DEVELOPMENT OF AN ADMINISTRATRION MODULE FOR THE ALERT AND REMINDER PACKAGE OF THE NURSING DOCUMENTATION SYSTEM HPAD

by

Miguel Angel Nieves Acevedo

A project submitted in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING
in
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2009

Approved by:

_____	_____
José A. Borges, Ph.D. President, Graduate Committee	Date
_____	_____
Néstor J. Rodríguez, Ph.D. Member, Graduate Committee	Date
_____	_____
José Navarro, MCpE. Member, Graduate Committee	Date
_____	_____
Miriam J. Nieto, Ph.D. Representative of Graduate Studies	Date
_____	_____
Jaime Seguel, Ph.D. Associate Dean for Academic Affairs College of Engineering and Acting Chairman of the Department	Date

ABSTRACT

This document describes the development of a system to extend the functionality of the Alert and Reminder Module (ARM) from the nursing documentation system HPAD. The system introduces features to adapt the ARM to any hospital needs. It provides mechanisms to add, edit and delete the reminder categories of the ARM. In addition, the system offers several customization features. The development of the system required the redesign of the ARM's source code, which was made using the Façade, Strategy, DAO, and the Simple Factory Design Patterns. Also, the tooltip of the ARM was redesigned with the goal to improve the displayed information and the mechanism to attend the reminders. A heuristic evaluation was conducted on the system which helped identify usability problems and make the respective corrections in order to improve the system's usability level.

RESUMEN

A través de esta documentación se describe el desarrollo de un sistema cuya función es la extensión de la funcionalidad del Módulo de Alertas y Recordatorios (ARM) del Sistema de Documentación de Enfermería (HPAD). El sistema introduce elementos que permiten que el ARM se pueda adaptar a diversas necesidades en los hospitales. De igual forma, provee mecanismos para añadir, editar y eliminar las categorías de recordatorios en el ARM. En adición, el sistema ofrece varios elementos para su personalización. El desarrollo de éste requirió el rediseño del código del ARM. El ARM se realizó utilizando los patrones de diseño *"*Façade*"*, *"*Strategy*"*, *"*DAO*"* y *"*Simple Factory*"*. Además, el *"*tooltip*"* del ARM fue rediseñado con el propósito de mejorar la información ilustrada y el mecanismo para atender los recordatorios. Finalmente, se realizó una evaluación heurística, la cual ayudó a identificar y corregir problemas de usabilidad, principalmente para el mejoramiento del nivel de usabilidad del sistema.

To My Family:

Mom, Dad,

Sisi, Tata, Nayi, Nata,

And My Love Yesy

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor and graduate committee president, Dr. José Borges for his guidance, effort and dedication to this project. His words and lessons throughout my master's degree helped me become a better student and professional. Thanks for being an outstanding teacher and at the same time an excellent human being.

I want to thank my graduate committee members Dr. Néstor Rodríguez for sharing his expertise and for his contributions to this project; Dra. Miriam Nieto for her time and contribution to this project, and Prof. José Navarro who has been a great inspiration in my professional life. Also, I would like to show appreciation to Dra. Celia R. Colón Rivera for her time, patience and for educating me on the health care area.

Also, I would like to thank my family for their support and love. Dad, thanks for being such a great father and role model, without your guidance I would not be here today. Mom, thanks for your love and all the things you have made for me. Sisters, thanks for your love and friendship, you know that I always be there for you. And the newest members of the family; Nayelis and Natasha, thanks for the love and happiness you girls bring to the family.

Finally, but no least I want to give thanks to my best friend and fiancée, Miss Yesenia González Ortiz, who has been my biggest support throughout my master's degree and has given me the strength to keep fighting throughout obstacles in life. Thanks for being such an exceptional person; you are the main reason for my accomplishments.

TABLE OF CONTENTS

TABLE LIST

FIGURE LIST

LIST OF ABBREVIATIONS

PBD	Paper-based Documentation
EHS	Electronic Health Systems
ARM	Alert and Reminder Module
ASM	Administrator and Statistical Module
DBMS	Database Management System
CCR	Computerized Clinical Reminders
UML	Unified Modeling Language

CHAPTER 1 – INTRODUCTION

1.1 Justification

Thanks to the advantages of technology many hospitals have been switching from paper-based documentation (PBD) to Electronic Health Systems (EHS) [Weimar09]. Some of the advantages that EHS offer as compared to the paper-based systems are:

- Fast information retrieval: EHS can find and access patient's medical records in seconds. This can be an extensive task in PBD systems if a large amount of patient records exists.

- Secure storage of patient data: Information in EHS can be encrypted making almost impossible to compromise patient data if the encryption key is not known. This is a dangerous aspect of the PBD if the patient records are not stored in a secure physical location.

- Promotion of guidelines: The development of well defined computer systems can help hospitals' employee work under certain guidelines or hospital defined procedures.

- Increase of service quality: An EHS can facilitate physicians and nurses medical literature. Also, it can advise and validate nurses' drugs administration to patients. Moreover, EHS can generate alerts based on abnormal result from patients'

laboratory results and exams. Thus, these systems can assist nurses in the decision making process which can help increase their quality of service.

- Integration: With a central database server the integration of different departments, hospitals, and agencies can be possible. Making patient data, diseases history, laboratory results and other information between these sources accessible at any time.

- Reports and statistics generation: Due to the fact that most of the EHS use databases to store their medical information, reports and statistics can be easily generated from these database records. With these reports and statistics hospitals can analyze their quality of service and identify areas for improvements.

Even with all the mentioned advantages, less than 10% of the hospitals in the United States have implemented health information technologies [Mackinnon09]. One of the main reasons for this is that physicians and nurses find available EHS frustrating due to its poor usability [Weimar09]. Another of the main reasons for this is the lack of flexibility. Due to the complexity of these systems, the diversity and variability of hospital protocols and processes, and to the constant changes of hospital procedures it's very difficult to implement an EHS that fits all the requirements of different hospitals.

Thus, for an EHS to have more acceptance and success the developers of this kind of software have to consider all the possible changes that may arise in the development or application's life cycle. These changes occur independently of the type of EHS they are working with, it can be a small, primary care physician or enterprise EHS. The best way to deal with these changes is to anticipate what may vary and making sure it can be changed without having to redesign. This is very important because most of the times, when addressing a change, the creators or designers of the software are not available to make the changes, so the responsibility is assigned to a new designer. This implies learning the software's logic and implementation, which could take more time than incorporating the changes. Obviously, the benefits of addressing changes on applications without having to redesign could mean a lot of money to a company.

All the above arguments were critical in the decision to redesign and extend the Alerts and Reminders Module (ARM) from the Nursing Documentation System HPAD [Perez05]. ARM is a subsystem of an Electronic Medical Record and was designed to provide reminders and alerts of patient's pending services or tasks to nurses and physicians. Originally, this module was created by Carlos Perez as part of his master's degree work in the University of Puerto Rico at Mayaguez in 2005. ARM is a visual component that consists of a table with a row for all the patients in a ward and in which the columns are identified by icons that represent the reminders and alerts, called aspects. Currently, the observed aspects are: diagnostic studies, draft notes, allergies, physician orders, medicine orders, drip orders, consultation orders, and general alerts (a detailed System description can be found in Chapter 3).

The main problem with ARM is the lack of adaptively to any particular hospital environment without the need of changing the source code and recompiling it. This is a critical problem because every hospital has their own metrics to evaluate patients' aspects status and what aspects to monitor. This means that ARM needs a way to change the tolerance of aspects and edit the aspects to monitor without the need of modifying the source code and even without a programmer.

The ARM provides many advantages to hospitals. The system offers a way to improve patients' health care by helping nurses in their workflow (meaning better nurses' performance). The system can advise the nurses when to attend a prescribed physician order and provides mechanisms to alert nurses of abnormal patients' conditions. For example, the system can detect if a patient is allergic to a drug on an order made by a physician, preventing an allergic or even a deadly reaction to the patient. ARM can also be used as an administrative tool by providing a report of the average turnaround times for ordered services or interventions, and a summary of the generated alerts.

1.2 Objective

The primary objective of this project is to develop an Administrator and Statistical Module for the Alerts and Reminders Module (ARM) from the nurse documentation system HPAD. This system will allow hospital system administrators to adjust the Alerts and Reminders system to their respective requirements. The module will provide full customization, making the system unique to each hospital. It will allow hospitals to use their badge or insignia

as the application logo and it will allow the specification and reordering of alerts and reminders. The system will also provide tools to generate reports of the patient alerts and reminders history.

In addition to the primary objectives, there were also several secondary objectives established for this project. The following list enumerates these objectives:

- Improve the information displayed by hints or tooltips.
- Migrate the HPAD database to Microsoft SQL Server 2008.
- Redesign the ARM implementation making use of design patterns, to make future changes easier.
- Redesign the deficiencies, problems, and errors encountered during the analysis of the original ARM.

The objectives of this project were realized using the following methodology:

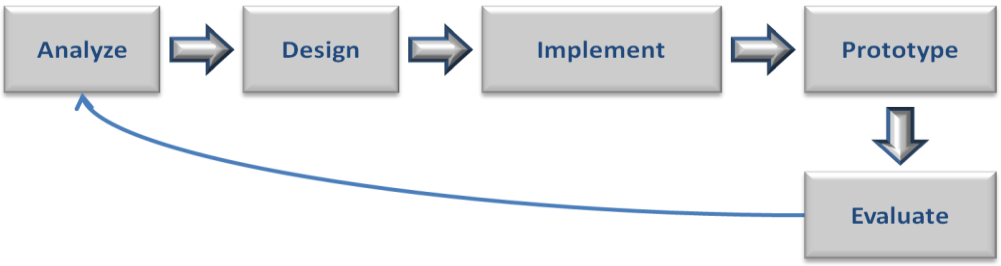

Figure 1.1 Project Methodology

A comprehensive analysis of the existing ARM was conducted in the first phase. The needed functionality for the new Administrator and Statistical module was studied during this analysis phase. In the design phase, information related to patients' reminders and their respective categories was gathered. In addition, the development of user interfaces was started in this phase. The third phase consisted of the implementation of the final prototype based on the

compiled information from previous phases. This phase included the enhancement of the original ARM source code, the interfaces layout, coding the complete functionality, and the modification and creation of the database tables. The final phase was a heuristic evaluation of the final prototype from a group of evaluators with the intention to identify usability problems. Finally, the evaluation of these usability problems was carried out and the necessary corrections were realized.

1.3 Project Outline

The next chapter presents a literature review of previous work related to Alerts and Reminder Systems and the project background. Chapter 3 provides a detailed description of the original Alert and Reminder Module from HPAD including its deficiencies. The description of the new Administrator and Statistical Module for the ARM is presented in Chapter 4. Various upgrades and enhancement made to the ARM are presented in Chapter 5. The result of a heuristic evaluation of the system is presented in Chapter 6. Finally, the conclusions and future work are presented in Chapter 7.

CHAPTER 2 – PREVIOUS WORKS

2.1 Introduction

The following studies represent the most relevant research related to this project. The primary focus will be on studies related to Alerts and Reminder Systems.

2.2 Alerts and Reminders Systems

An observational study of nurses and providers interacting with Computerized Clinical Reminders (CCR) was conducted by Jason J. Saleem and his colleagues [Saleem05]. They observed 35 nurses and 55 physicians using a CCR module from an electronic medical record system incorporated in the Veterans Administration Medical Centers all over the United States. The primary goal of their study was to find the barriers and facilitators of using a CCR system.

The results of their observation show that main barriers of using a CCR were:

- Coordination between nurses and provider (role confusion)
- Attending reminders while patients have been discharged or taken away from nurses/providers care.
- Perception of clinical reminders as a non-core work activity (i.e. not enough time to take care of reminders or paper-based workarounds).
- Lack of flexibility (no available option to satisfy some reminders).
- Poor usability (poor interfaces design and lack of computer performance).

On the other hand, the observed factors for making easier the use of a CCR were:

- Limiting the number of reminders (this can help nurses/providers reminder fatigue).
- Strategic location of computer workstations (a proper computer location/positioning can help the nurses/providers use the CCR while keeping eye contact with patients).
- Integration of reminders into workflow and the ability to report system problems and receive a prompt feedback.

In another study conducted by Martin Reitstätter and his colleagues [Reitstätter06], a knowledge base to classify patients' results from a biochemical test of thyroid hormones was developed. They measured serum levels of thyroid stimulating hormone (TSH), total thyroxin (TT4), triiodothyronine (TT3), thyroxin-binding globulin (TBG), and thyrotrophic-releasing hormone (TRH)-stimulated TSH. These measurements were classified as normal, increased, and decreased and depending on the combination of the results and on the patient's diagnoses, they were classified as obligatory, possible, or excluded.

They developed a CCR system to monitor diagnosis inputs made by physicians and nurses at the Department of the Division of Endocrinology and Metabolism of the Vienna General Hospital, using the knowledge base as the business rule. They concluded that with this CCR System the diagnosis inputs could be evaluated to see if the physician/nurse had made the right decision to assign it or if it had been an unjustified referral due to misinterpretation of laboratory results.

In a similar study conducted by Hasman, A. and his colleagues [Hasman00], they characterized CCR Systems as passive (systems with information that has to be actively retrieved

by the health professional) and active (systems that make decisions based in a knowledge base and that can advise the heath professional). In this study, the CCR system was based in clinical protocols or guidelines. Hasman and his colleagues concluded that physicians do not accept all types of decision supports. They indicated that systems with diminished user role should be avoided and that decisions support systems should allow the health professional to work in their own way and should only intervene when the decisions are wrong. For them, CCR systems are more promising because they leave the physician in control and only provide feedback when the physician is wrong.

In a referenced study by Hasman, a CCR to promote practices guidelines of the intensive care unit of the Catharina Hospital in Eindhoven was developed by De Clercq [DeClercq99]. The guidelines were represented as objects that encapsulated a production rule (using the syntax "IF expression THEN GiveReminder") and had an explanation of the guideline and a message that would be given by the RS when the guideline was not followed. A total of 58 different guidelines were entered into the system. This system was tested with a data set of 803 patients and resulted that 88% of the generated reminders were classified as correct. In the other hand, the reminders classified as false alarms were generated by guidelines too generic and were easily corrected.

