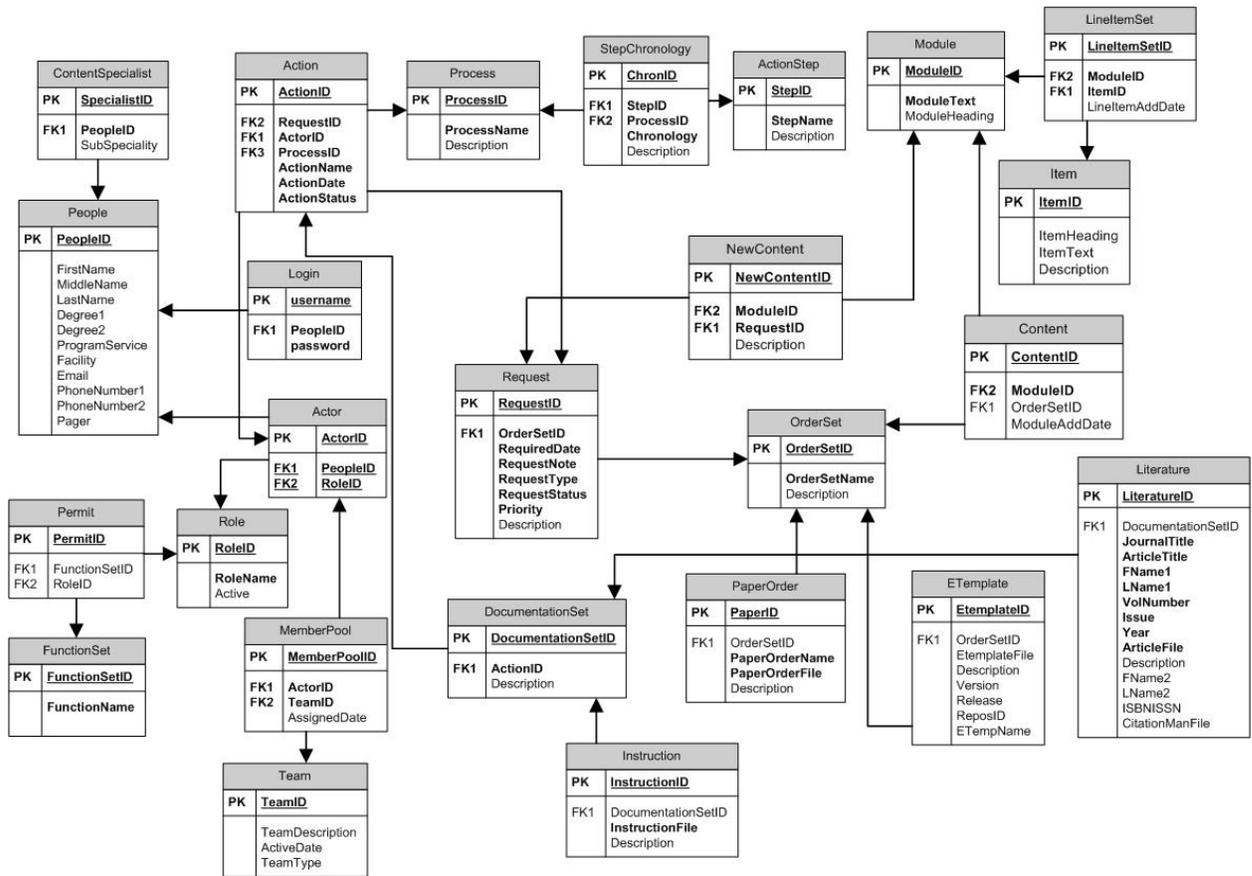


Figure 5 Detailed Entity Relationship Diagram

Figures 2 and 3 show the database schema. Tables in the database are represented by a rectangle shape, and actions are represented by diamonds. Some tables are linked to multiple tables using a triangle. The triangle signifies that the tables that are connected through the triangle all inherit the attributes of the main table. This technique is used when more than one table share the same attributes, so instead of having the same information in more than one table, these attributes are stored in one table and other tables inherit these attributes. This technique is expected to remove redundancy and increase database optimization.

4.2 Database Architecture

This system heavily relies on tracking a certain request and all the changes made to it from step one all the way to the end. Therefore, a well-constructed database had to be built to store all of that information. There are many relationships between the different modules in the system as well as between a given request, its template and the final working product. Therefore upon creating all the necessary tables, we found out that there are large numbers of Many-to-Many (M-N) relationships between some major tables. A relationship that is multi-valued in both directions is a many-to-many relationship. For example in Figure 4, there can be many requests assigned to each user and many users can act on any request. These relationships can be difficult to identify and to resolve and hard to deal with in such a complicated system. Also, Associative relationships such as One-to-Many (1-M) are more flexible in particular situations.

4.2.1 Database Optimization and Tuning

The ERD depicted in Figure 4 is a result of a series of database optimization, and tuning. For example, to ensure the efficiency of queries, we took several iterations of design to refine our database schema from 60 tables in the original design to roughly one third of the number of tables in the final design.

This optimization process includes consolidations of contents from various tables and restructuring the design to a star-like architecture where Table Request plays as a heart of the entire database activities. Moreover, a noteworthy optimization process was to provide version tracking of templates for a certain request using Table E_template which gives an extra dimension of advantages over the paper-based process.

To deal with M-N relationship, we add an extra table in between so that each table will have a one-to-many (1-M) relationship. An example of this process is shown in Figure 7 that illustrates a solution to Figure 6. In general, if tables T1 and T2 have an M-N relationship; a new table, T3, is created. This new table will have its own Primary Key (PK) as well as T1 and T2 PKs as its Foreign Key (FK). This way T1 and T3 have a 1-M relationship and as well as T2 and T3. With this, T1 and T2 would still be connected by a “middle” entity which still preserves each table’s keys.

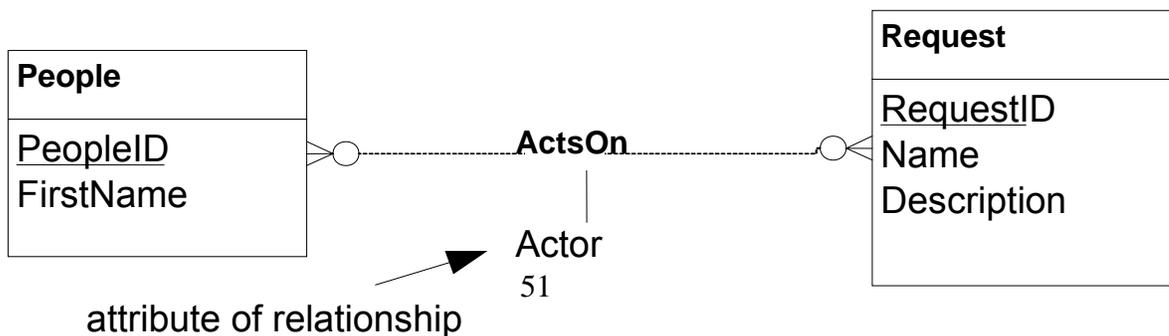


Figure 6 M-N relationship

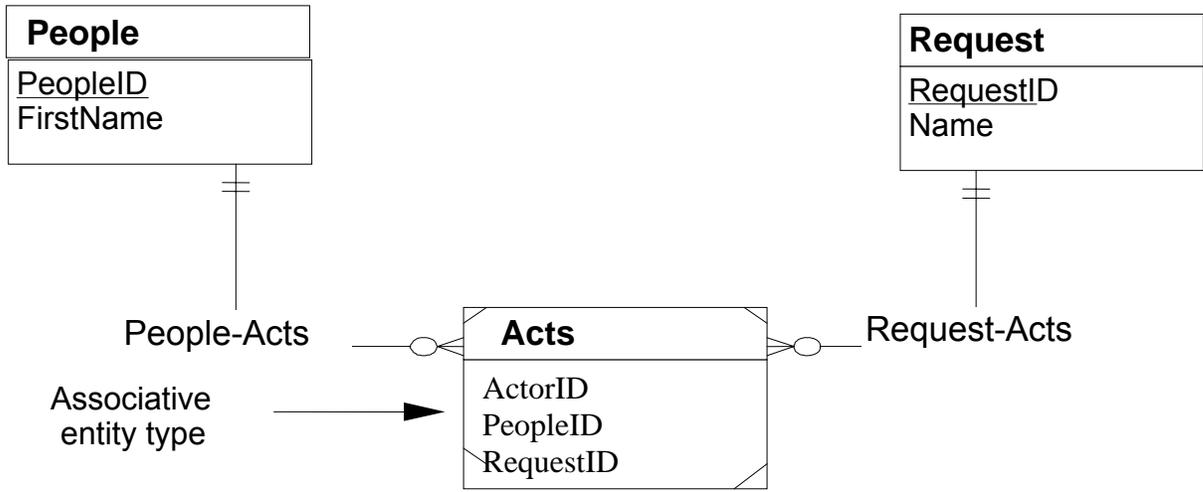


Figure 7 1-M relationship

4.2.2 Database Tables

4.2.2.1 Tables Description

This section discusses most tables in the Database, their attributes, functionalities and relationships with other tables. Before explaining each table, we would like to mention a few Entity Relationship definitions. A Primary Key (PK) is a column in a table whose values uniquely identify the rows in the table. A primary key value cannot be NULL. Also see candidate key. A Foreign Key (FK)

is a column in a table that does not uniquely identify rows in that table, but is used as a link to matching columns in other tables.