A study to identify patterns of use of computerized clinical reminders (CCR) across an integrated healthcare system and to describe institutional factors associated with their implementation was conducted by Constance H. Fung and his colleagues [Fung04]. They surveyed 261 participants from 104 Veterans Health Administration (VHA) health care facilities regarding the number and types of CCR available at each facility. The results showed that the

number of categories or conditions at different facilities ranged from 1 to 15. The most commonly implemented CRR conditions (used by 85% of the facilities) were VHA national performance measures (e.g. tobacco cessation, immunizations and diabetes mellitus) and the least commonly implemented CCR conditions were post-deployment health evaluation and management, medically unexplained symptoms, and erectile dysfunction.

Even thought VHA facilities are part of the same system and has common information technology infrastructure, Constance H. Fung and his colleagues concluded that the VHA facilities had variations in the implemented CCR conditions. They found that a possibility for these variations is that facilities with higher perceived utility and ease of use of clinical reminders are more likely to implement a larger number of CCR conditions.

2.3 Project Background

The concept of the original ARM was based in a study conducted by José A. Borges and his colleagues [Borges97]. They proposed an automatic system for auto-supervision named SAAS (*Sistema Automático de Auto Supervisión*). SAAS was intended to be used in the Emergency Room (ER) of local hospitals in Puerto Rico and its main objective was to keep track and supervise of all the processes in the ER, having the patients as the major focus.

SAAS proposed keeping track or monitoring the patients' physical location, pending services, waiting time, and other information related to conditions, treatments, and necessary resources. To achieve the mentioned task, SAAS proposed the use of several equipments, such as:

- Strategically placed sensors in different areas of the ER to monitor patients flow.
- Magnetic ID card (for every patient), to facilitate access to patient's records.

- Magnetic ID bracelet, to register the patient at the different areas of services.
- Bar code labels, to achieve items identification provided to patients (e.g. labs, drugs to be administered and materials supplied to the patient).

The proposed system had the potential of improving the quality of service and workflow in the ER.

Given the fact that this project is based in a previous work done by Carlos Perez [Perez05], it is important to illustrate his contribution. The principal objective of his work was to design a prototype of an Alert and Reminder Module (ARM) for a nurse documentation system. The developed prototype gives nurses and physician an efficient way to navigate between patient's electronic medical records, and most importantly, it constantly supervises patient's orders and services status to generate reminders and alerts. The overall description of the prototype is not discussed, because it will be described with further details on following chapters.

As part of the work by Carlos Pérez, a usability study was conducted on the prototype, and the results of this study revealed that age was a key factor on the participants' task completion time. It took more time for older nurses to perform tasks compared to younger nurses.

The previous works discussed in this chapter contributed to acquire the needed knowledge for the present work. Their information eased the understanding to implement Alert and Reminder Systems in Electronic Health Systems.

CHAPTER 3 - SYSTEM DESCRIPTION

3.1 Introduction

The first part of this chapter describes the original ARM system from the point of view of user interfaces. The ARM is part of a nursing documentation system, called HPAD, which manages hospital patients' information. ARM was developed by Carlos Perez [Perez05] as part of his graduate studies in the University of Puerto Rico in Mayaguez and the main purpose of the system was to generate alerts and reminders of physician orders existing in the database records of HPAD.

Originally this system was developed using JAVA 1.4.2 [JAVA] and open source community database server MySQL 4.1.12 [MySQL]. After many studies and graduate students research, HPAD and ARM have gone through a lot of changes. The latest work was done by Carlos Duarte [Duarte08], in which he made a re-engineering of the application's source code implementation. His work contributed to a great improvement to the application's source code and overall performance. The tools used for the re-engineering process were JAVA 1.5 [JAVA] and Microsoft's SQL Server 2000 [SQLServer].

In the final stages of this chapter a list of the deficiencies and limitations of the original version of ARM are described.

3.2 Login Interface

The Login interface shown in Figure 3.1 is the first window a user sees when the application starts. This interface serves to validate the authenticity of the user, thus preventing unauthorized use of the system and making possible the ability to track changes in the system.

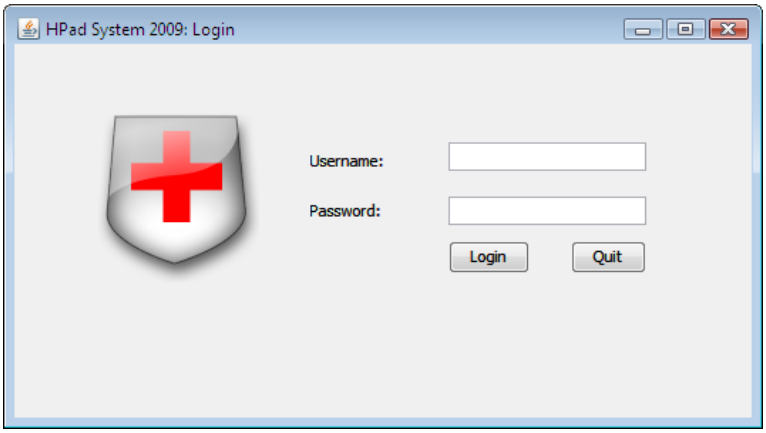

Figure 3.1 Login Interface

3.3 Patient List Interface

The Patient List Interface shown in Figure 3.2 is the main window of ARM. This interface is divided in two sections; the header section where the HPAD logo, current logged user information (Name, and License Number), a combo box with the hospitals' working areas and a button to log out from the ARM are displayed; and the main section that consists of a table in which the rows include the names of all the patients in a ward and in which the columns are identified by icons that represent the reminders and alerts, called aspects. The table's columns represent a reminder category and are identified by particular icons. For example, a reminder category for medications uses a pill image as icon. These reminder categories consist of different

related physician order types (See Table A-1 for a detailed classification of the ARM physician orders types).

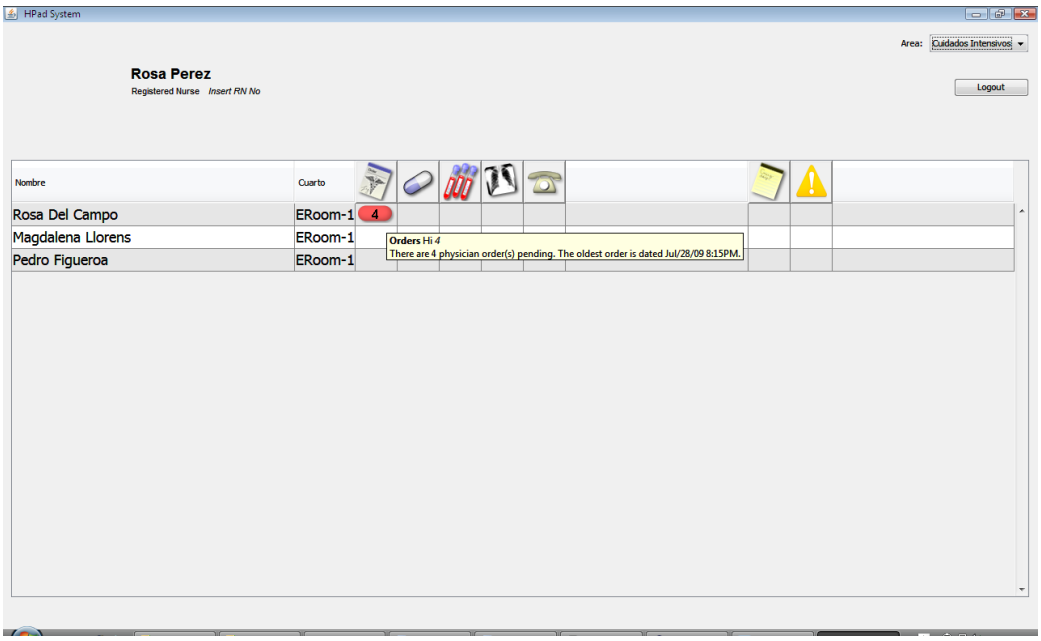

Figure 3.2 Patient List Interface

ARM indicates that there is an active reminder or alert by placing a small icon on the row that identifies the patient and in the column corresponding to the pending service. A small size green icon indicates a pending service or alert with low urgency level. A medium size yellow icon indicates a pending service or alert with medium urgency level and the large size red icon indicates a pending service or alert with high urgency level. The amount of pending services or alerts for a given aspect is also indicated in the center of every icon. Thus, if a patient X has 5 pending services of type Y and the highest urgency level is medium, the column representing the service Y from patient X will have a medium size yellow icon with a number 5 in the center.

The ARM determines the urgency level of an alert or reminder based an urgency calculation strategy (See Table 4.1) and on the thresholds values of every particular physician order type. For example, the urgency level of the discontinue order type is calculated based on the elapsed time since the order was created, thus, if the elapsed time has exceeded the low limit threshold the ARM will generate a low level reminder which will be represented by the small size green icon.

Another useful property of this table layout is the use of tooltips to provide user fast information access. When the mouse is placed over a reminder icon a tooltip with the details of the reminder is displayed. The tooltip (Figure 5.3) displays the details (order title, level, order date, current date and elapsed time) of all the active reminders of a particular reminder category.

Finally, when a user desires to manage or respond to an alert or reminder they need to click on the desired reminder. This action will open an HPAD interface associated to the pertinent reminder which will allow the nurse to attend the selected alert or reminder. Consequently, the icon representing the reminder is removed from the ARM once it has been attended.

3.3 HPAD Interfaces

As previously stated, all the reminders and alerts generated in the ARM are originated by the physician orders stored in HPAD's database records. Thus, it is very important to describe the interfaces from HPAD, which are the ones used to attend the alerts and reminders.

These interfaces are divided into two sections, the header which contains the patient's personal information (name, gender, weight, height, room, primary physician, account number,

record number and health care plan name) and the main section that its composed by different tabs (Orders, Meds, Drips, Vitals, Notes, Assessment, Intake/Output and Pain) that the user can use to manage the patient's physician orders and clinical information.

The patients' actives alerts and reminders are also displayed as multi-layers icons in the upper right corner of these interfaces. The icons represent the same alerts and reminders from the Patient List Interface previously described with the addition of the patients allergies. The icons' background color represents the urgency level of the reminders categories (green for low, yellow for medium and red for high urgency level). Also, the icons use a badge with a number to indicate the total number of active reminders in each category. These icons are removed from the interfaces once they have been attended. The following figure is an example of these icons.

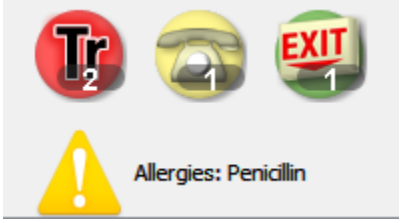

Figure 3.3 Multi-Layer Icons

3.3.1 Order Interface

The Order interface (Figure 3.4) is used to manage all the physician orders created for the patient. This interface is divided in two sections, the left section where the date, type, physician, and order status is displayed in a table, and the right section that shows the details of the selected order.

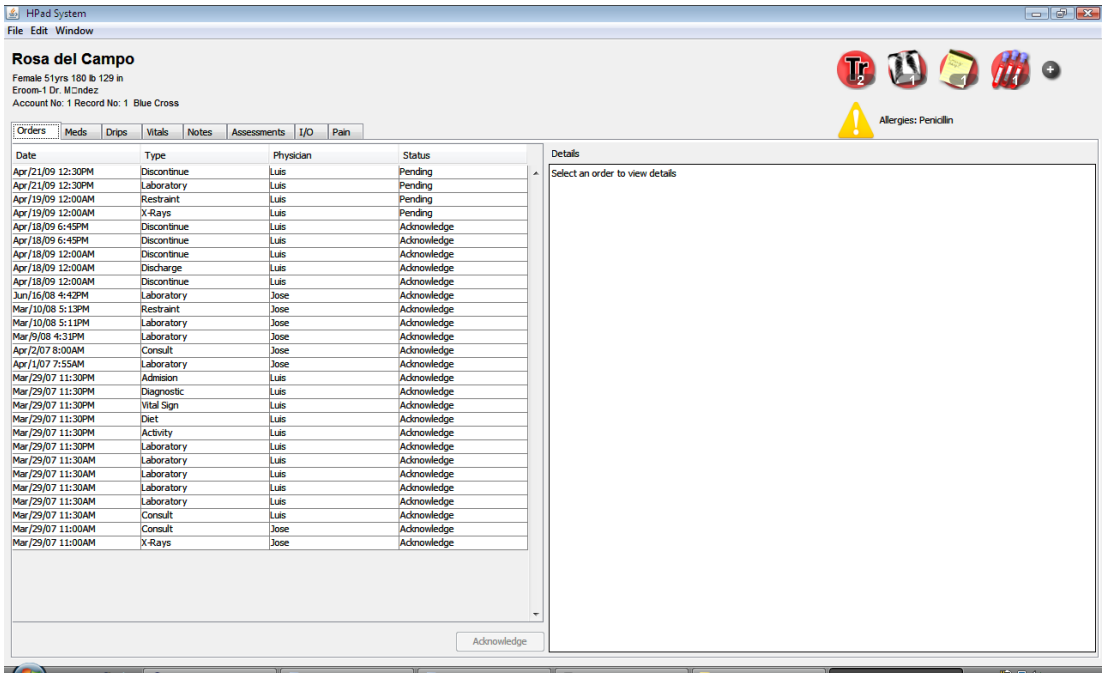

Figure 3.4 Order Interface

When the user clicks on the Acknowledge button an Acknowledge Order window (Figure 3.5) opens with the order information. Prior to acknowledging the order the user can add any additional comments of procedures taken to accomplish the order. Once the user clicks on the save button, the status of the order changes from "Pending" to "Acknowledge" in the Order Interface.