The People table stores all user information such as first Name, last Name and contact information. It has one Primary Key called PeopleID. The People's table has only one relationship with another table called Actor, where one system user can have one or more Roles.

The Role table has a composite Primary Key (Role ID and Role Name), its function is to identify what role each user has. Any given role can be taken by many users which mean that there is a M-N relationship between the two tables which requires a new table to break that relationship into 1-M and 1-M Role has an attribute named Active which is of type Boolean. If the key "Active" is true for one of the roles that means this role is active, otherwise it is inactive in the system

Actor is the table that links between People and Role. It has one Primary Key ActorID and three foreign key's PeopleID, RoleID and RoleName, in other words the PK of the other two tables.

CCDT, Content Board and Content Specialist are three sub-class tables in the Database that inherit attributes from its super-class table Role. These tables share the same primary key of the Role table. Some Modules in the system have an individual user assigned to it and others have a group of users. In the case of these three tables, there are groups of users assigned to each tables. Therefore, it is important to relate each of these tables to the correct RoleID in the system.

Under this setting, each Board or Team can act as one single group which is assigned to a specific role.

The ERD for this system is created so that it has one centralized table that links all major components in the Database. This table is the Request table which has a composite primary key with two attributes and four foreign keys. This table stores all necessary information about a certain request when it is submitted to the system until it becomes a working product. When a Content Specialist manages a Request, two of the three tasks they have to do are to upload supporting literature for the given request and an Instruction file. Those files will then be used by the Facilitator when creating a template for the request. For each request there is a unique number (DocumentationID) in the Documentation table. The documentation basically gives a specific number for the documents of each Request. Each of these two tasks has its own table, namely Literature and Instruction. A given request can have multiple literature files and Instruction files. Each of these files can be used for more than one request. Therefore, a table called DocumentationSet was created to act as a bridge of the M-N relationship between Tables Request and Documentation.

Once a request has been approved and a Template for that request has been made, a few tables have to hold the contents of that Template. Each Template consists of contents including Line Items and Modules. Each Request has a unique ContentID in the Content table. Line Items and Modules are stored in Tables LineItem and Module, respectively. Both tables use ContentID as their foreign keys.

Once a Template is created, it is stored in the E_template table which holds all versions of a given Template for a specific request. The table is linked to OrderSet which has an OrdersetID that can identify which request a Template belongs to by joining it with Tables Content and Request.

The system is designed so that there is a chronological order for the actions made on a request. These actions are stored in table Process and their order is stored in another - Step. For example, Process has 'Stakeholder submit Request' as a description to one of the processes and its StepID, or order in the list of action, is equal to one. Therefore, the first action on any request is to be submitted by a Stakeholder. When this task is done, StepID will increment by 1 and the next process will be applied to the request by sending the request to the correct module.

4.2.2.2 Data Definition Language (DDL)

Data Definition Language (DDL) is a part of the SQL language responsible for defining and managing all objects in a database. Since the Database used for the system was SQL, all 25 tables were created using MySQL syntax. The procedure used in creating the tables was to create the independent tables first, followed by creating the dependant tables. The reason behind taking that approach is that tables with dependencies need to link to a primary key in other tables. If the independent table with the primary key was not created before hand, then the link between the two tables will be lost. For example, the tables below:

```

1. CREATE TABLE People
2. (
3. PeopleID INTEGER,
4. HospitalID INTEGER,
5. Fname VARCHAR(32),
6. Mname VARCHAR(32),
7. Lname VARCHAR(32),
8. Degree1 VARCHAR(32),
9. Degree2 VARCHAR(32),
10. ProgramService VARCHAR(32),
11. Facility VARCHAR(32),
12. Email VARCHAR(32),
13. Phone1 INTEGER(15),
14. Phone2 INTEGER(15),
15. Pager VARCHAR(15),
16. PRIMARY KEY (PeopleID),
17. FOREIGN KEY (HospitalID) REFERENCES Hospital ON DELETE
    CASCADE
18. );

1. CREATE TABLE Role(
2. RoleID INTEGER,
3. RoleName VARCHAR(32) NOT NULL,
4. Active CHAR(1),
5. PRIMARY KEY (RoleID)
6. );

1. CREATE TABLE Actor(
2. ActorID INTEGER,
3. PeopleID INTEGER,
4. RoleID INTEGER,
5. PRIMARY KEY (ActorID),
6. FOREIGN KEY (PeopleID) REFERENCES People ON DELETE
    CASCADE,
7. FOREIGN KEY (RoleID) REFERENCES Role ON DELETE CASCADE
8. );

1. CREATE TABLE Process(
2. ProcessID INTEGER,
3. ProcessName VARCHAR(32) NOT NULL,
4. Description VARCHAR(32),
5. PRIMARY KEY (ProcessID)
6. );

1. CREATE TABLE OrderSet(

```

2. OrderSetID INTEGER,
3. OrderSetName VARCHAR(32) NOT NULL,
4. Description VARCHAR(32),
5. PRIMARY KEY (OrderSetID)
6.);

1. CREATE TABLE Request(
2. RequestID INTEGER,
3. OrderSetID INTEGER,
4. RequiredDate DATE NOT NULL,
5. RequestNote VARCHAR(32) NOT NULL,
6. RequestType VARCHAR(32) NOT NULL,
7. RequestStatus VARCHAR(32) NOT NULL,
8. Priority VARCHAR(32) NOT NULL,
9. Description VARCHAR(32),
10. AssignStatus char(1),
11. AssignDate DATE,
12. Standardizable VARCHAR(32),
13. PRIMARY KEY (RequestID),
14. FOREIGN KEY (OrderSetID) REFERENCES OrderSet ON DELETE
CASCADE
15.);

1. CREATE TABLE Action(
2. ActionID INTEGER,
3. RequestID INTEGER,
4. ActorID INTEGER,
5. ProcessID INTEGER,
6. ActionName VARCHAR(32),
7. ActionDate DATE,
8. ActionStatus VARCHAR(32),
9. AssignID INTEGER,
10. PRIMARY KEY (ActionID),
11. FOREIGN KEY (RequestID) REFERENCES Request ON DELETE
CASCADE,
12. FOREIGN KEY (ActorID) REFERENCES Actor ON DELETE CASCADE,
13. FOREIGN KEY (ProcessID) REFERENCES Process ON DELETE
CASCADE

The above statements were executed in order to preserve the referential integrity between the tables. For example, since table Action has a foreign keys

that references tables Request, Actor and Process therefore, those tables have to be created prior to the creation of table Action in order to successfully link the tables. Also, tables Request, Actor reference other tables, thus, referenced tables should be created first.

Moreover, since there are many relationships between the different tables, we have decided to use the 'on delete cascade' (line11-13 in Action) so that every time we remove any data from one table the corresponding data in other linked tables will be removed as well. For example, deleting ActorID in table Action will delete ActorID in Actor. This technique removes redundancy and duplication to ensure data integrity.

4.2.2.3 SQL Statements

The entire system is built of Database interaction; most pages either store information or retrieve information from the Database. Therefore, in this section we will display a few SQL statements, their relational algebra statements and a brief description of its functionality.

1. On the login page of the system the user types their username and password. This statement will return back the username of these credentials that the user has entered. If there are no records for these credentials then an error message is displayed.

Query 1:

```
SELECT
    L.username
FROM
    Login L
WHERE
```

```
L.username = '$userName' AND L.password = '$userPass';
```

2. This following statement is performed after the user has been granted access to the system. The statement gets the unique ID for this user in the system so that any actions performed by this user will be stored under his ID. This way every action taken can be back tracked to the user.

Query 2:

```
SELECT
    P.PeopleID
FROM
    People P, Login L
WHERE
    H.username = '$userName' AND H.PeopleID = P.PeopleID;
```

3. This query retrieves all the information attributed to a specific request submitted by a certain user.

Query 3:

```
SELECT
    R.CB_Priority,
    DATE_FORMAT(R.RequiredDate, '%m-%e-%Y') as
    RequiredDate,
    DATE_FORMAT(A1.ActionDate, '%m-%e-%Y') as ActionDate,
    DATE_FORMAT(R.RequestDate, '%m-%e-%Y') as RequestDate,
    DATE_FORMAT(R.CB_Date, '%m-%e-%Y') as CB_Date,
    R.Priority,
    R.RequestID,
    A1.ActionName,
    A1.ActionStatus,
    R.Description,
    P.FName,
    P.LName,
    P.MName,
    P.ProgramService,
    R.RequestType,
    R.RequestNote,
    R.RequestStatus,
    R.Standardizable
FROM
```

```

Request R, People P, Action A1, Actor A2
WHERE
R.RequestID = $requestID AND A1.ActionName='Assign to
Content Specialist' AND R.RequestID = A1.RequestID
AND A1.ActorID = A2.ActorID AND A2.PeopleID =
P.PeopleID;

```

4. When a request is submitted to the system the following query is used to insert all the information into the correct fields in the table.

Query 4:

```

INSERT INTO Request VALUES
($requestID',
'$requestID',
'$date',
'$Subject',
'$RequestType',
'1',
'$Priority',
'$Description',
'N',
', '$Standardizable', '$Date', ", ");

```

5. When the Content Board approves a request, the Board provides new information about the request, such as the priority for the request. This new information is entered into the table as new updates.

Query 5:

```

UPDATE Request
SET
Standardizable = '$standardizable', CB_Priority = '$priority',
AssignStatus = 'T', AssignDate='$today',
RequestStatus = '2', CB_Date = '$date'
WHERE
RequestID = $_SESSION[RequestID];

```

6. The following query shows all the literature reviews in the system in alphabetical order.

Query 6:

```
SELECT
    I.LiteratureID, I.ArticleTitle, A.RequestID, A.ActorID, I.JournalTitle
FROM
    Literature I, Action A, DocumentationSet D
WHERE
    I.DocumentationSetID = D.DocumentationSetID AND D.ActionID =
A.ActionID ORDER BY I.ArticleTitle;
```

7. This query displays the progress of all the requests in the system. The order of the requests can vary based on the category, such as name of the request or status.

Query 7:

```
SELECT
    R.RequestID, R.RequestNote, P.Fname, P.LName, R.RequestStatus,
P.Degree1, date_format( R.RequestDate, '%m/%d/%Y' ) AS RequestDate
FROM
    Request R, People P, Action A1, Actor A2
WHERE
    R.RequestID = A1.RequestID AND A1.ActorID = A2.ActorID AND
A2.PeopleID = P.PeopleID AND A1.ActionName = 'Initiate a Request'
ORDER BY '$name';
```

8. The system versions the different templates for a given request. This is done by giving a unique number to each version starting at 1.

Query 8:

```
INSERT INTO E_template
VALUES
    ('$EID', '$OrderSetID', '$content', '$OrderName', '1', "", "", '$OrderName',
'$Date' );
```

9. The following query displays all the Line Items associated with a given module for a specific request.

Query 9:

```

SELECT
    I.ItemText
FROM
    Item I, LineltemSet L, Module M, NewContent N, Request R
WHERE
    R.RequestID = '$requestID' AND R.RequestID = N.RequestID AND
    N.ModuleID = '$ModuleID' AND N.ModuleID = M.ModuleID AND
    M.ModuleID = L.ModuleID AND L.ItemID = I.ItemID;

```

10. This final query retrieves and shows the final working product for a given request.

Query 10:

```

SELECT
    E.ETempName, R.RequestType, R.Description, R.RequestID,
    date_format( R.RequestDate, '%m/%d/%Y' ) AS
    RequestDate,date_format( R.CB_Date, '%m/%d/%Y' ) AS CB_Date,
    R.CB_Priority, R.Standardizable, R.RequestNote, date_format(
    E.addtempDate, '%m/%d/%Y' ) AS addtempDate, P.Fname, P.Lname,
    P.Mname
FROM
    OrderSet O, People P, Request R, E_template E, Actor A1, Action A2
WHERE
    P.PeopleID = A1.PeopleID AND A1.ActorID = A2.ActorID AND
    A2.RequestID = R.RequestID AND R.OrderSetID = O.OrderSetID AND
    O.OrderSetID = E.OrderSetID AND O.OrderSetID = '$OrderSetID';

```

4.2.3 Security settings

A user can log in under different types of roles: an Administrator, a Stakeholder, a Content Board member, a Facilitator, a Content Specialist, a Peer Specialist, Clinical Program Director, a CCDT member, and a Systems Analyst. Depending on his or her role, each person is restricted to the role tabs and the corresponding functions in left-hand frame for that role. When a person logs in, the system checks to see which roles a person possesses. These roles are stored in an array which is then stored in a session variable. Later, when the person clicks on a specific role tab, the corresponding file will check the role session variable for that person to make sure he or she does indeed have access to that role's functions. This prevents a person from accessing functions for a role that he or she should not use or see.

Similarly, we use a session variable to keep track if a person is logged in or not. This session variable is checked in the same fashion as the role session variable. Each time a file is accessed, it first checks to make sure that the person is logged in. Additionally, we employ the use of MD5 encryption on user passwords. All passwords are stored in the database as a 32-bit encrypted string. When a user enters a password in the web application, it is immediately changed to an MD5 encryption variable in the PHP. This variable is then checked against the database to see if they match. This prevents any person from seeing the real string values for the passwords in the database. This is especially useful if the database were to be hacked. This prevents sending passwords in a non-encrypted format between the server and database.

We did not assign any system-level privileges mechanisms via discretionary access control during the development process.

4.3 Outline of the Database Management System Process

Table 3 Outline of the Database Management System Process

#	Actor and Action	DBMS Table	Attributes/Action Date
1	Stakeholder initiates Request – System auto assigns RequestID and dateAssigned The CB generates the initial list of content, although stakeholders may request up to 25% of the content.	(Request)	DateAssigned
2	Content Board Reviews Content Request and if they decide request is necessary, they assign a Facilitator and a Content Specialist The CB also collects any existing order sets in use, images them, and uploads them ‘tied’ to the content request.	(Request)	DateAssigned FacilitatorID DateAssigned SpecialistID
3	If Content Board decides request is unnecessary they hit failed request button. Request is flagged as failed.	(Request)	FailedDate
4	Once Content Specialist is assigned she can view existing order-sets that have been	(OrderSet)	

	imaged and entered into the OrderSet table by the CB.		
5	Then, Content Specialist gathers PDFs or paper articles to support her decision from the hospital library. The Content Specialist then uploads the literature into the Literature Table. The system generates a Date when the articles are uploaded.	(Documentation) (DocumentationSet) (Literature)	LiteratureID GatheredDate
6	The Content Specialist also checks with one or more Peer Specialists and indicates who she has consulted with by picking from a Peer Specialist Box. The system generates consult actionDate when Content Specialist has checked with them and selected them in the interface.	(Request)	PeerID actionDate
7	The Content Specialist then scans/inputs a file of her comments/instructions into the paperScan field of the Corroborates table. To finish the process, the Content Specialist either hits the “Pass to Facilitator” button and an actionDate is generated in the system OR if the Content Specialist concludes that no changes should be made to existing	(Documentation) (DocumentationSet) (Instruction)	actionDate OR FailedDate

	ordersets or no new ordersets created, he/she hits the Failed Button. Either way, by hitting one of these buttons the Content Specialist is saying that her decision is ready for the Facilitator.		
8	<p>The Facilitator is responsible for checking the system daily to see if the Content Specialists are finished corroborating or failing requests. For requests requiring further action, the Facilitator then creates an electronic order-set template after viewing the Content Specialist's comments/instructions. A createDate and templateID are system assigned to this new Template. The Facilitator pulls whatever required modules or line items are needed from the Modules table and Line Items table into select boxes and this Template is stored in the tempFile field in the Template table. The createDate is generated by the system when the Facilitator submits this Template to the tempFile.</p>	(Orderset) (Content) (LineItem) (Module)	CreateDate TemplateID OrderSet
9	The Template must then be validated by the Clinical Content Design Team and the	(Request) (E_template)	CcdtDate CcdtFailedDate DirectorDate

	Clinical Program Director. The CCDT will click passed or failed buttons.		
10	The Clinical Program Director will click passed or failed buttons. The Template must be validated by the Content Board by clicking the passed or failed buttons.	(Request) (E_template)	FailedDate
11	If it is passed by all three modules, the Systems Analyst uses the Template to build a working product order set. Once the Systems Analyst complete this task the SA clicks a button that adds a working product createDate that is tied to the TemplateID.	(Request) (E_template)	CreateDate ProductID
12	This signals the Content Specialist to review the working product then, clicks buttons to indicate she has approved the working product (that is tied to this Template) at this time.	(Request)	ActionDate
13	This signals the Content Board to review the working product and they click a button here to indicate they have either approved the working product at this time.	(Request)	ActionDate
14	The Clinical Program Director then reviews	(Request)	ActionDate

	and validates the working product and then clicks a button here to indicate he has approved the working product at this time.		
15	On the go-live date for the CIS, the Systems Analyst could then click on a button to generate an ActivateDate.		(Future work)
16	Eventually, after the CIS is determined to be fully implemented, for annual review or changes, the existing Order Sets in the CIS - via the request process - could be pulled from The CIS into this Decision-tracking DBMS to close the loop and provide ongoing tracking of electronic Order Sets.		

5. Conclusion, Limitations and Future Work

5.1 Conclusion

With today's advanced technology it is a surprise how some clinical institutes still heavily depend on paper-based forms. The need for a CPOE system in health institutes is growing rapidly for many reasons. The services provided by current paper-based systems are not sufficient enough to meet the requirements of the growing health informatics sector. To increase efficiency and productivity of order-entry systems, a computerized system with multiple interactive components is needed. The advantages of a CPOE include order legibility, improved response time, reduction in adverse drug reactions, reduced cost of care, and improved patient outcomes. (Foster, Antonelli, 2002) This research introduces a new system called Clinical Content Tracking System that to our knowledge has not yet been installed in most Hospitals in America. This tool helps physicians build clinical contents, in an electronic format, that will be then used in the CPOE system.

Our new system provides solution to limitations in current systems. Hospitals with experiences in CPOE systems suggest a methodology to avoid errors done by data entry specialist. A system that minimizes order-entry errors is expected to have a high overall efficiency. We believe that CCTS is designed to tackle this particular issue through methods and algorithms described in previous sections.