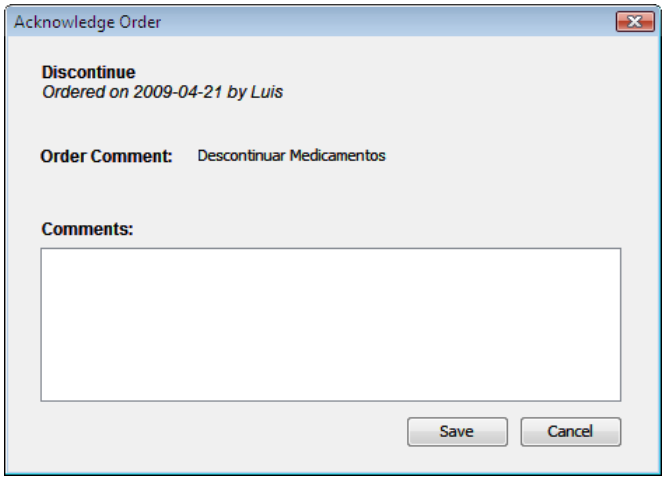

Figure 3.5 Acknowledge Order Interface

3.3.2 Meds and Drips Interfaces

The Meds and Drips interfaces (Figure 3.6) have an identical design and functionality. The purpose of having the two of them is to differentiate between medication given as drips and any other medication order. These interfaces are used to manage the patient's pending medications. The left section of these interfaces has a table with the medications, dose and status, and the right section show the details of the selected medication order.

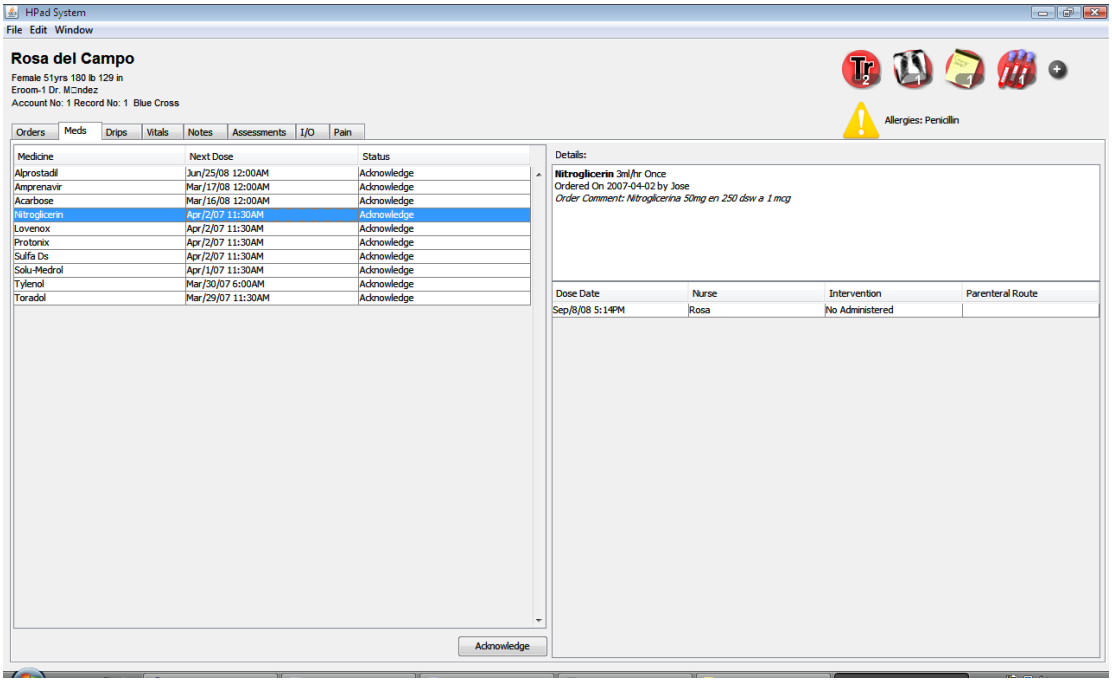

Figure 3.6 Meds Interface

When the users clicks on the Acknowledge button an Acknowledge Medication window (Figure 3.7) opens with the medication order details. Before the user can save the order he needs to select the intervention type which can be Administered (in this case the parental route and amount of the medication needs to be entered) or No Administered (in this case a reason why the medication was not issued needs to be documented).

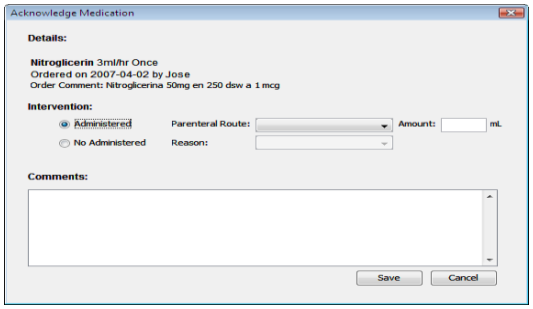

Figure 3.7 Acknowledge Medication Interface

3.3.3 Vitals Interface

The Vitals interface (Figure 3.8) is used to document the patient's vital signs. This interface displays the previously measured vital signs in a table with the date, temperature, blood pressure, pulse, respiration, O2 Saturation and the nurse name. Also, a graphical representation (chart) of the vital signs is displayed in the right section of the interface.

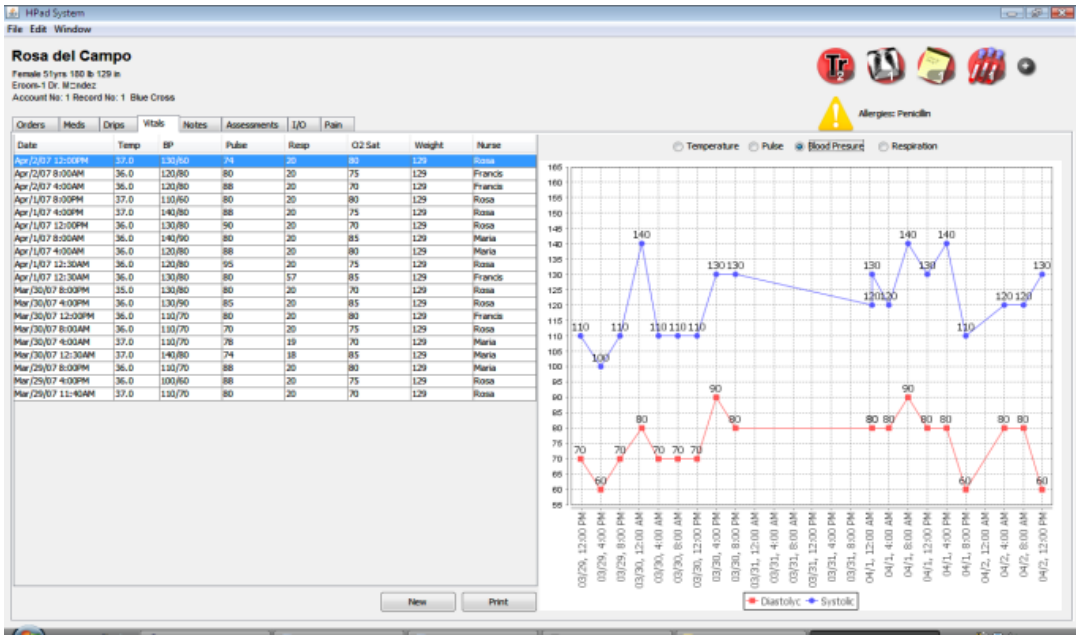

Figure 3.8 Vitals Interface

If the user clicks on the New button a Vitals Measurement window (Figure 3.9) opens. This window allows the user to enter all the patient vital signs measurement, once all the values all entered the user can click on the Save button and the values will appear on the Vitals interface table.

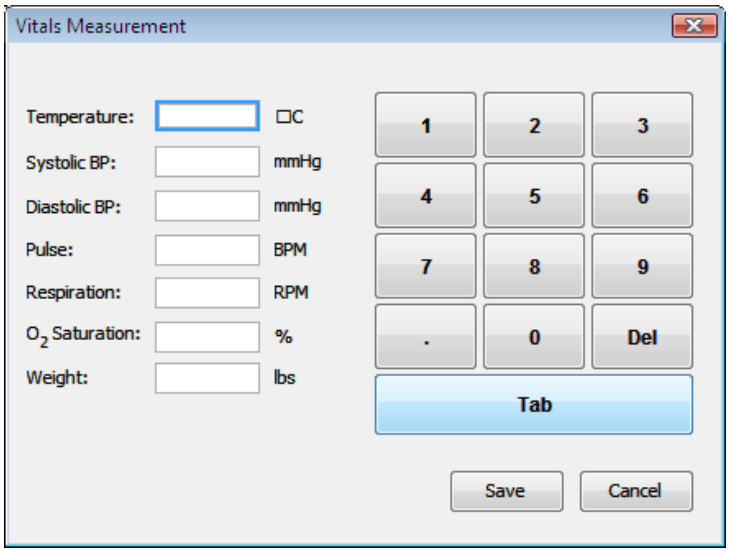

Figure 3.9 Vitals Measurement Interface

3.3.4 Notes Interface

The Notes interface (Figure 3.10) is used view to notes entered regarding the patient clinical information. This interface has a table with the date, focus, author, and type of the patient's notes in the left section and the details of the selected note are displayed in the right section of the interface. Also, this interface permits the user to filter the displayed notes by selecting the note type in a combo box.

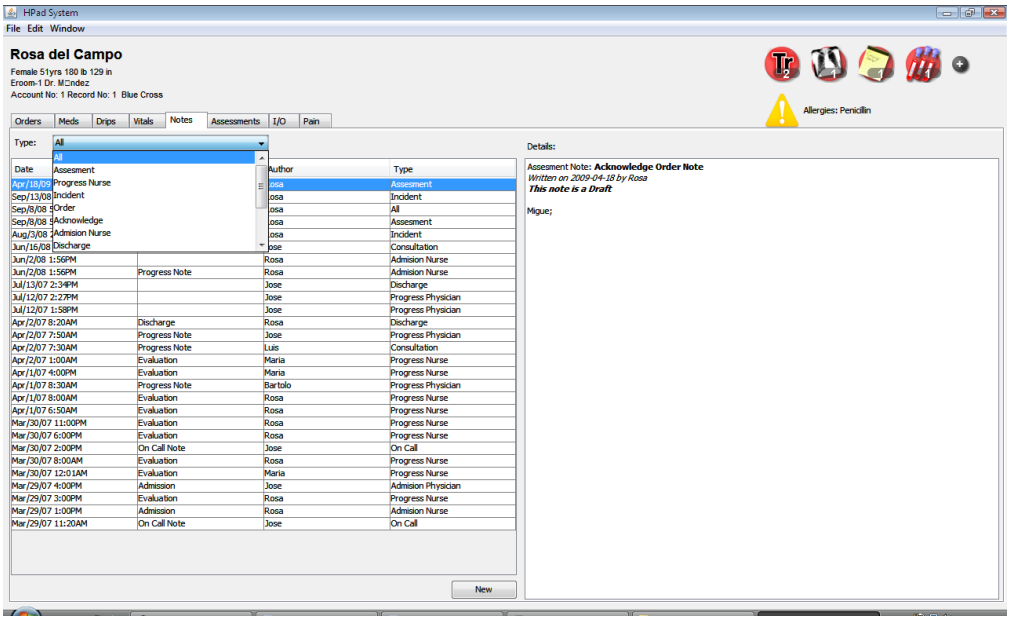

Figure 3.10 Notes Interface

If the user clicks on the New button a New Note window (Figure 3.11) opens. This interface allows the creation of new notes.

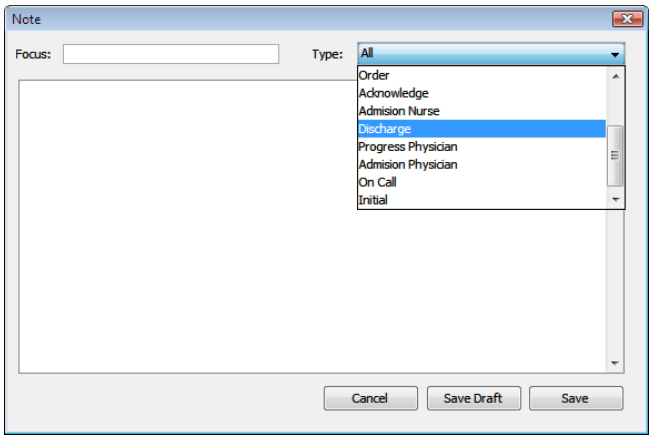

Figure 3.11 New Note Interface

3.3.5 Assessment Interface

The Assessment interface (Figure 3.12) is used to manage the patient assessments. This interface is divided in two sections. The left section which has a table with the assessments made to the patient categorized by date and nurse, and the right section displays the details of the selected assessment.

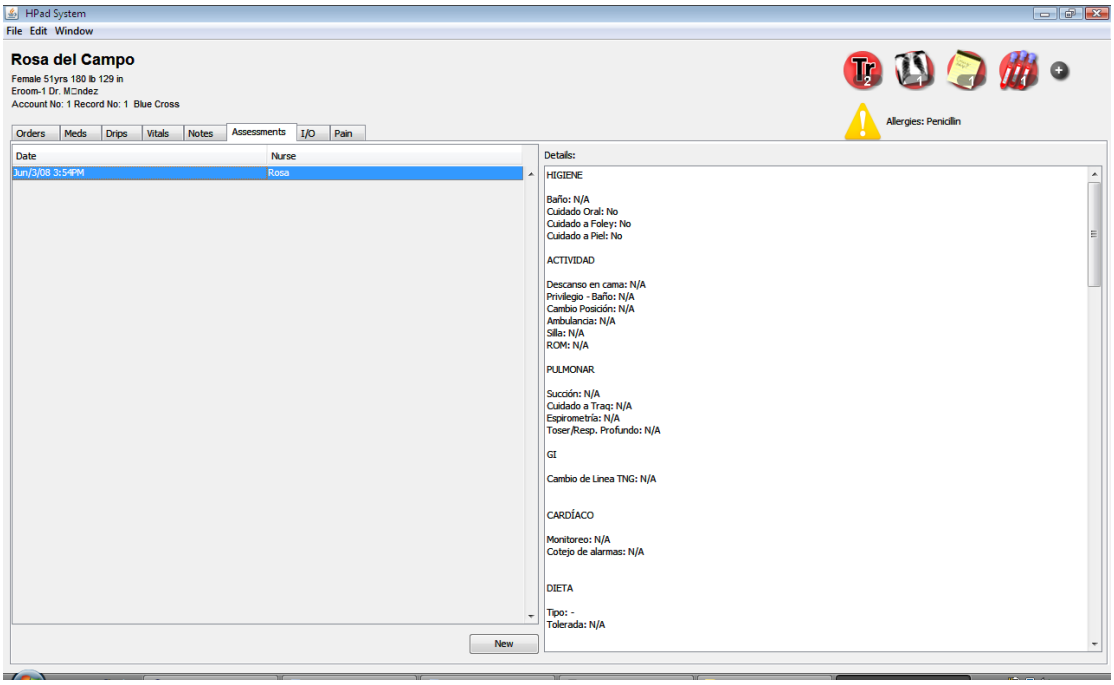

Figure 3.12 Assessment Interface

To create a new Assessment the users need to click on the New button that will open a New Assessment (Figure 3.13) window. After the users select all the necessary values they can click on the Save button and the new assessment will appear in the Assessment interface.