CCTS has a built-in request tracking feature that allows a member to track the progress and status of their request in the system. This feature increases system productivity because members can instantaneously find out if the order entry is valid or invalid and therefore, submit a new request or modify the previously submitted request.

Finally, prior to developing CCTS, a research was conducted to find common limitations in current CPOE systems and new needs for an up-to-date, efficient system. We believe that this research will contribute to the health system at large and hospitals in specific by providing a system that can carry out the expected tasks of a CPOE system.

5.2 System Limitations

One of the limitations of creating an XML library is that items inserted into the library might include errors. These errors can not be identified by the system because there is no standard library that items entered to the system can be checked against. Therefore, a specialist will have to approve each and every item going into the library to ensure it is correct both clinically and linguistically.

In addition, we want to create indexes and an interface to search the literature (e.g. stored scholarly articles, clinical guidelines) collected to support specific requests in additional ways (e.g. by article author, article title, journal title, name of content specialist who selected the literature and his or her program service, all literature collected over time and multiple review requests for an order set, etc.)

5.3 Future Work

In future upgrades, the system will tie into the CPOE system to close the loop and allow and trigger scheduled annual reviews of the order sets in the CPOE. Eventually, the system will allow the storage, review decision-tracking and grooming of clinical content other than order sets: e.g. medical logic modules and documentation templates.

Moreover, in order to create a Template in the system, the user has to type everything out from order set name to all line items. It is our goal in the future to create a library of all order set names, modules and line items entered in to the system through users. When a user wants to create a Template an Extensible Markup Language (XML) form with all the information in the library will be displayed to the user to choose from. This will reduce data redundancy in the system as well as make the process of making a new template more efficient.

An XML library categorizes all the modules and line items into classes. For example, Neurology, one of the categories in a template, can have 'x' number of modules and each module can have 'y' number of line items. A user who is willing to create a template with Neurology as one of its categories can find the category Neurology. The user can then find which modules and line items can be used. If none apply to this template, then the user can add new items to the library. Such a breakdown will facilitate the search for template items through a huge library.

Finally, it is essential to analyze our system against similar paper-based systems in order to evaluate our system. Therefore, we believe that an

experiment has to be conducted on a paper-based physician order-entry system and CCTS. The goal behind this experiment would be to show the comparison of overall efficiency, turnaround times, and advert event reduction between both systems.

Reference

- Ash JS & Bates DW. (2005). Factors and Forces Affecting EHR System Adoption: Report of a 2004 ACMI Discussion. *J Am Med Inform Assoc.* 12:8-12.
- Woong-Sub Park, Joon S. Kim, Young Moon Chae, Seung-Hum Yu, Chang-Yup Kim, Sang-A. Kim and Sang Hyuk Jung. (2003). Does the physician order-entry system increase the revenue of a general hospital? *International Journal of Medical Informatics.* Volume 71, Issue 1. Pages 25-32.
- Beuscart-Zépher MC, Pelayo S, Anceaux F, Meaux JJ, Degroisse M, Degoulet P. (2005). Impact of CPOE on doctor-nurse cooperation for the medication ordering and administration process. *Int J Med Inform.* 74(7-8):629-41. Epub 2005 Apr 14.
- Rainu Kaushal, MD, MPH, Ashish K. Jha, MD, Calvin Franz, PhD, John Glaser, PhD, Kanaka D. Shetty, MD, Tonushree Jaggi, BA, Blackford Middleton, MD, MBA, MSc, Gilad J. Kuperman, MD, PhD, Ramin Khorasani, MD, MPH, Milenko Tanasijevic, MD, MBA, David W. Bates, MD, MSc Brigham and Women's Hospital CPOE Working Group. (2006). Return on Investment for a Computerized Physician Order Entry System. *J Am Med Inform Assoc.* 13:261-266.
- B. Chaudhry, J. Wang and S. Wu *et al.* (2006), Systematic review: Impact of health information technology on quality, efficiency, and costs of medical care, *Ann Intern Med* 144, pp. 742–752.
- Ali NA, Mekhjian HS, Kuehn PL, Bentley TD, Kumar R, Ferketich AK, Hoffmann SP. (2005). Specificity of computerized physician order entry has a significant effect on the efficiency of workflow for critically ill patients. *Crit Care Med.* 2005 Jan;33(1):110-4.
- Thomas Schabetsberger, Elske Ammenwerth^a, Stefan Andreatta, Gordon Gratl, Reinhold Haux, Georg Lechleitner, Klaus Schindelwig, Christian Stark, Raimund Vogl, Immanuel Wilhelmy and Florian Wozak. (2005). From a paper-based transmission of discharge summaries to electronic communication in health care regions. University for Health Sciences, Medical Informatics and Technology (UMIT).
- Schloeffel, Peter, et al. (2001). Background and Overview of the Good Electronic Health Record. (http://www.gehr.org/Documents/BackgroundOverview_of_GEHR.htm) retrieved from the World Wide Web, October 31, 2007.
- Koeller, Rodney L. (2002). *IT Applications in Healthcare: The Electronic Medical Record.* U. of Maryland. <http://faculty.ed.umuc.edu/~meinkej/inss690/koeller.pdf>, retrieved from the World Wide Web, October 31, 2007.

Ash JS, Gorman PN, Seshadri V, Hersh WR. (2002). Computerized physician order entry in U.S. hospitals: results of a 2002 survey. *J Am Med Inform Assoc.* 2004 Mar-Apr;11(2):121-4

Leandro Cordero MD, Lynn Kuehn MSN, RN, Rajee R Kumar PhD and Hagop S Mekhjian MD. (2004). Impact of computerized physician order entry on clinical practice in a newborn intensive care unit. *Journal of Perinatology* (2004) 24, 88–93.

Hagop S. Mekhjian, MD, Rajee R. Kumar, PhD, Lynn Kuehn, MSN, RN, Thomas D. Bentley, MS, RNC, Phyllis Teater, MBA, Andrew Thomas, MD, MBA, Beth Payne, MS, RN, CCRN, and Asif Ahmad, MS, MBA. (2002). Immediate Benefits Realized Following Implementation of Physician Order Entry at an Academic Medical Center. *Am Med Inform Assoc.* 2002 Sep–Oct; 9(5): 529–539.

Sujha Subramanian, Sonja Hoover, Boyd Gilman, Terry S. Field, Ryan Mutter, Jerry H. Gurwitz. (2007). Computerized Physician Order Entry with Clinical Decision Support in Long-Term Care Facilities: Costs and Benefits to Stakeholders. *Journal of the American Geriatrics Society* Volume 55 Issue 9 Page 1451-1457, September 2007

Del Fiol. G, Rocha. R, Richard L. Bradshaw, Nathan C. Hulse, and Lorrie K. Roemer. (2005). An XML Model That Enables the Development of Complex Order Sets by Clinical Experts. *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, VOL. 9, NO. 2, JUNE 2005; 216-229

Kuperman GJ, Teich JM, Gandhi TK, Bates DW. (2001) Patient safety and computerized medication ordering at Brigham and Women's Hospital. *Jt Comm J Qual Improv.* 2001 Oct;27(10):509-21.

Eric G. Poon, David Blumenthal, Tonushree Jaggi, Melissa M. Honour, David W. Bates, and Rainu Kaushal. (2004). Overcoming Barriers To Adopting And Implementing Computerized Physician Order Entry Systems In U.S. Hospitals. 184-190

Advisory Board Company. (2001). *Computerized Physician Order Entry: Lessons from Pioneering Institutions* (Washington: Advisory Board Company, 2001), 11.

Potts AL, Barr FE, Gregory DF, Wright L, Patel NR. (2004). Computerized physician order entry and medication errors in a pediatric critical care unit. *Pediatrics.* 113(1 Pt 1):59-63.

Nebeker JR, Hoffman JM, Weir CR, Bennett CL, Hurdle JF. (2005). High rates of adverse drug events in a highly computerized hospital. *Arch Intern Med.* 2005 May 23;165(10):1111-6.

Eric G Poon, MD, David Blumenthal, MD MPP, Tonushree Jaggi, BA, Melissa M Honour, MPH, David W Bates, MD MSc, Rainu Kaushal, MD MPH. (2003). Overcoming the Barriers to the Implementing Computerized Physician Order Entry Systems in US Hospitals: Perspectives from Senior Management. AMIA 2003 Symposium Proceedings – Page 975.

Institute of Medicine. (1999) To err is human. Building a safer health system. Washington, DC: National Academy Press.

Rite Shane. (2002) Computerized physical order entry: Challenges and opportunities. Am J Health-Syst Pharm. 59:286-8.

Foster RA, Antonelli PJ. (2002). Computerized physician-order entry: are we there yet? Otolaryngol Clin North Am. 2002 Dec;35(6):1237-43.

Appendix System Demonstration

Hospital's Name Logout

Administrator Stakeholder Content Board Facilitator Content Specialist Peer Specialist CCDT Systems Analyst Clinical Program

Add New User

First Name:

Middle Initial:

Last Name:

Facility:

Degree 1:

Degree 2:

E-mail Address:

Pager: (optional)

Phone Number 1: Enter digits only. No hyphens, brackets, etc.

Phone Number 2: (optional)

Program/Service:

Please choose a unique username and password for this user.

Username:

Password:

Please select this person's roles (use control to select multiple roles).

Note: The role of Stakeholder will automatically be added for each new user.

Figure 8 Add new users to the system.

An administrator has to fill the above form in order to add a new user to the system. Each user has a unique username. Therefore, the administrator has to assign unique usernames. The administrator sets roles for each user. By default, each user gets the Stakeholder role assigned to them.

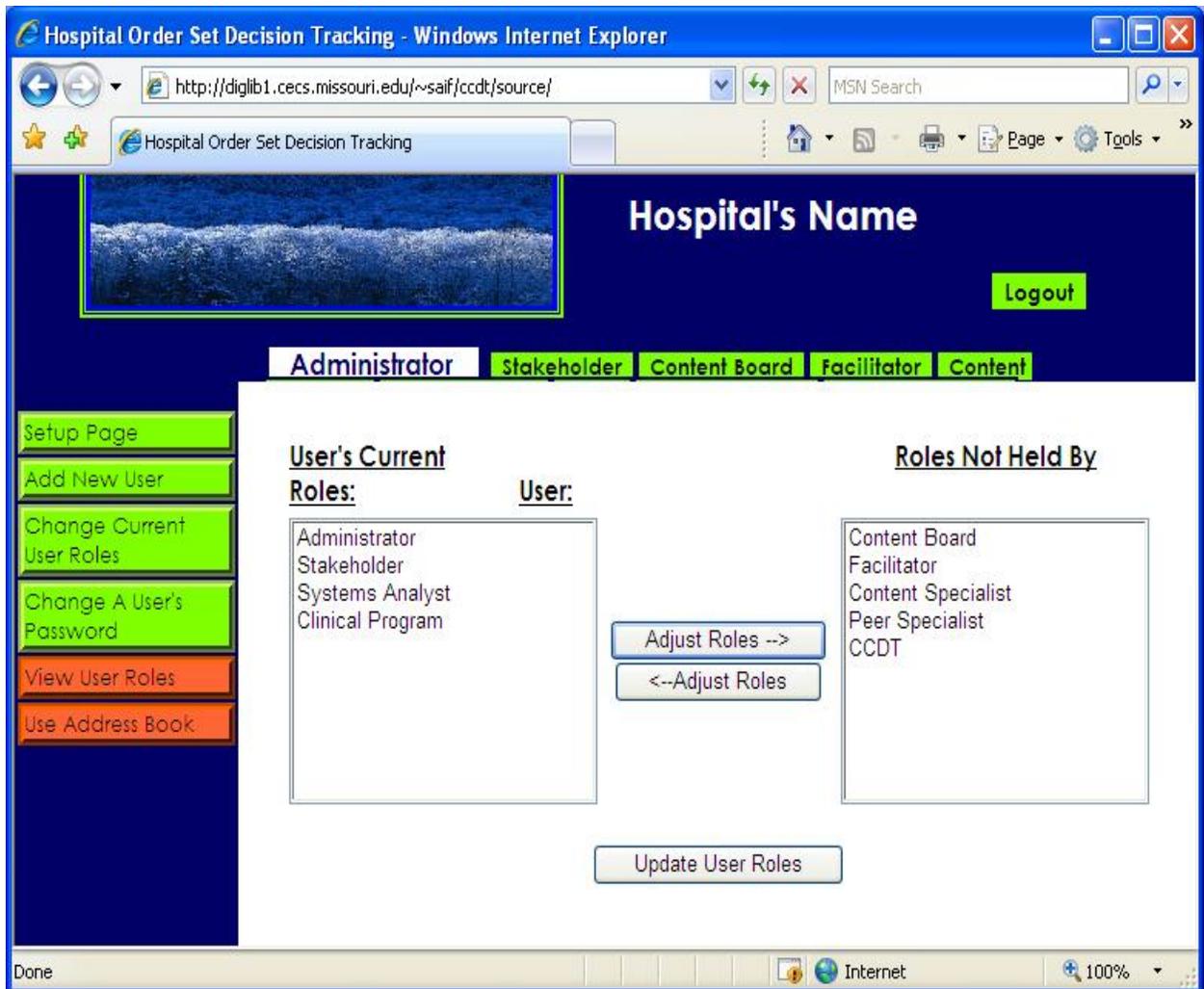


Figure 9 Changing roles and permissions for users.

In order to create a dynamic system, the option of changing roles for a certain individual had to be implemented. The administrator can view roles held by the user and accordingly can add or remove roles. The only role that must be assigned to any given user is the Stakeholder because every user should have the ability to submit requests and this is only done by the Stakeholder

The screenshot shows a web browser window titled "Hospital Order Set Decision Tracking - Windows Internet Explorer". The address bar contains the URL "http://diglib1.cecs.missouri.edu/~saif/ccdt/source/". The page features a dark blue header with a landscape image on the left, the text "Hospital's Name" in the center, and a "Logout" button on the right. Below the header is a navigation menu with buttons for "Administrator", "Stakeholder", "Content Board", "Facilitator", and "Content". On the left side, there is a vertical sidebar with buttons: "Make a Request" (green), "View Requests You've Made" (green), "View User Roles" (orange), "View Existing Order Sets" (orange), and "Use Address Book" (orange). The main content area is titled "Create Your Request" and contains the following form fields and options:

- Type of Clinical Content: Order Set Other
- Order Set Name:
- Please Explain Your Request:
- Preferred Completion Date:
- Request a New Order Set Modify Existing Order Set
- Preferred Priority: Low High Urgent
-

The browser's status bar at the bottom shows "Done", "Internet", and "100%" zoom level.

Figure 10 Submit a request to the system.

Figure 10 shows the form that any user has to fill to submit a request. This form is validated before it is submitted using JavaScript, all fields are required.

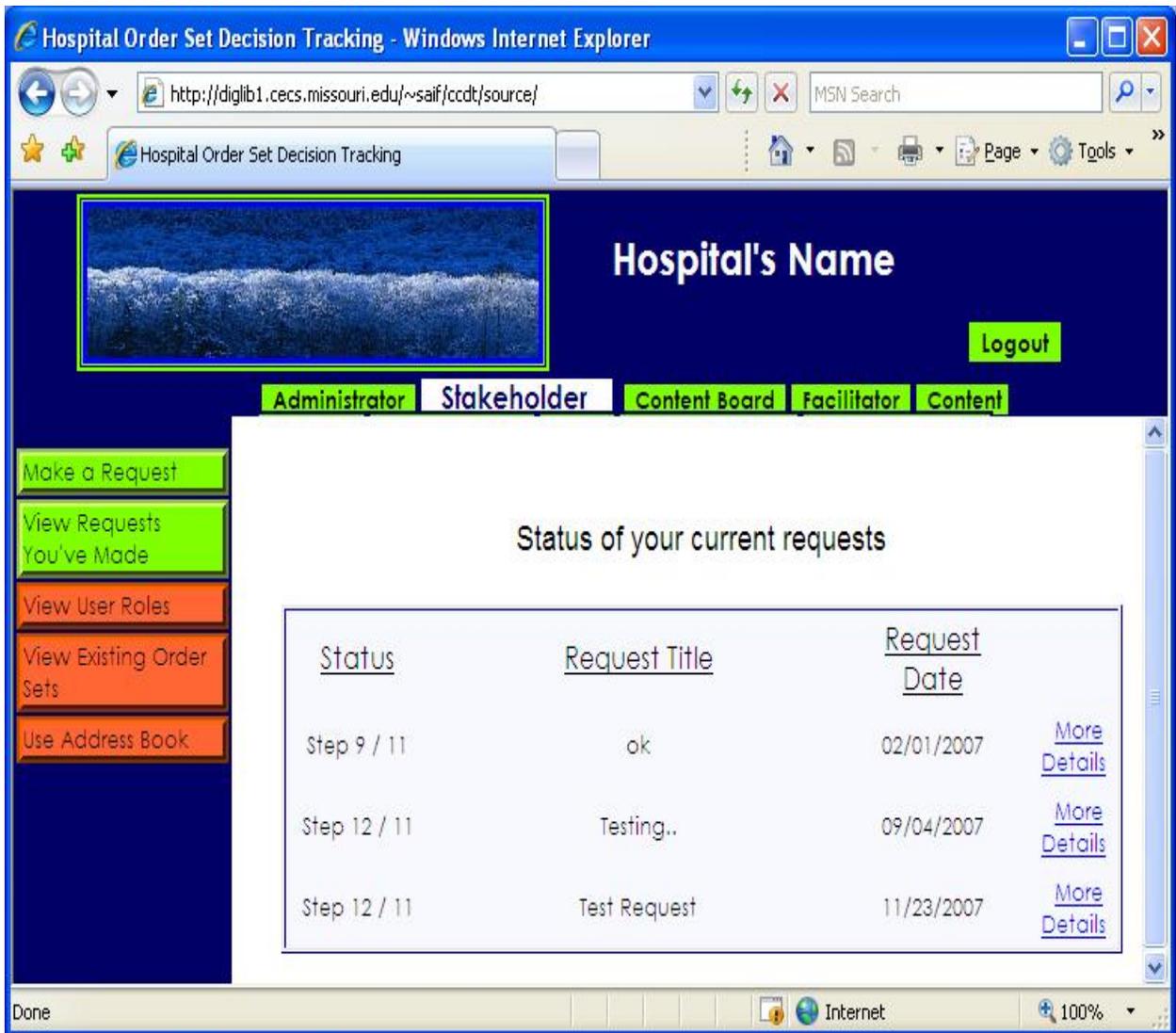


Figure 11 Track submitted Requests

After submission, Stakeholder can track any request they submitted and its progress. All information attributed to a request can be viewed by the Stakeholder such as status, priority; date approved and where in the system is the request.