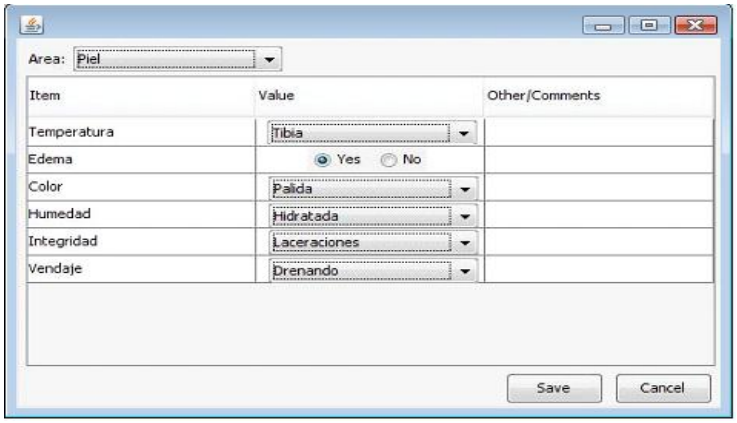

Figure 3.13 New Assessment Interface

3.3.6 Intake/Output Interface

The Intake/Output interface (Figure 3.14) is used to document the measurements taken for patient intake and output. This interface has a table with the measurements date, type, flow, milliliters and nurse name. This interface allows the user to filter the displayed information by: measurement flow, type, or period.

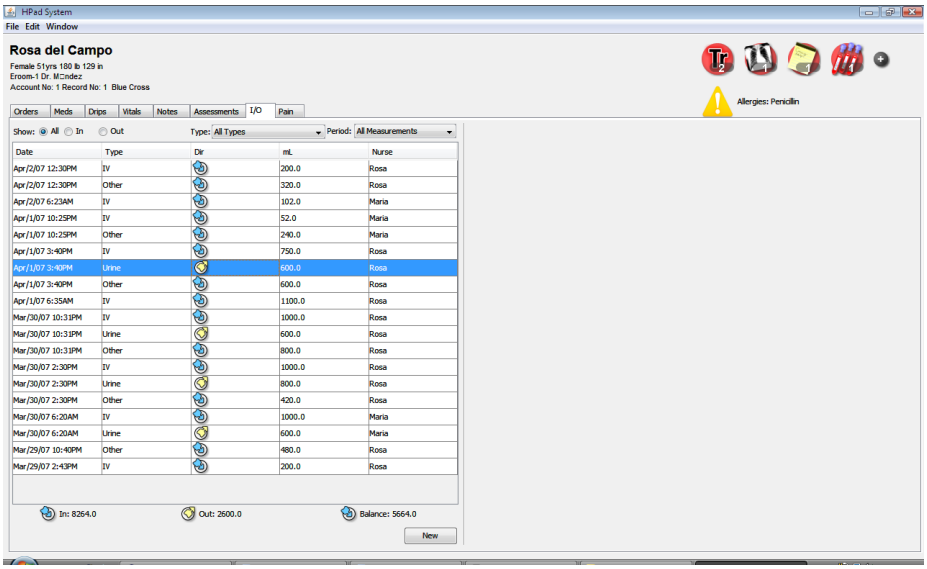

Figure 3.14 Intake/Output Interface

If the user clicks on the New button a New Intake & Output window (Figure 3.15) opens. After the users select the flow, type, and amount they can click the Save button to store the information. Once the information is stored it will appear in the Intake/Output interface.

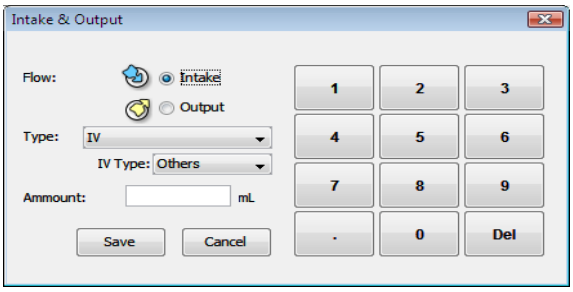

Figure 3.15 New Intake & Output Interface

3.3.7 Pain Interface

The Pain interface (Figure 3.16) is used to document the patient's pain assessments. This interface has a list with the date, ID and classification of the assessments in the left section. In the right section of the interface a detailed description with a graphical representation of the selected pain assessment is displayed.

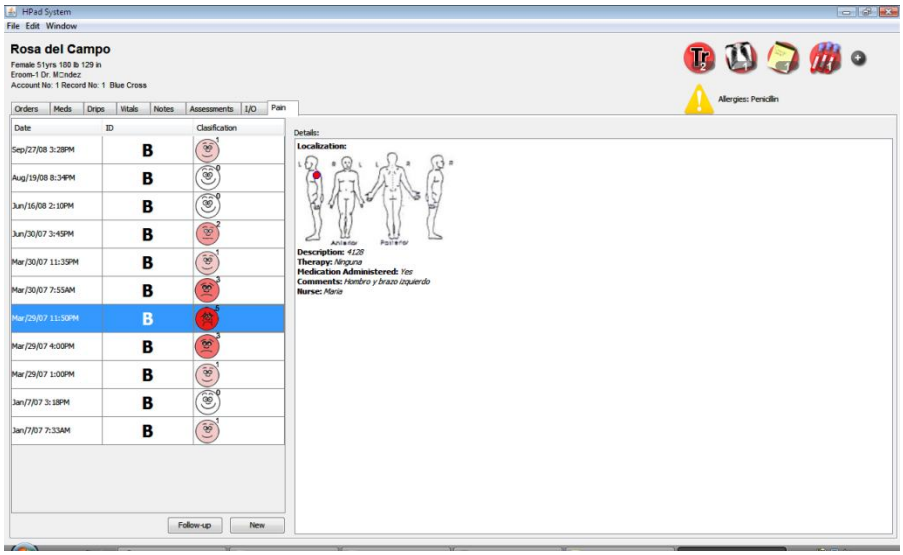

Figure 3.16 Pain Interface

To create a new pain assessment, users must click on the New button which will open the New Pain Interface form (Figure 3.17). A new assessment is created by clicking on the Save button after entering the required information.

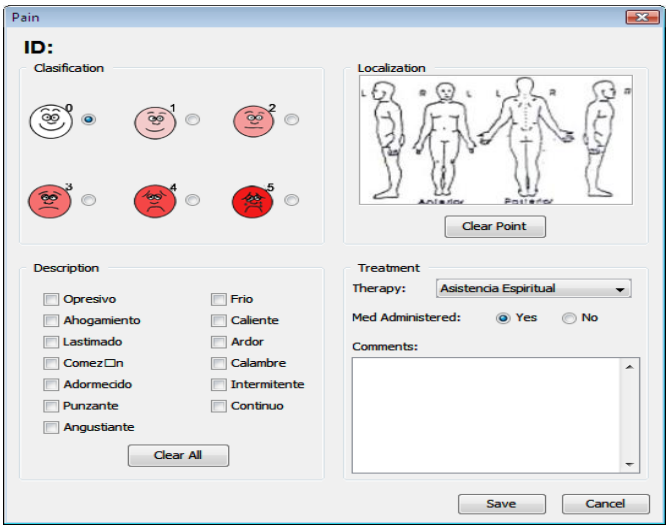

Figure 3.17 New Pain Interface

3.4 ARM Deficiencies and Missing Functionality

The following section describes the deficiencies and missing functionality found during the analysis phase of the ARM functionality and implementation. These, represent an opportunity to improve the overall functionally of HPAD including the ARM subsystem.

- *Predefined Thresholds* - As previously discussed, the main goal of the ARM is to generate alert and reminders from physician orders stored in the database record from HPAD. This alerts and reminders are generated based in predefined values called

thresholds, for example a Physician Discharge Order (in which the urgency is calculated based in the elapsed time, a detailed description of urgency calculation is explained in Chapter 4) the low level threshold could be 1 hour, the medium level could be 2 hours and the high level 3 hours. The problem with predefined thresholds is that hospitals have different ways of attending their patient's physician orders. Also, a hospital may add a new process into their system which may not be mature enough to know the acceptable thresholds, so the hospital may want to establish some thresholds initially and as time passes and the process become more robust they may want to change the thresholds to more accurate values. Thus, a way to allow special users to change these thresholds values needs to be added into the ARM functionality.

- *Lack of Customization* - As discussed in previous chapter, the ARM uses icons to represent the reminder categories, these icons cannot be changed without the need to go into the source code to make the change and then recompiling the code. Also, the layout or ordering of the ARM's reminder categories are hard coded and cannot be changed. Furthermore, the ability to disable/enable any particular reminder category is missing in the original ARM. Thus, the mechanism and features to offer this kind of customization needs to be added into the ARM functionality.

- *Lack of Reports* - Even with all the studies and research that HPAD has been through, there has never been a project done to implement a report or statistical module for the system. This could be a very useful feature for the analysis of patients' clinical information and quality of service evaluations.

- *Database Management System (DBMS)* - The ARM uses the Microsoft SQL Server 2000 Standard Edition as DBMS. This is a deprecated system with outdated features and data types. Thus, migrating the database records from SQL Server 2000 to the latest version from Microsoft which is the SQL Server 2008, have to be considered.

- *Limited Tooltip* – ARM presents a tooltip with a short description whenever (Figure 3.2) the mouse is placed over an active reminder or alert. Important information from the active reminders or alerts (for example remaining time until next medication or order due date) is missing from the tooltip. Therefore a redesigned tooltip needs to be implemented.

- *Development Tool* – The latest tool used for the development of the ARM and HPAD was the open source development platform Eclipse 3.2. Even thought this tool is very powerful, it does not provide any GUI editing features what makes the interfaces editing tasks very difficult compare to other tools with this type of functionality.

- *Source Code* – Due to a great deal of redesigns, HPAD and all of his subsystems have been affected by plenty of code patches, which makes the source code more difficult to maintain or change. Therefore, to be able to implement the new functionality mentioned in the first bullet the code of the ARM should be redesigned from scratch.

- *Other Issues* – After analyzing ARM's functionality, a critical feature was not working properly. This feature is the one in charge of storing the currently logged user's personal

information in the application. Thus, the logic in charge of managing this information needs to be revised.

The following chapter presents a new subsystem or module that extends the functionality of the original ARM subsystems. This new module is the answer to the previously mentioned ARM deficiencies and missing functionality.

CHAPTER 4 – ASM SUB SYSTEM DESCRIPTION

4.1 Introduction

The first part of this chapter describes the mechanisms used by the ARM to calculate the urgency levels of the physician order. This is followed by a description of the new Administrator and Statistical Module (ASM). The ASM was developed using NetBeans 6.5.1 [NetBeans09], JAVA 1.6 [JAVA09] and SQL Server 2008 Standard Edition from Microsoft Corporation [SQLServer09]. The system requires a display resolution of at least 1280 pixels by 800 pixels. The description of the ASM is from the point of view of user interfaces.

4.2 Physician Orders Urgency Calculation

This section describes the classification and calculation mechanisms of the urgency levels for the physician orders. The urgency level is calculated based on different strategies. The following list enumerates these strategies:

- *By Deadline*: In this type of urgency strategy the urgency level is calculated based on the remaining time until the physician order due date and time. For example, an order could have a low level urgency if there is 1 hour remaining for its due date, on the other hand it could have a medium level urgency if there is 30 minutes remaining, and finally it could have a high level urgency if the order has passed the due date.

- *By Elapsed Time*: The urgency level in this type of strategy is calculated based on the elapsed time since the order creation. For example, if 1 hour has passed since

the order was created the order could have a low level urgency, alternatively if the elapsed time is equal or greater than 2 hours the order could have a medium level urgency and finally if the elapsed time of the order is greater than 3 hours the order could have a high level urgency.

- *By Working Shift*: This urgency strategy is basically the same as the *By Deadline* strategy, the only difference is that the due date of the order is not the date used to calculate the urgency level, instead the next working shift of the hospital with the order creation day is used for the urgency level calculation. Assume that the regular working shifts of hospitals are from 7:00 a.m. to 3:00 p.m. for the first shift, from 3:00 p.m. to 11:00 p.m. for the second shift, and from 11:00 p.m. to 7:00 a.m. the third shift. Thus, for orders created from 7:00 a.m. to 2:59 p.m. the due hour will be 3:00 p.m., in the other hand for orders created from 3:00 p.m. to 10:59 p.m. the due hour will be 11:00 p.m., and for orders created from 11:00 p.m. to 6:59 a.m. the due hour will be 7:00 a.m. For example, if an order is created on July 4 at 8:50 a.m. the due date for the calculation will be July 4, 2009 at 3:00 p.m., and if the order was created on the same day but at 5:55 p.m. the due date for the calculation will be July 4, 2009 at 11:00 p.m.

- *By Periodical Intervals*: This urgency strategy uses the same inputs as the *By Deadline* strategy to calculate the urgency level, but for this strategy the thresholds values will change based on how recurrent the order is. For example, a medication order could be prescribed to a patient for every two hours, so for this

type of recurrence the urgency strategy could use 30 minutes for the low level urgency level, 15 minutes for the medium urgency level and 0 minutes for the high level urgency. The thresholds values for this type of strategy are going to be much higher according to how recurrent the order is. Thus, an order prescribed for every 4 hours will have less time to be attended than an order that is prescribed for every 8 hours.

- *By Quantity*: Although this strategy is not used currently by any of the physician orders it is important to be mentioned, because it may be used in the future. This strategy calculates the urgency level based on the quantity of orders in a reminder category complying with a particular aspect. For example, this strategy could be used to calculate the urgency level based on the total of pending physician orders in a particular reminder category. For example, a reminder category could have a low level urgency if it has 3 pending orders, also it could have a medium level urgency if it has 5 pending orders and finally it could have a high level urgency if it has 8 or more pending orders.

Now that the urgency calculations strategies have been explained, it is important to present which strategy uses every available physician order in the HPAD System. Table 4.1 provides a detailed presentation of the physician order urgency calculation; these classifications were established based on compiled information as results from meetings with Dr. Celia Colon Rivera of the University of Puerto Rico at Mayaguez.

Table 4.1 Physician Orders Urgency Strategies

Order Name	By Deadline	By Elapsed Time	By Working Shift	By Periodical Intervals
Discontinue		✓		
Activity		✓		
Diet		✓		
Intake & Output			✓	
Restraint		✓		
Therapy				✓
Vital Sign				✓
Notes			✓	
Diagnostic		✓		
Physical Exam		✓		
X-Rays		✓		
Drip				✓
Medicine				✓
Consult		✓		
Laboratory	✓			
Discharge		✓		
Admission		✓		

4.3 Reminder Categories Interface

The Reminder Categories interface (Figure 4.1) is used to manage the ARM's reminder categories. This interface provides the user the necessary components to create, edit, and delete existing reminder categories. This interface is divided in two sections, the left section were the existing reminder categories are displayed in a table with the picture, name, and status of the categories; and the right section that shows the details (name, status, image file name and the physician order) of the selected reminder category in a text panel. In addition, a button to logout from the ASM appears on the right upper corner of the window.

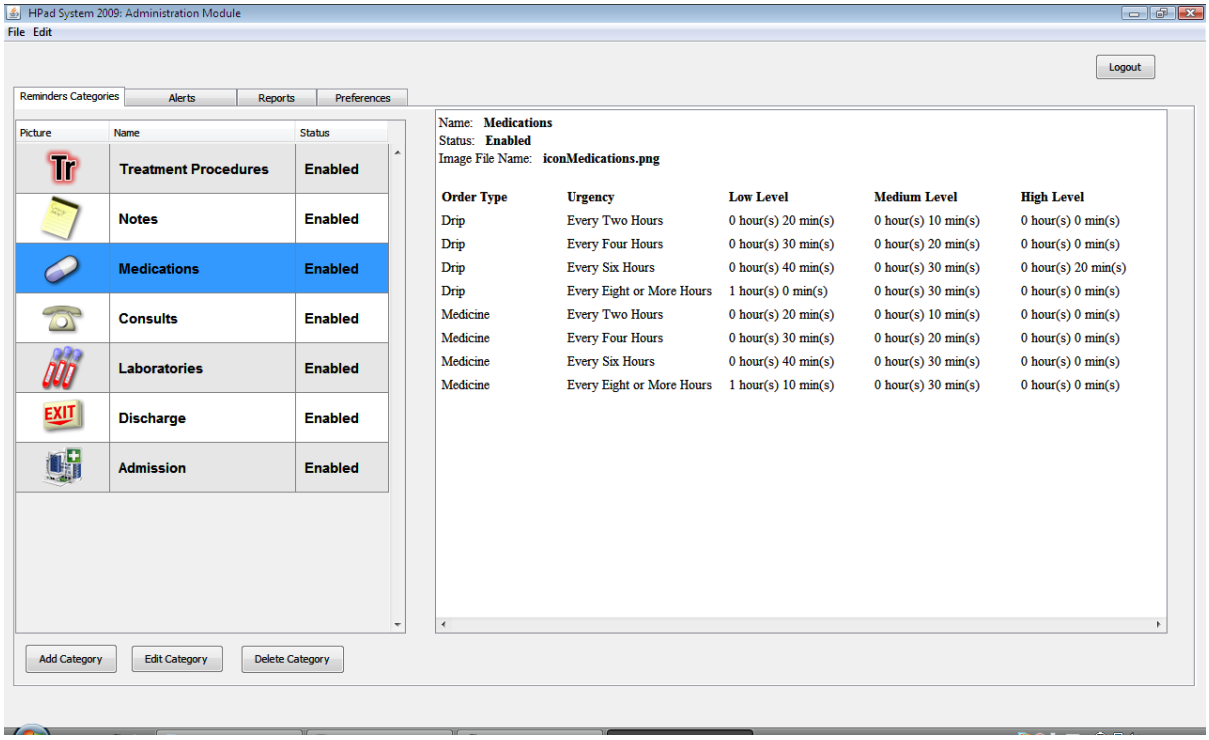

Figure 4.1 Reminder Categories Interface

Suppose a user wants to create a new reminder category with the following parameters:

- Name: Admission
- Status: Enabled
- Image: iconAdmission.png
- Order Types:
    - Admission
        - Low Level: 1 hour
        - Medium Level: 2 hours
        - High Level: 3 hours

The user needs to click on the Add Category button which will open an Add Reminder Category window (Figure 4.2). In this interface the user must select the category name, status, picture, and the associated physician orders.

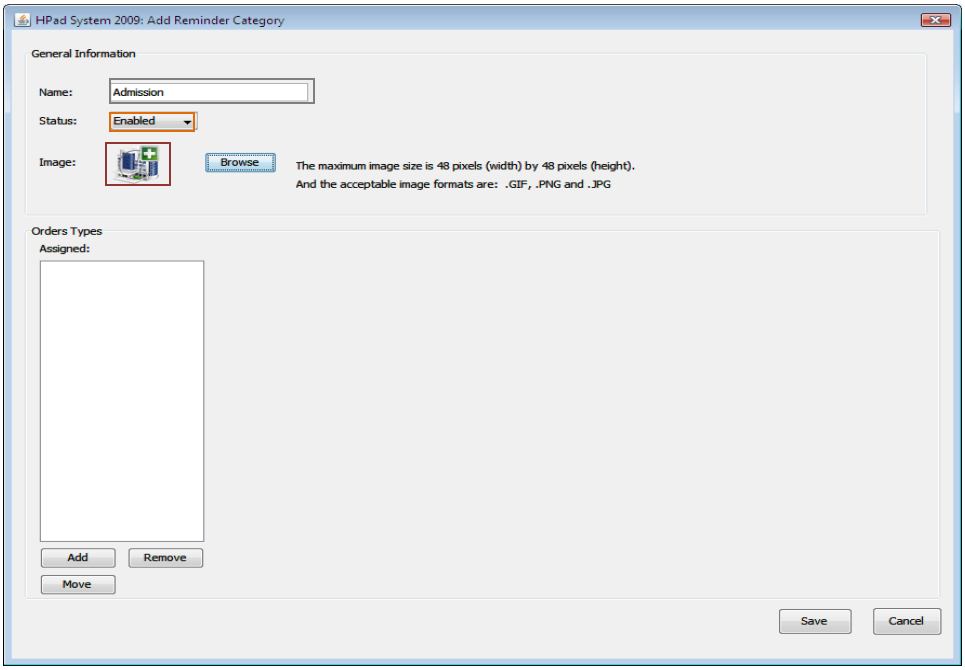

Figure 4.2 Add Reminder Category Interface

To select the physician orders types of the new reminder category the user needs to click on the Add button of the Order Types section and this will open an Available Physician Order Types window (Figure 4.3). In this window the user needs to select the desire physician order type which in this case is the Admission order type. After the user has selected the desire order type, he/she needs to click on the Add button to include the order type in the reminder category.

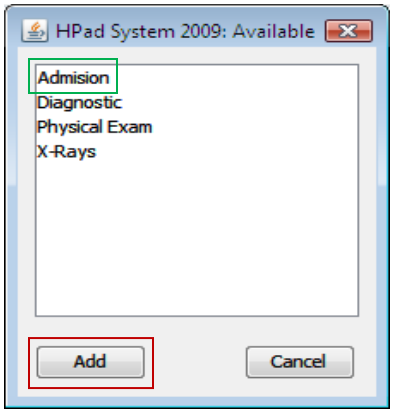

Figure 4.3 Available Physician Order Types Interface

After the Admission physician order type have been selected, the user must enter the thresholds values which in this example are 1 hour for low level, 2 hours for medium level and 3 hours for high level (Figure 4.4).

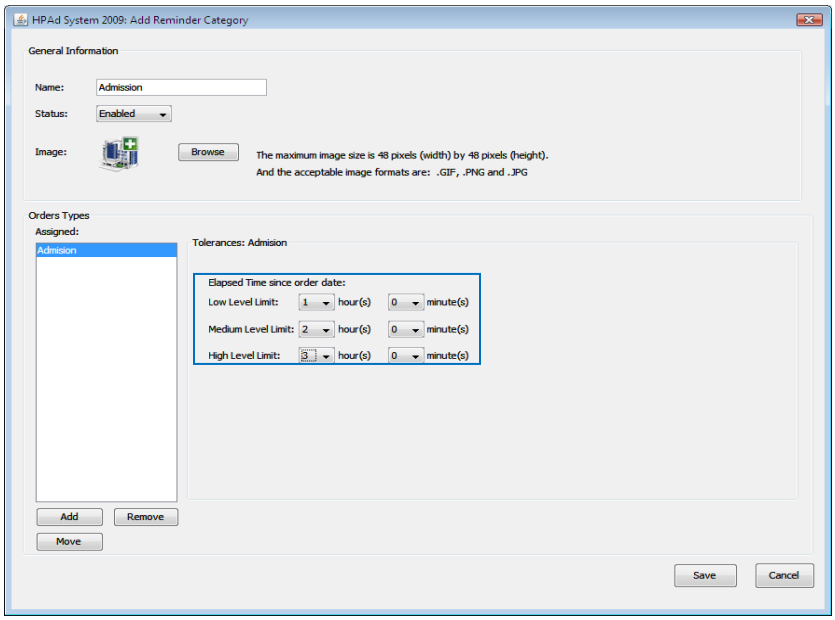

Figure 4.4 Add Reminder Category Threshold Values

Finally, when the user clicks on the Save button of the Add Reminder Category window, the information is saved in the database records and a message window (Figure 4.5) indicating the results of the process appears. After the user click on the OK button of the message window the Reminder Categories Interface information is updated with the new category.

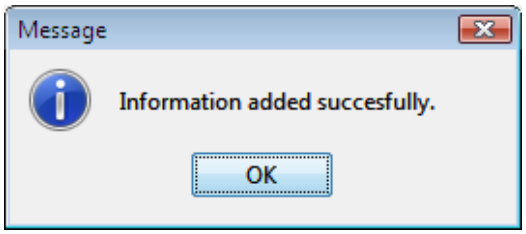

Figure 4.5 Message Window

4.4 Edit Reminder Category Interface

The Edit Reminder Category interface shown in Figure 4.6 is similar to the Add Reminder Category interface with the difference that the button to submit the information in the Edit Reminder Category is labeled “Update” and in the Add Reminder Category interface the button is labeled “Save”. Also, this interface is used to edit an existing reminder category and the Add Reminder Category interface is used to create new categories.

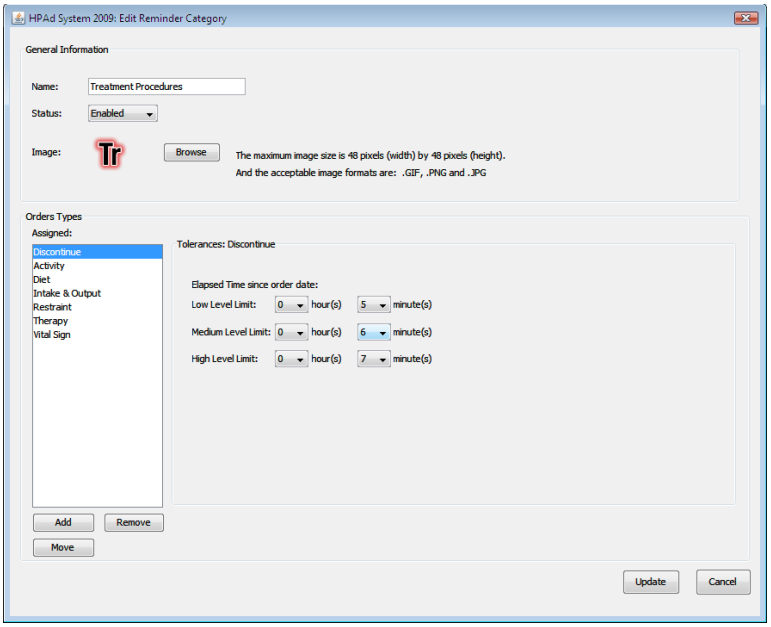

Figure 4.6 Edit Reminder Category Interface

4.5 Delete Reminder Category Interface

The Delete Reminder Category interface shown in Figure 4.7 allows the users to eliminate a reminder category from the system. In order to remove the category the user needs to select the desire category in the Reminder Category interface (Figure 4.1) and then click on the Delete Category button which will open the Delete Category interface. Then, when the user

confirms category deletion by clicking on the Yes button the information of the category is completely removed from the database records.