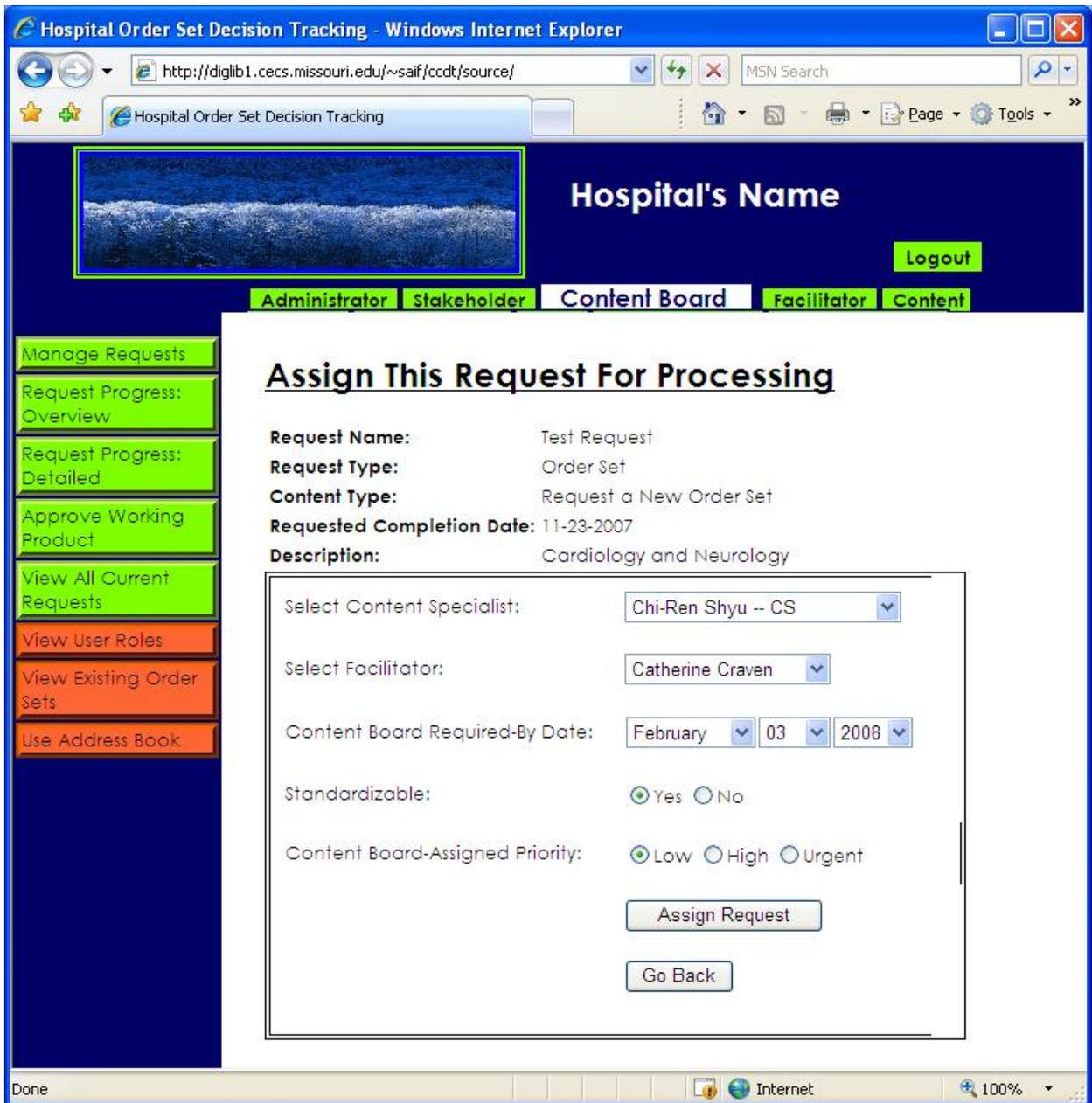


Figure 12 Content Board processing form for approved request. When the Board approves the request submitted, the board has to assign Content Specialist and Facilitator to the request as well as, a new required date and priority. The drop down menus is populated from the Database with names of users in each module.

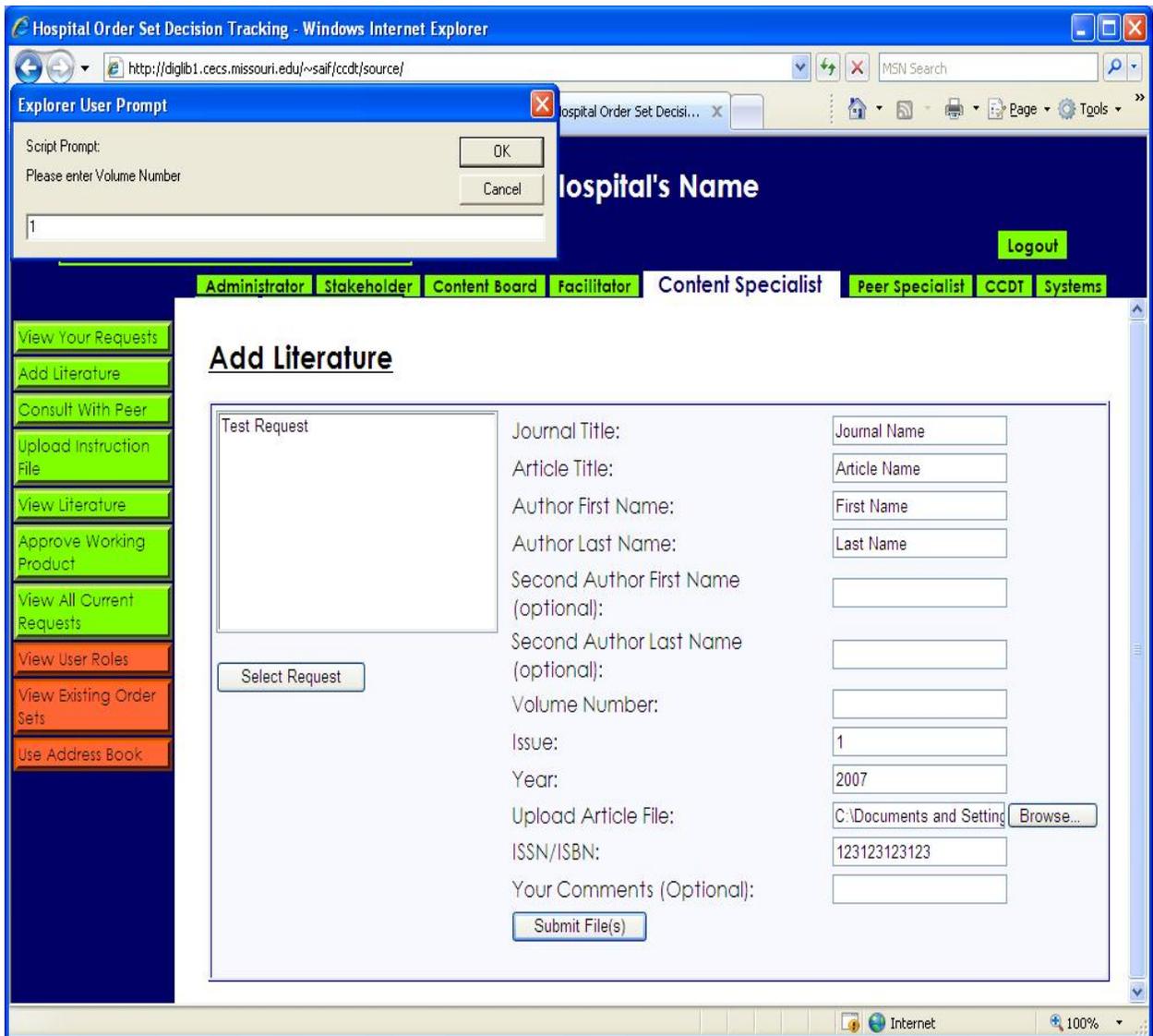


Figure 13 Upload Literature relevant for specific request.

For each request there has to be supporting literature in the system. Content Specialist has to fill a form and then upload at least one literature file. The form is validated through Jscript. IF a field is left empty or has incorrect information an input box is displayed for the user to correct that field.