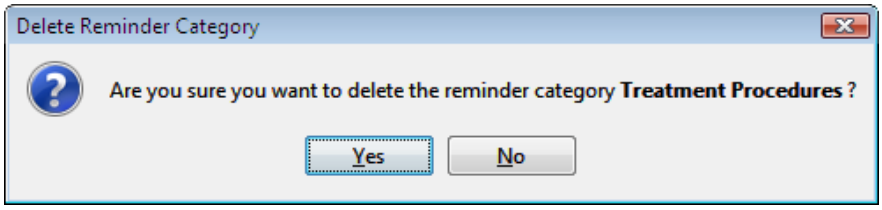

Figure 4.7 Delete Reminder Category Interface

4.5 Alerts Interface

The Alerts interface (Figure 4.8) allows users to manage the ARM's existing Alerts which are based on patient's abnormal laboratory results. This interface is divided in two sections, the upper section where the general information of the alerts category is displayed; and the bottom section where the different alerts are displayed in a table with the name and status. In addition, the acceptable laboratory values of the selected alert are displayed in the bottom section.

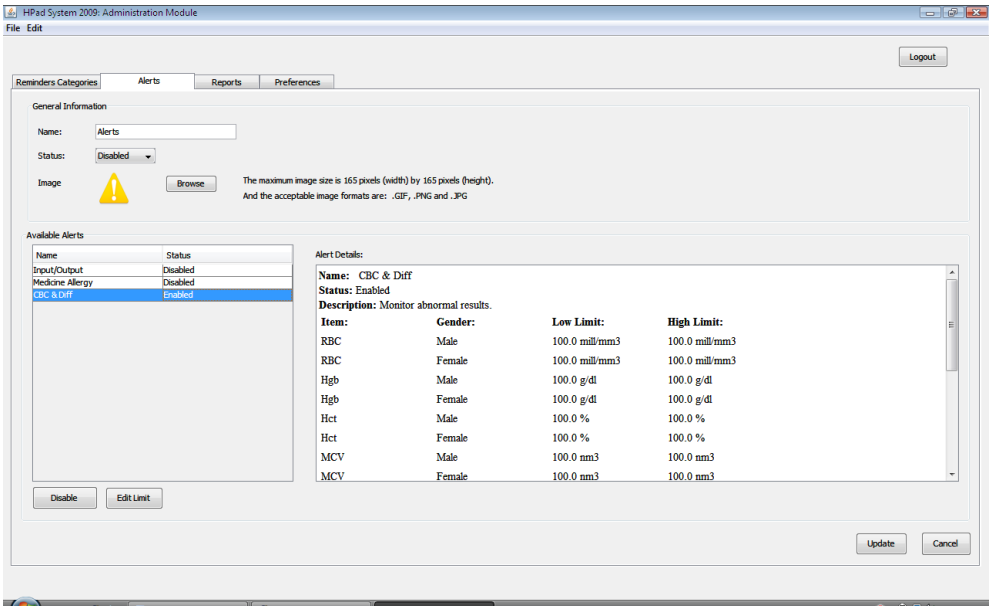

Figure 4.8 Alerts Interface

This interface provides a textbox to enter the alerts category name. The alerts status can be changed using the combo box in the header section. When the user clicks on the Browse button, a file dialog is displayed to allow users to select the image for the alerts category.

In order to edit the limits or acceptable values of an alert category, the user must select the desired category and click on the Edit Limit button which will open an Edit Limits Window (Figure 4.9). This window consists of a table classified by item, gender, low limit, high limit, and measurement. After the new limits have been set, the user needs to click on the Update button to save the new limits in the database records.

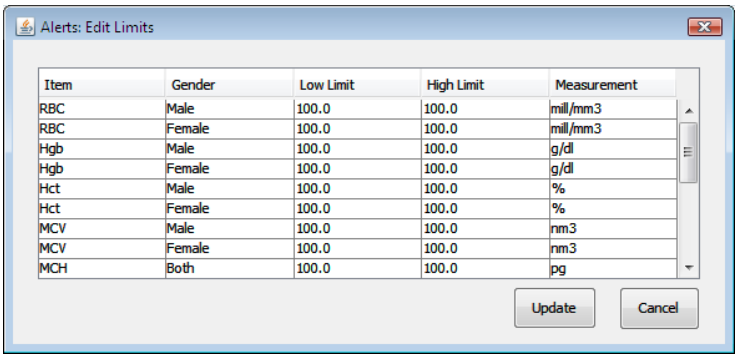

Figure 4.9 Edit Limits Interface

4.6 Reports Interface

The purpose of the Report interface, shown in Figure 4.10, is to give users the ability to generate reports of the turnaround levels and times of the attended physician orders. This interface provides a combo box to select if the report is going to be based on a patient, working area, or order type. Also, additional combo boxes are provided to filter the report by specific working shift and by the type of chart (Average Turnaround Acknowledge Level, Average Turnaround Acknowledge Time and Average High Level Turnaround Time).

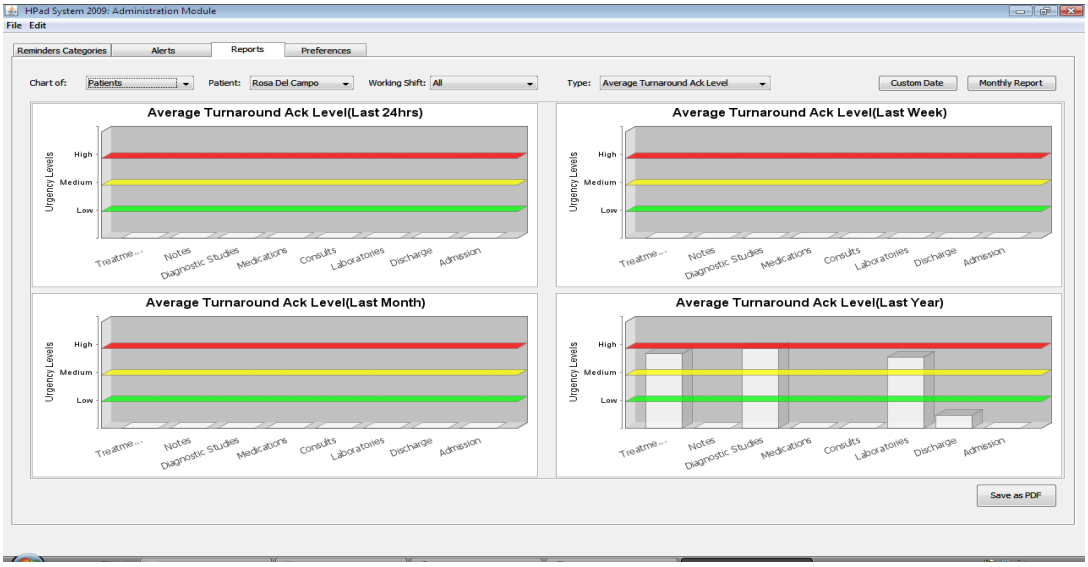

Figure 4.10 Reports Interface

For example, if a user wants to create a report for the turnaround acknowledge level of the physician orders attended for the patient Rosa Del Campo from May 1, 2009 to June 30, 2009, then the user needs to click on the Custom Date button which will open the Custom Date Report window (Figure 4.11). In this interface the user can establish the start and end date of the report using the date chooser components provided in the interface. Thus, to create the desire report the user needs to select May 1, 2009 on the date chooser labeled From and June 30, 2009 on the date chooser labeled To.

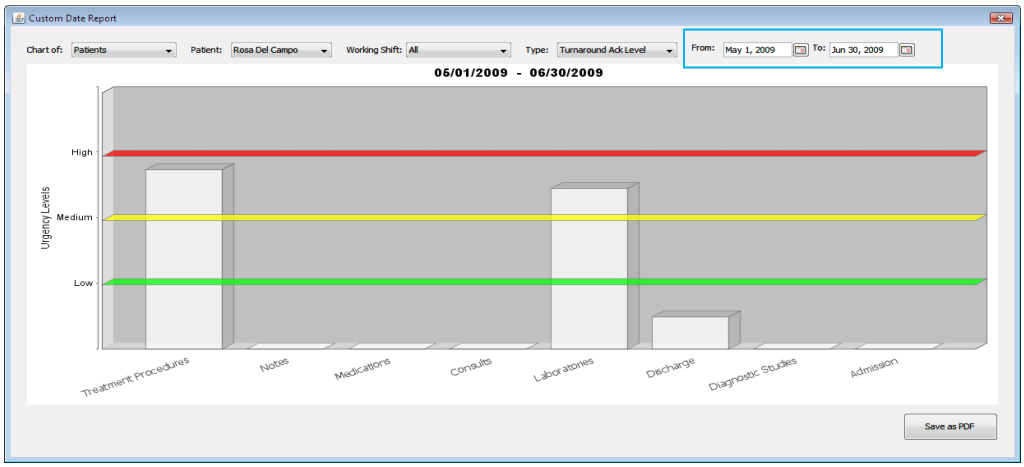

Figure 4.11 Custom Date Report Window

When the user wants to save the generated charts, they can click on the Save as PDF button (provided in both interfaces), and a File Save dialog (Figure 4.16) will appear asking for the desire storage location of the PDF charts file.

Finally, if a user wants to create a monthly report for the month of May, 2009, he/she must click the Monthly Report button of the Reports interface which will open a Generate Monthly Report window (Figure 4.12). After the user has selected May, 2009, he/she most click on the Ok button which will open a File Save Dialog (Figure 4.16) to select the desire file location. Then, when the user selects the destination location a Monthly Report Status window (Figure 4.13) will appears until the report is generated. An example of the first page of a monthly report can be found on Appendix C.

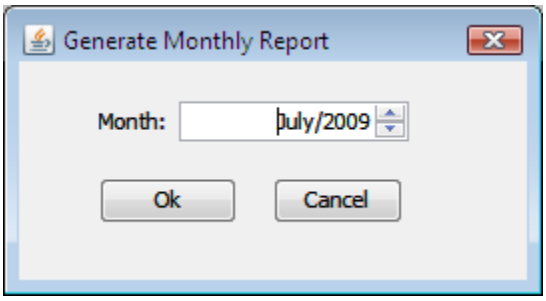

Figure 4.12 Monthly Report Interface

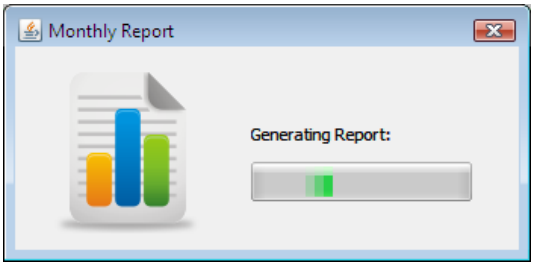

Figure 4.13 Monthly Report Status

4.7 Preferences Interface

As the name may suggest, the purpose of the Preferences interface shown in Figure 4.14 is to offer the mechanisms to customize the appearance of the ARM. This interface is divided in two sections; the General Information section which displays general information of the ARM; and the Categories Order section were the layout of the Reminder Categories are displayed.

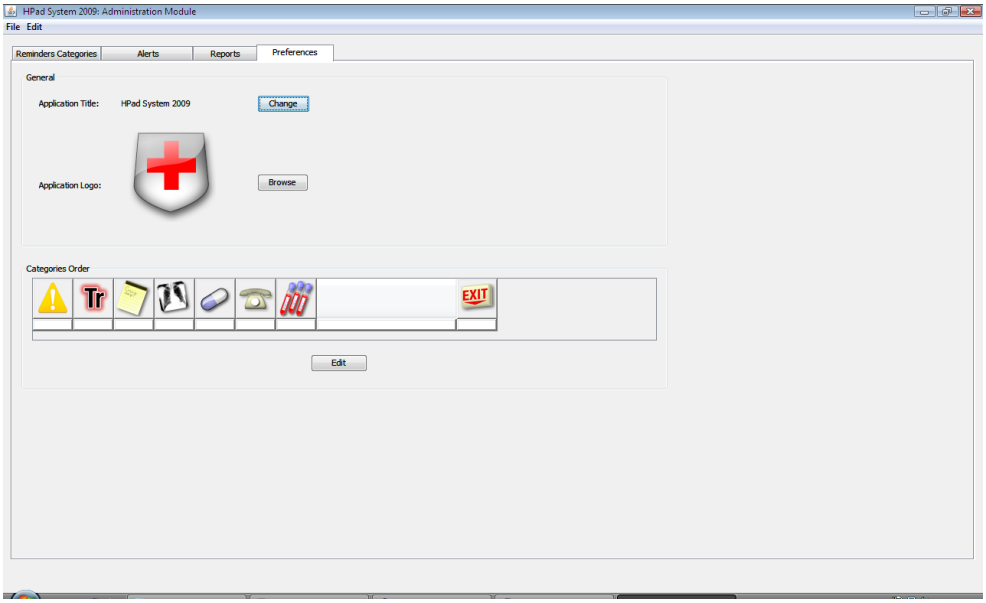

Figure 4.14 Preferences Interface

In order to change the application title the user must click on the Change button which will open a Change Application Title window (Figure 4.15). Once the title is saved, all the titles of the application windows will be updated.

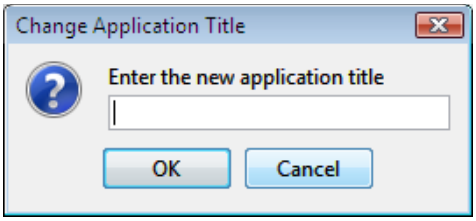

Figure 4.15 Change Application Title Interface

Furthermore, users can also change the logo for the application. A click on the Browse button will open a File Save Dialog window (Figure 4.16) used for this purpose. After the user has selected the new application logo, all the places where the logo is used will be updated.