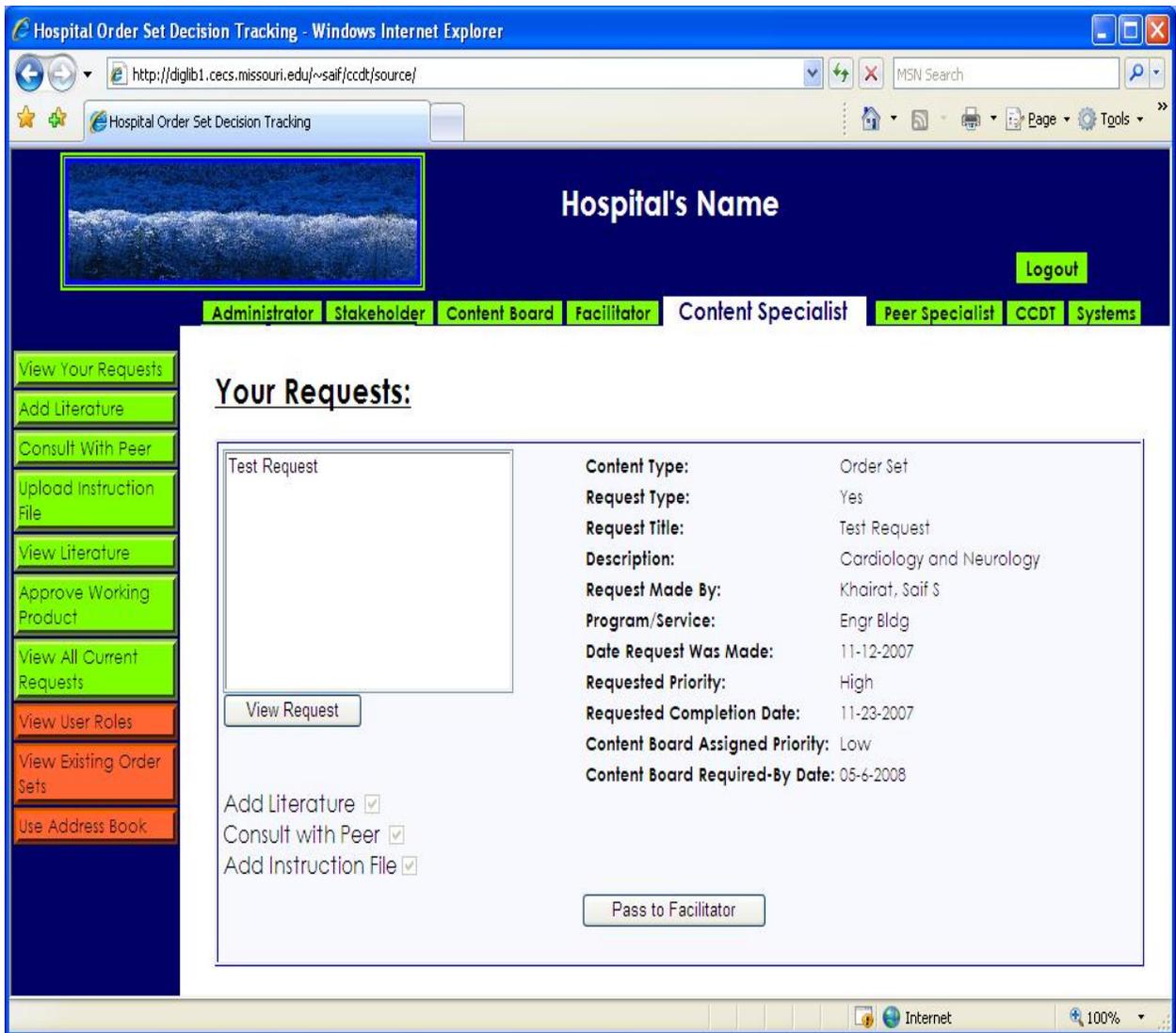


Figure 14 Pass the request to Facilitator

For a request to be passed from the Content specialist (CS) to the Facilitator the CS has to fulfill three criteria; upload literature, consult a peer and add instruction file. When all three tasks are done the CS can then pass the request and its attached documents to the Facilitator.

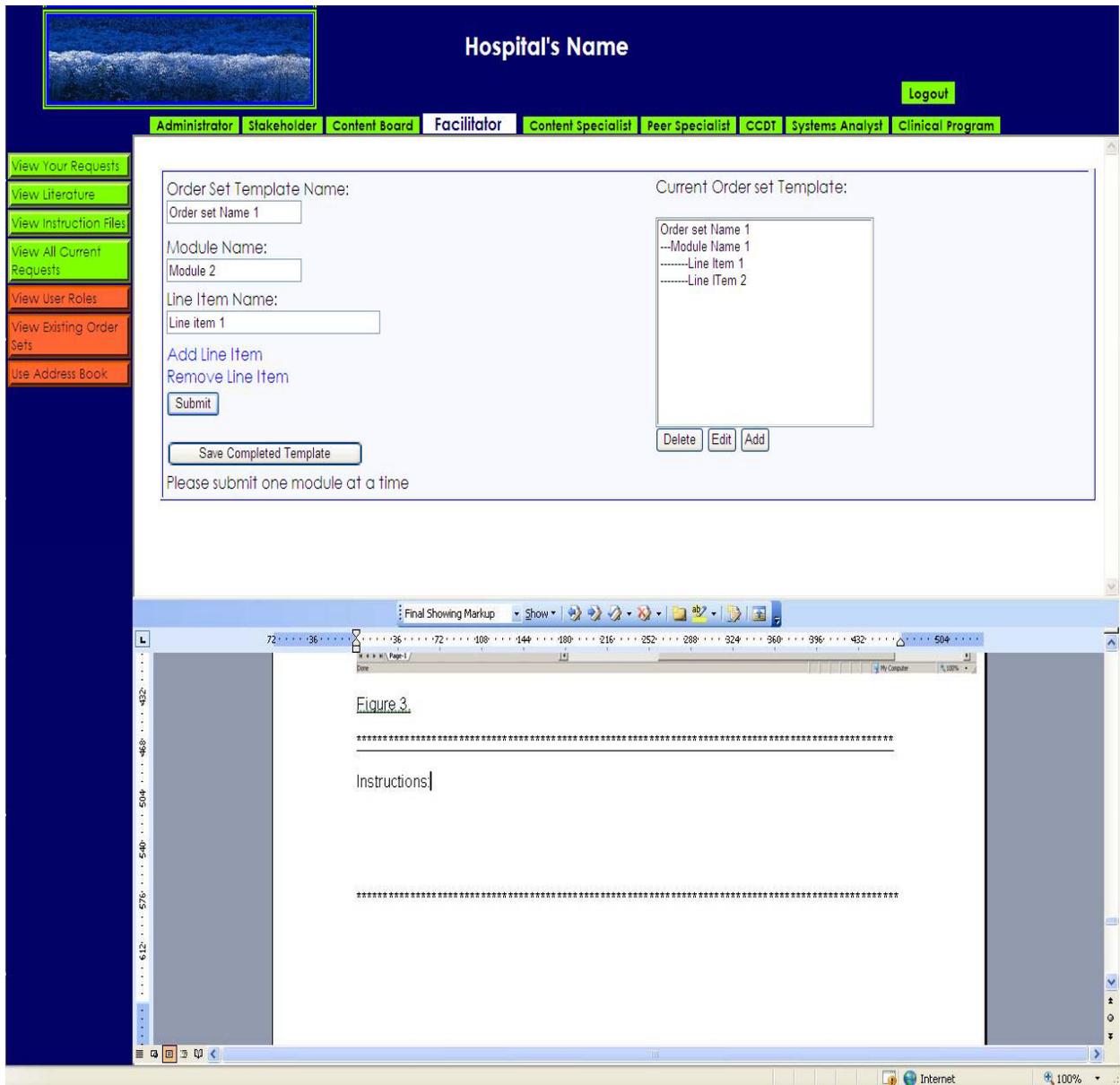


Figure 15 Make a template for a given Request using attached documents

The Facilitator creates a template for each approved request. In order to create this template, the Facilitator needs to view the instruction file sent by the CS. Therefore, the page is divided into two parts, the template creation form and the instruction file. In the template creation part, the Facilitator can view the current template and can add/delete/update any item in the template.

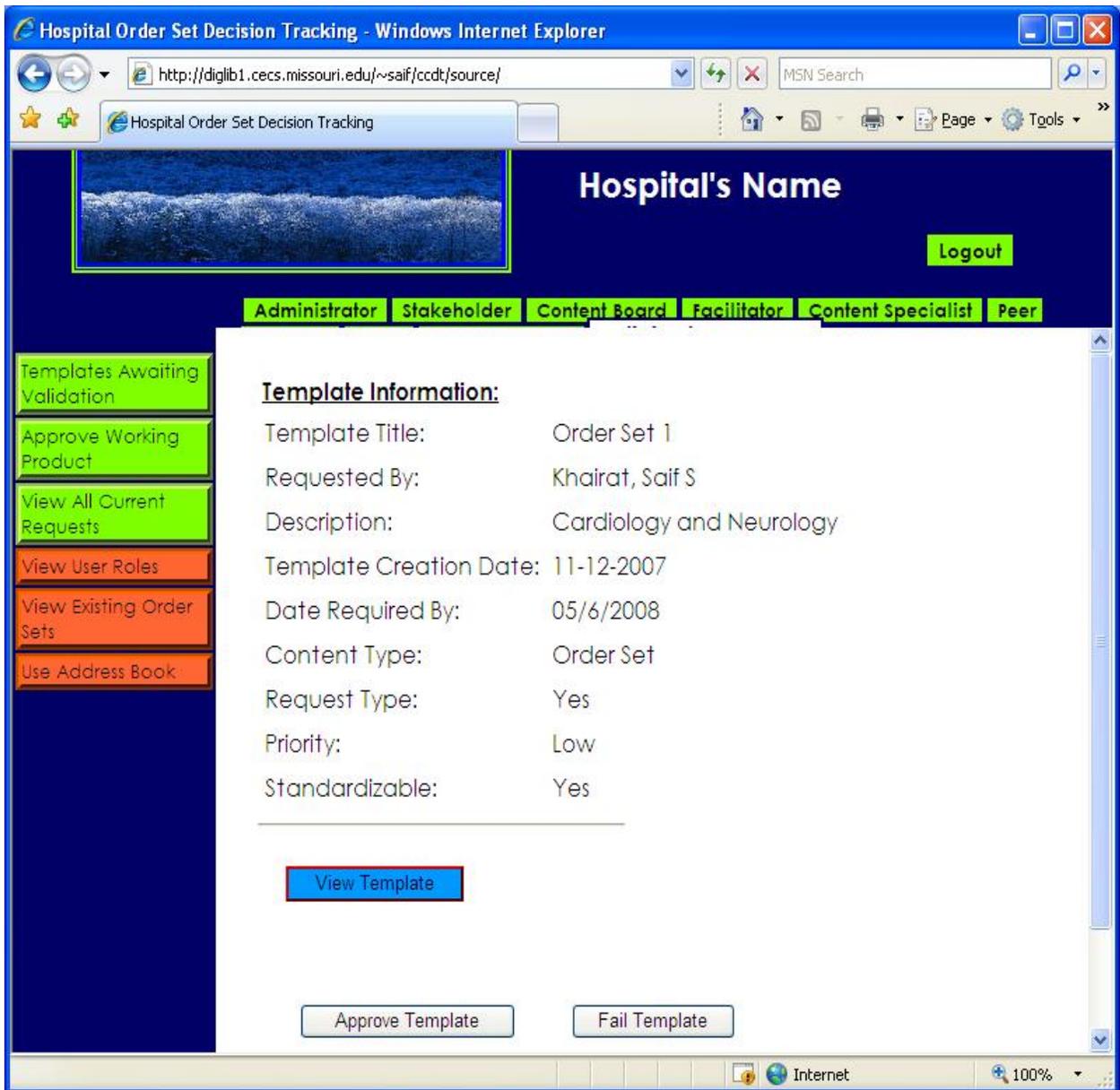


Figure 16 Approving Template for a request.

After the Template has been created, two different modules, Clinical Program Director and CCDT, have to approve of the template. Each module can view the template and then either accept or reject the template. If the template is rejected, the template will go back the CS for modifications.

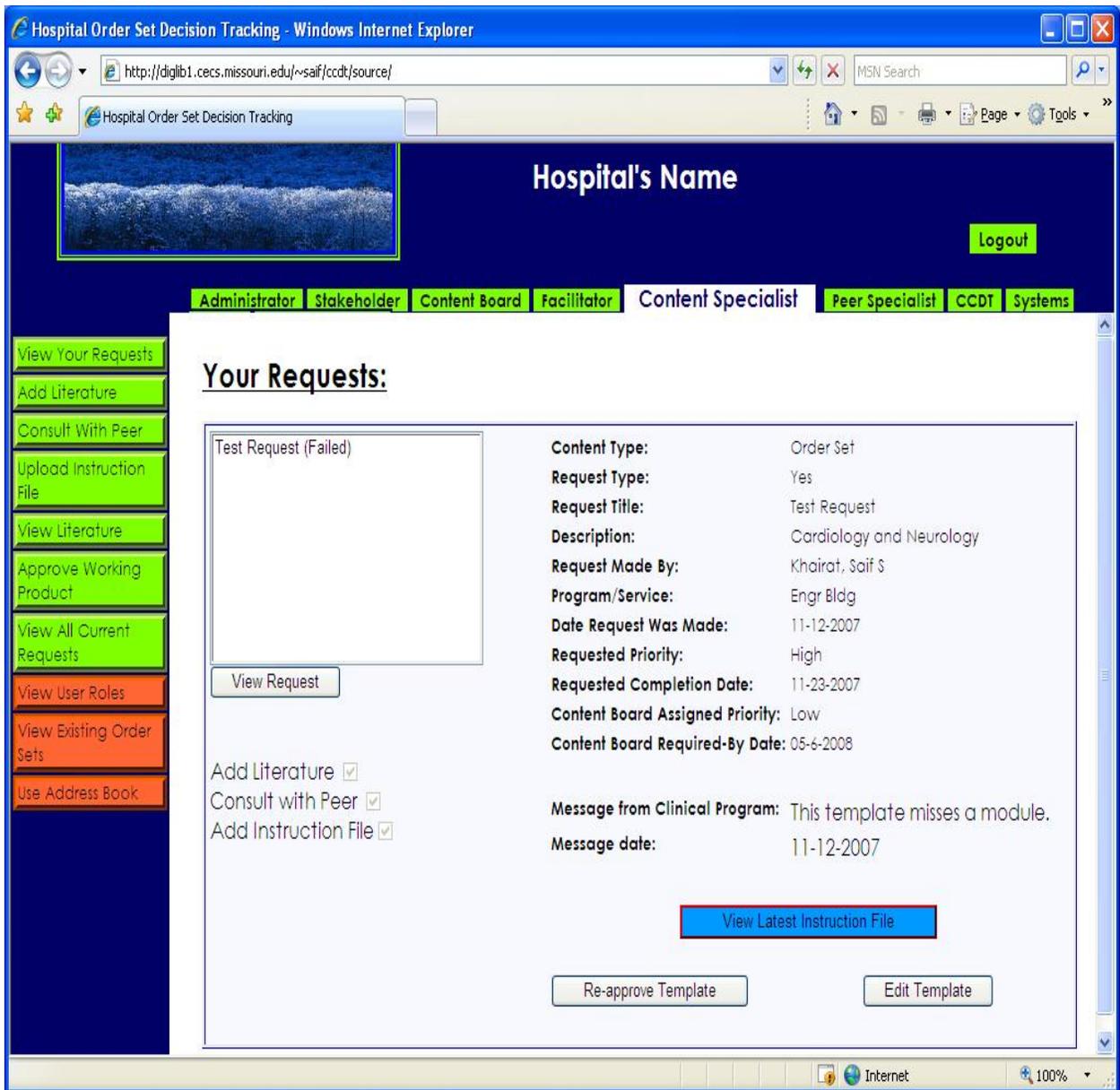


Figure 17 Rejected Templates at Content Specialist.

If a template is rejected it goes back to the CS. The CS sees all the information related to the Template, the Template itself and a reason for the rejection. The CS has two options either to re-approve the current Template without changes or edit the current Template and modify it. In both cases, the Template goes back to the Module that rejected it.



Figure 18 Convert template to working product

The System Analyst (SA) is the last user to sign off on the Template. The SA can view the final template and if they approve of it then it will be converted to a Working product in the system.

The screenshot shows a web browser window with the URL <http://diglib1.cecs.missouri.edu/~saif/ccdt/source/>. The page header includes a navigation menu with roles: Administrator, Stakeholder, Content Board (selected), Facilitator, Content Specialist, and Peer. A sidebar on the left contains menu items like 'Manage Requests', 'Request Progress: Overview', 'Request Progress: Detailed', 'Approve Working Product', 'View All Current Requests', 'View User Roles', 'View Existing Order Sets', and 'Use Address Book'. The main content area displays a table for 'Request Name: Test Request'.

Actor	Role	Action	Action Date	Completed
Saif Khairat -- Engr Bldg	Stakeholder	(1/12): Makes a Request	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Content Board	(2/12): Reviews and assigns or fails request	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Content Specialist	(3/12): Submits instructions or fails request	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Facilitator	(4/12): Creates template from instructions	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Content Specialist	(4/12): Upload Literature	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Content Specialist	(4/12): Upload Literature	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Content Specialist	(5/12): Checks with Peer Specialist	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Content Specialist	(5/12): Upload Instruction File	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Content Specialist	(5/12): Upload Instruction File	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Clinical Program	(6/12): Approves template or fails it	11/12/2007	Yes
Saif Khairat -- Engr Bldg	CCDT	(7/12): Approves template or fails it	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Systems Analyst	(8/12): Indicates template has been used to implement a working CPO in the CPOE system	00/00/0000	Yes
Saif Khairat -- Engr Bldg	Content Specialist	(9/12): Approves working product in CPOE system	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Content Board	(10/12): Approves working product in CPOE system	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Clinical Program	(11/12): Approves working product in CPOE system	11/12/2007	Yes
Saif Khairat -- Engr Bldg	Systems Analyst	(12/12): Indicates go-live in CPOE system	11/12/2007	Yes

Figure 19 Track a specific Request

The Content Board has the option of tracking any request. The track shows the action made in the request, the actor and their program service, date and whether it is completed or not.

Vita

Saif is an Egyptian who was born in Cairo, Egypt. He lived in Egypt and then moved to Kuwait at an early age where he did most of his pre-college education. After graduating from high school, Saif moved to the U.S.A to study Computer Science. In December 2005, he graduated with a Bachelor's degree in Computer Science with Honors from the *University of Missouri*. A month later, Saif started his Master's degree in Computer Science at the same school with Dr. Chi-Ren Shyu as his advisor.

It is worth mentioning that during Saif's stay in America, he heavily depended on his parents for moral, financial, and all other kinds of support. For which he is deeply grateful and thankful.

The next move for Saif is to work at a well-established company, where he can put the knowledge he acquired into practice. On the long run, Saif plans to go back to Egypt to help his parents, family and friends in particular and his country at large.

Finally, a quote that has helped Saif get through the difficult times during his study in specific and life in general.

“So, verily, with every difficulty, there is relief” (Quran, 094.005)