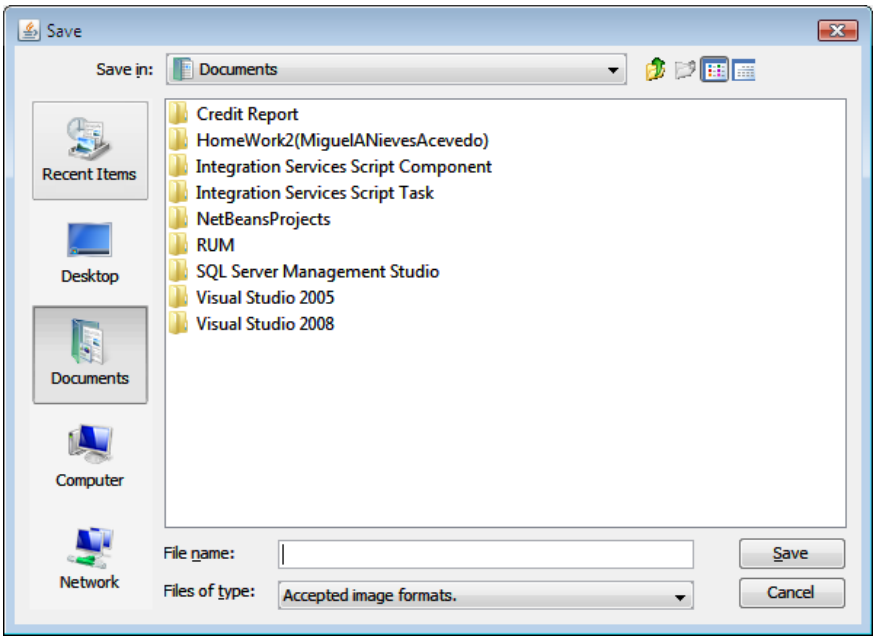

Figure 4.16 File Save Dialog Interface

Finally, the users can change the order in which the categories in the ARM's Patient List Window (Figure 3.2) are displayed. To make the change users must first click on the Edit button which will open a Change Categories Order window (Figure 4.17). In this window the user can change the order of the categories by making a drag and drop movement of the desired category to the desired location. In addition, if the users want to add a space between any of the categories, then they can click on the Add Blank Column button which will add a blank column at the end of the available categories. When users finish customizing the categories layout they can save the changes by clicking the Update button.

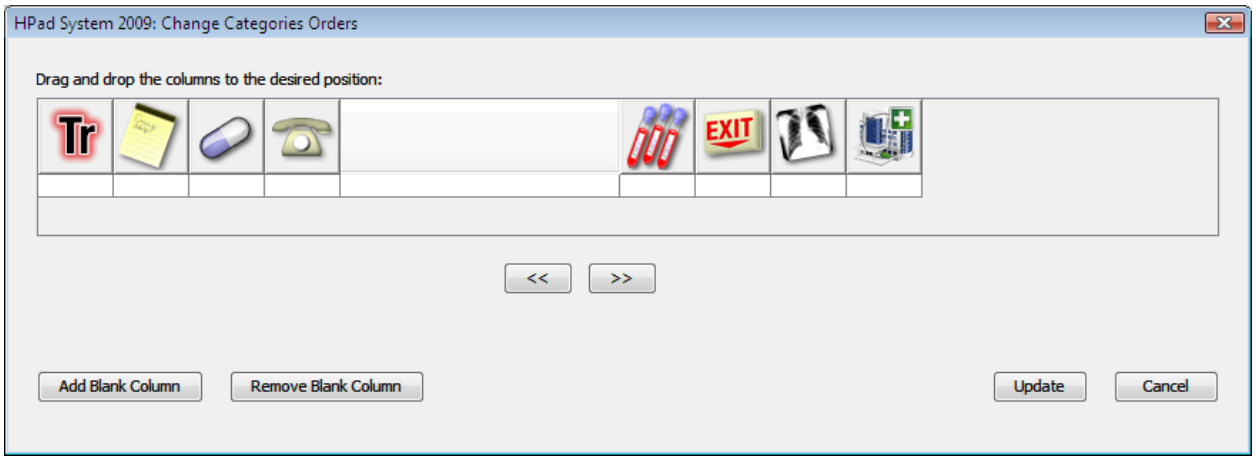

Figure 4.17 Change Categories Order

This chapter described the functionality added in order to extend ARM and to make it fully customizable for the hospital's needs. The following chapter describes some upgrades and enhancements made to the original system.

CHAPTER 5 - ARM'S UPGRADES AND ENHANCEMENTS

5.1 Introduction

In order to be able to implement the new Administrator and Statistical Module (ASM), the original ARM and several functionalities from HPAD were redesigned. In addition of improving the application's overall performance and maintainability for future developments, some tools were changed or upgraded to the most recent releases. The following sections describe these upgrades and enhancements.

5.2 Source Code Redesign

One of the biggest challenges in the implementation of the ASM was to introduce the new functionality while trying to reuse the original ARM's source code. However, after analyzing the original implementation it was evident that ARM's source code was suffering from a very high level of fragmentation making the redesign from scratch, necessary.

> *"Coupling refers to the strength of a connection between two routines. Coupling is a complement to cohesion. Cohesion describes how strongly the internal contents of a routine are related to each other. Coupling describes how strongly a routine is related to others routines. The goal is to create routines with internal integrity (Strong cohesion) and small, direct, visible, and flexible relations to other routines (Loose coupling)"* (Allan Shalloway) [Shalloway05]

After the decision of redesigning the ARM from scratch was made, one thing was very clear, a better, improved, and well defined source code was needed. To accomplish this, the use of object oriented design principles like design patterns were applied. In addition, in the development of the new source code, the Java classes were created taking into consideration the *Strong cohesion* and *Loose coupling* design principles. A simplified UML class diagram of the new ARM source code can be found in Appendix A.

Design Patterns are recurring solutions to design problems that occurs over and over again in software development. Providing proven solution for issues related to software development is the main advantage of design patterns. By using already established designs or solutions, developers have a head start on their problems and can avoid known mistakes. They do not have to reinvent solutions for commonly recurring problems. In addition, Design Patterns are flexible solutions which are easy to modify and maintain. The reason for this is that they are time tested solutions that have evolved into structures that can handle changes more readily compared to what often first comes to mind as a solution [Shalloway05].

> *"The purpose of the Facade design pattern is to provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use. This can be used to simplify a number of complicated object interactions into a single interface"* [Gamma07]

The Facade pattern was used to minimize the interaction complexity of the Java classes in the ARM source code, making future changes, enhancements or troubleshooting, easier to

manage. An example where this pattern can be seen is in the process of retrieving all the patients' information from the database and displaying it in the user interfaces of ARM. This task involves a connection to the database, a prepared statement to store the SQL query, a result set to store the result from the query and after all that, getting the information to the user interface and displaying it. With the use of the Facade pattern this complex task was simplified by using a class hiding all the process and interfaces used to accomplish the data retrieval and only offering a simple method called *getPatientData()* which does all the complex steps in the background and then the retrieved data can be manipulated in any desired way to make the display.

> *"The Strategy pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable. Strategy lets algorithm vary independently from clients using it"* [Gamma07]

The strategy pattern was used in the implementation of the physician's orders urgency calculation discussed in section 4.2. With the use of the strategy pattern the different types of urgency calculation were managed the same way without the need to know which specific type was been used. One of the benefits of using this pattern it is that if a new urgency calculation type needs to be added into the system, the source code will only be affected on the logic choosing which particular strategy to use at run time, thus, making future enhancement easier. An UML representation of the use of the strategy pattern can be seen in Figure A-1 of Appendix A.

"The Simple factory pattern uses a class where every method corresponds to an interface and the return value of the method is an instance of the implementation class of the interface" [Yan06]

This pattern was used in the ***AspectFactory*** and ***StrategyFactory*** classes (Figure A-1 on Appendix A). The purpose of using this pattern was to encapsulate all the logic instantiating objects of similar hierarchy. Thus, making future changes in the hierarchy, easier. Another example of the use of this pattern was in conjunction with the previously discussed pattern, using the simple factory simplified the logic to decide which urgency strategy to use (to instantiate) at runtime.

"*Data Access Object (DAO) patterns provide an abstraction between business logic layer and persistent storage layer through encapsulation of the access to data resources and wrapping up the implementation details of data resources.****"*** [Cheng09]

The DAO pattern was used to isolate the business logic from the database access needs of the application. The main advantage of this pattern is that persistence logic (database access) can be changed without affecting the business rules as long as the method's interfaces remain the same. Also, thanks to the isolation of the business layer and the persistence layer, the developers can concentrate on their respective domains. For example, the developers in charge of displaying the patients information stored in HPAD's database records do not need to worry on how the data is going to be retrieved from the database, the only thing they need to know is which DAO object needs to be used to get the desire information and how they are going to manipulate it to make the needed display.

5.3 Tooltip Redesign

The ARM uses a tooltip to provide details of the alerts and reminders of patients. The purpose of the tooltip is to provide immediate information to the user about the actives physician orders in a reminder category without the need to go into a HPAD interface related to the active reminder. The Original ARM's tooltip (Figure 5.1) only displays the highest urgency of the reminder category physician orders, the total quantity of reminders, the date of the oldest order, and a static text.

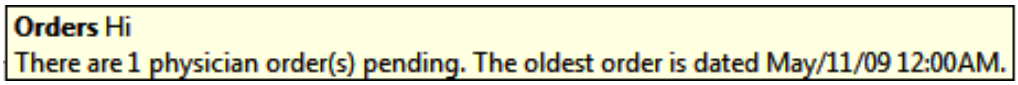

Figure 5.1 Original Tooltip

As can be appreciated from figure 5.1, the original Tooltip design lacks some useful information like individual details of the physician order including the elapsed or remaining time, order title, and current urgency level. For this reason a new and improved tooltip (Figure 5.2) was designed to display the physician order details individually. The following list shows the information included on the new tooltip:

- The reminder category title which in figure 5.2 is "Treatment Procedures".
- Individual listing of every reminder grouped by order types.
- The urgency level for every reminder listed.
- The creation date and current date of each order listed.
- The elapsed or remaining time of every reminder listed.

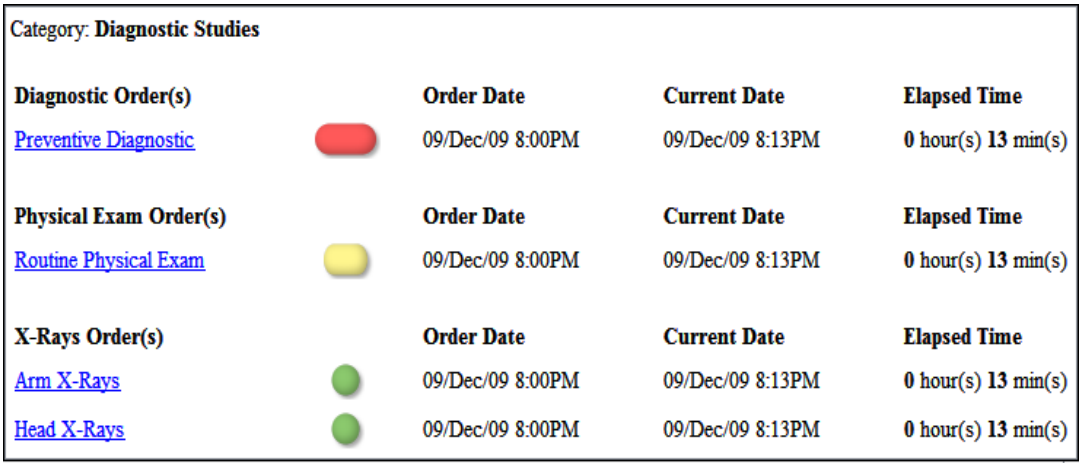

Figure 5.2 Redesigned Tooltip

In addition, a mechanism to attend or respond to a specific physician order details was introduced in the new tooltip. This new mechanism allows the users to attend a specific physician order by clicking on a specific physician order title. This action will open a HPAD interface with the details of the selected physician order.

5.4 Database Management System

In order to improve the performance of ARM and HPAD, a migration from SQL Server 2000 to the 2008 version was realized. In addition to the improved performance, this new database management system (DBMS) offers new features like new data types, policy based management, data compression, transparent data encryption and the introduction of the merger operator [SQLServer09].

Thanks, to the previous redesign work done to HPAD by Carlos Duarte [Duarte08] the migration of the database was significantly straightforward. The incorporation of the Template pattern in the database connection process by his work simplified the migration. The steps needed for the migration were exporting the database from SQL Server 2000 to a backup file and

importing it to the SQL Server 2008 database engine using the Management Studio [SQLServer09]. Finally, for the new ARM to be able to connect to the new database engine a new subclass of the Template created by Carlos was developed.

5.5 Development Tools

The development tool used for the previous work done to ARM's source code was Eclipse 3.2. Although this tool is powerful, it does not provide features to edit or create interfaces from the design point of view. Therefore, with the purpose of facilitating the design and management of user interfaces, the source code was imported to NetBeans 6.5.1, a more complete tool that offers user interfaces editing features.

The enhancements and upgrades discussed in this chapter contributed to improve the performance of the ARM. For example, a place where a speed improvement can be appreciated is in the Patient List Interface (Figure 3.2). Before the upgrades and enhancements were realized the loading time of the interface was 4.55 seconds in average [Duarte08] and after the changes were made the loading time was decreased to an average of 2.85 seconds. It is very important to remember that the new ARM uses more dynamic information like reminder categories, thresholds values, images and customization information which require more database access and even with these additions, it is able to load the information 1.5 seconds faster than the original ARM. In addition of helping to improve the performance of the application, the redesign of the source code is going to facilitate future changes. Finally, the upgrades made will help the ARM stay up to date with the new available technologies.

CHAPTER 6 – HEURISTIC EVALUATION

6.1 Introduction

An essential factor that determines the acceptance of user interfaces is their level of usability. Usability is a qualitative attribute that assesses how easy user interfaces are to use [Nielsen93]. Thus, making user interfaces with a high level of usability is an important aspect of all software. This can be achieved by incorporating usability engineering principles in the applications' development cycle. Jakob Nielsen associates usability with the following systems attribute [Nielsen93]:

- *Learnability* – refers to how easily the users learn to use a system.
- *Efficiency* – refers to the performance of users once they have learned the system.
- *Memorability* – refers to how easy it is to remember how to use the system, so casual users do not need to learn the system again when returning to it.
- *Errors* – refers to the ability of a system to reduce and prevent user errors.
- *Satisfaction* – refers to how satisfied are the users \when using a system.

There are many different activities and methods used to evaluate and improve the usability of user interfaces. Some of these are: interviews, focus groups, cognitive walkthroughs, task analysis, think aloud, usability testing, and heuristic evaluation. Heuristic evaluation is one of the most popular usability inspection methods. Heuristic evaluation involves having a small

set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics") [Nielsen09].

6.2 Heuristic Evaluation

6.2.1 Procedure

The following sections describe a study conducted by three evaluators with knowledge in Usability Engineering, Human-Computer Interaction, and the heuristic evaluation method. The procedure started by giving the evaluators an overview of the system to be evaluated. They were asked to focus on suitable graphics and colors, relevant information, constant feedback, good consistency, good error messages, and errors prevention. As it is common on heuristics evaluations, the result of their evaluations was a list of usability problems found and a severity rating of each problem.

With the aim of facilitating the user interfaces flow and exercising most of the components of the user interface, a list of common tasks was given to the evaluators. The following list enumerates those tasks:

1. Login into the administrator module using username: root, and password: admin
2. Read the reminder categories list.
3. Create a new reminder category with the following characteristics:
    a. **Name**: Usability Evaluation
    b. **Status:** Enabled
    c. **Image:** admission.png (located in the users' desktop)
    d. **Order Types:**

i. Admission
    1. Low Level: 10 hour
    2. Medium Level: 20 hours
    3. High Level: 30 hours

4. Save the new category

5. Logout from the administrator module

6. Login into the user module using username: 3000, and password: 3000

7. Verify that the category is showing up in the patient's list window.

8. Logout from the user module

9. Login into the administrator module using username: root, and password: admin

10. Edit the reminder category created in the step 4.

    a. **Name**: Admission
    b. **Status:** Enabled
    c. **Image:** admission.png (located in the users' desktop)
    d. **Order Types:**
        i. Admission
            1. Low Level: 1 hour
            2. Medium Level: 2 hours
            3. High Level: 3 hours

11. Save changes

12. Disable the reminder category edited in step 10.

13. Logout from the administrator module

14. Login into the user module using username: 3000, and password: 3000

15. Verify that the category is not showing up in the patient's list window.

16. Logout from the user module

17. Login into the administrator module using username: root, and password: admin

18. Enable the reminder category disabled in step 12.

19. Place the reminder category as the first reminder category in the GUI representation.

20. Logout from the administrator module
21. Login into the user module using username: 3000, and password: 3000
22. Verify if the reminder category placement was updated.
23. Logout from the user module
24. Login into the administrator module using username: root, and password: admin
25. Change the Alert CBC & Diff limits to:
    i. Plaquetas: Low Limit = 500 High Limit = 1000
26. Save changes
27. Generate a report of the patient Rosa Del Campo turnarounds acknowledge levels including all the working shifts from May 1, 2009 to July 4, 2009.
28. Save report as usabiltyEvaluation in your desktop.

6.2.2 Results

The observations shown in tables 6.2 to 6.4 are the results of the heuristics evaluations. The evaluators categorized them using Jakob Nielsen's usability severity rating (Table 6.1) These ratings are used to find the more severe usability problems which can help to allocate the available resources when deciding which user interface problems to fix. If the severity ratings indicate that several disastrous usability problems remain in an interface, it will probably be unadvisable to release it. But one might decide to go ahead with the release of a system with several usability problems if they are all judged as being cosmetic in nature [Nielsen09].

Table 6.1 Usability Severity Ratings [Nielsen09]

Description	Scale
Cosmetic problem only: need not be fixed unless extra time is available on project	1
Minor usability problem: fixing this should be given low priority	2
Major usability problem: important to fix, so should be given high priority	3
Usability catastrophe: imperative to fix this before product can be released	4

Table 6.2 Usability Problems Observed by Evaluator 1

Observation	Severity
The username label in the login window says "User Name" and is suppose to say "Username".	1
The Quit and Login buttons in the login window are too close to the textboxes.	1
The Quit button in the login window appears before the login button.	2
The letters size in the login window is little bit small.	1
The button for saving a new reminder category says "Ok". It may be better to use "Save" rather than "Ok"	2
There is no visible button to logout from the ASM	1
The Edit Category button in the reminder category window is placed apart from the other buttons to manage the reminder categories.	1
It may be better to add a combo box to edit the status of the reminder categories in the reminder category window	2
The button for saving the values of the alerts says "Update". It may be better to use "Save" rather than "Update"	2
The generate button in the custom date report window has no functionality.	1

Table 6.3 Usability Problems Observed by Evaluator 2

Observation	Severity
The image used for the treatment category should read Tx instead of Tr. Tx is the correct medical abbreviation for treatment.	1
No help in the categories window.	1
Form validation should highlight fields that did not passed validation.	1
The edit button on the reminder category should be placed with the add and deleted category buttons	2
The label Categories order on the preference window maybe it could read "Categories Display Order"	1

Table 6.4 Usability Problems Observed by Evaluator 3

Observation	Severity
The login window is too big.	1
The login button should appear before the quit button on the login window.	1
There is no message indicating that there is no available physician order in the Add/Edit reminder category window.	1
The spinners in the add/edit reminder category to change the thresholds values takes too long to edit the wanted value. The spinners should be change to combo boxes.	2
There is no need to use the Word "category" on the buttons of the reminder category window.	1
The edit category button should be placed between the add and delete buttons on the reminder categories window.	1
The logout button on the patient list window should be placed on the upper right corner.	1
The header for the urgency level on the tooltip should be added.	1
When canceling the save operation of a report the message saying "select the desired location of the report" should not appear.	1
The drag and drop reordering task in the Edit order window may be a little complicated. Thus, two buttons ("<<" and ">>") to change the order must be added	2
The "Update" message that appears after a create or edit operation most be removed.	2

6.2.3 Implications for redesign

The results of the evaluations were considered in the redesign of the ASM prototype system. The following list enumerates the changes made to the prototype system in response to the heuristic evaluation results:

1. The username label text was changed from User Name to Username.
2. More space was added between the Login and Quit button, and the textboxes.
3. The Login button was placed before the Quit button.
4. The reminder category save button text was changed from Ok to Save.

5. The edit category button was aligned with the others related buttons.
6. A combo box to edit the status of the reminder categories was added.
7. The text of the button to save the alerts values was changed from Update to Save.
8. The generate button in the custom report window was eliminated.
9. The login window size was reduced.
10. A message to indicate the availability of physician orders was added.
11. The spinners used to change the thresholds values were replaced by combo boxes.
12. A button to logout from the ASM was added.
13. Buttons to reorder the reminder categories order were added.
14. A help menu was added to the ASM.
15. The categories order label was changed to Categories Display Order

The results of the evaluations discussed in this chapter showed that only minor usability problems (the highest severity rating was 2) were found on the ASM's user interfaces. Although the observations did not present major usability problems, they helped fix some issues that were not visualized while developing the application. Therefore, the heuristic evaluation contributed to increase the usability level of the ASM.

CHAPTER 7 – CONCLUSION AND FUTURE WORK

7.1 Recapitulation

The main objective of the work described on this document was to design a system to extend the functionality of the Alert and Reminder Module (ARM) from HPAD. The new system called the Administrator and Statistical Module introduces new and advanced features to the original ARM functionality. The software was developed using Java which supports a wide variety of operating systems. The following list enumerates the new features introduced by the ARM:

- Ability to create, edit, and delete existing ARM's reminder categories
- Ability to change the physician orders threshold values without the need to change the application's source code and re-compilation.
- Ability to edit the alerts category information, including the acceptable patients' laboratory result values.
- A report management system, that will help hospitals evaluate their services. The system offers the ability to generate reports as PDF, making possible the use of reports in hospitals business related presentations.
- Full customization. Hospitals can use their emblem as the application logo and change the layout of the ARM user interface columns.

In order to be able to integrate the new functionality with the existing one, the source code of the Original ARM was redesigned from scratch. The redesign was based on object-oriented programming principles and incorporated the use of design patterns. The Façade, Strategy, Simple factory and the Data Access Object were the design patterns used while redesigning.

An important contribution was the redesign of the tooltip used by the ARM to provide details of the alerts and reminders of patients. The new developed tooltip displays more details about patient's reminders, like the elapsed or remaining time, order title and current urgency level of the physician orders which were missing in the original tooltip. In addition, the functionality to attend a physician order by clicking on the corresponding reminder was added by the new developed tooltip.

Some aspects of the development environment used by previous works were improved with the goal to increase the performance of the application and to facilitate future source code changes. This improvement was a change of the development tool from eclipse 3.2 to NetBeans 6.5.1, due to the fact that NetBeans offers advances user interface editing features that facilitates the design process.

Finally, the SQL Server 2000 database management system used by HPAD and the ARM was upgraded to the most recently released version, which is the SQL Server 2008. This DBMS offer better performance and features that were not available in the originally used DBMS.

The contribution of this project to both the ARM and HPAD will be very important for the acceptance of these software in hospitals. The ASM extension will allow them to establish the desired thresholds values for every physician order and to categorize them in any particular reminder category. This extension together with the customization features added, will allow hospitals to change the appearance and operation of the ARM and HPAD to make them unique and to match their particular needs. Moreover, the report management system included in ASM offers hospitals an important tool that could be used to study, evaluate, and improve their services. Furthermore, besides the benefits provided by the new functionality, the redesign made as part of this project will facilitate future development, changes, and extensions to the software.

7.2 Future Work

A work that should be considered for a future project is the development of an Alert and Reminder Module for Physicians. This system should have the same functionality as the ARM discussed in this work but with reminders related to physicians' tasks, like pending physical exams, and patients' abnormal laboratory results.

Even though the heuristic evaluation results showed minor usability problems, it is recommended to conduct a usability test with the real users of the system carrying out typical task, to identify usability problems by measuring the real users satisfaction, efficiency, and errors.

Another work that should be done is the implementation of a web-based application providing a similar functionality as the reports in the ASM. This way authorized hospital

personnel can view and analyze their services without need to have an installed application from any place.

Duplicated information and unused tables and fields are present on the current database model used by HPAD and the ARM. Therefore, in order to remove information redundancy it is recommended to review and redesign the database model as future work.

REFERENCES

[Borges97] Borges, J., González, M., et. al., "SAAS: Automatic System for Auto-Supervision in an Emergency Room", Proceedings of the 10th IEEE Symposium on Computer-Based Medical Systems (1997), pp. 48-53.

[Cheng09] Cheng, F., "A New DAO Pattern with Dynamic Extensibility", Proceedings of the Second International Conference on Information and Computing Science Volume 1 (2009), pp. 21-22.

[Crespo05] Crespo, G. "A Comparative Study of Nurses Accessing Electronic Patient Record Systems with PDAs and Tablet PCs". M.S. Thesis University of Puerto Rico, Mayagüez, P.R. (2005).

[DeClercq99] de Clercq, P.A., Blom, J.A, et. al., "A strategy for developing pratices guidelines for the ICU using automated knowledge acquisition techniques" in Journal of Clinical Monitoring and Computing Volume 15 (1999), pp. 109-117.

[Duarte08] Duarte, C., "Object-Oriented Re-Engineering Of The Nursing Documentation System HPAD", M.E. Thesis University of Puerto Rico, Mayaguez, P.R. (2008).

[Gamma07] Gamma, Erich, Richard Helm, Ralph Johnson and John Vlissides. "Design Patterns: Elements of Reusable Object-Oriented Software". Addison-Wesley books, 36th printing. (2007).

[Fung04] Fung, C., Woods, J., et. al., "Variation in Implementation and Use of Computerized Clinical Reminders in an Integrated Healthcare System" in The American Journal of Managed Care Volume 10 Number 11 (2004), pp. 878-885.

[Hasman00] Hasman, A., Bindels, R. et. al., "On The Use Of Reminder Systems in Healthcare" Proceedings of the 26th Euromicro Conference Volume 1 (2000), pp. 20-25.

[Mackinnon09] Mackinnon, W., Wasserman, "Integrated Electronic Medical Record Systems: Critical Success Factors for Implementation", in System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference (2009), pp. 1-10.

[Nielsen93] Nielsen, J. "Usability Engineering". Academic Press A Harcourt Science and Technology Company. 1993.

[Perez05] Perez, C., "Usability Study of a Patient Alerts and Reminders Display Component for an Electronic Medical Record System", M.S. Thesis University of Puerto Rico, Mayaguez, P.R. (2005).

[Ramos05] Ramos, J. "A Design and Implementation of an Electronic Physician Orders Entry System". M.E. Thesis University of Puerto Rico, Mayagüez, P.R. (2005).

[Reitstätter06] Reitstätter, M. Vierhapper, H., et. al., "Quality Improvement through a Reminder System in an Outpatient Department for Thyroid Disease" International Conference on Intelligent Agents, Web Technologies and Internet Commerce (2006), pp.142-146.

[Saleem05] Saleem, J., Patterson, E., et. al., "Exploring Barriers and Facilitators to the Use of Computerized Clinical Reminders" in Journal of the American Medical Informatics Association Volume 12 Number 4 (2005), pp. 438-447.

[Shalloway05] Shalloway, A., Trott, J., "Design Patterns Explained: A New Perspective on Object-Oriented Desgin", Addison Wesley Books 2th Edition (2005).

[Weimar09] Carrie, W., "Electronic Health Care Advances, Physician Frustration Grows", in ACPE Health Care Technology Survey (2009), pp. 8-15.

[Yan06] Yan, B., Huang, B., et. Al. "Cooperative Design and Development of Logistics Information System Based on Simple Factory Pattern", Proceedings of the IEEE Asia-Pacific Conference on Services Computing (2006).

WEB REFERENCES

[DAO09] Core J2EE Patterns - Data Access Object.
Available: http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

[Interface09] Object-Oriented Programming Concepts, What is an Interface.
Available: http://java.sun.com/docs/books/tutorial/java/concepts/interface.html

[JAVA09] Java Website.
Available: http://www.java.com

[MySQL09] MySQL Sun Microsystems Website.
Available: http://www.mysql.com

[NetBeans09] NetBeans Website.
Available: http://www.netbeans.org

[SQLServer09] Microsoft SQL Server 2008 Website.
Available: http://www.microsoft.com/sqlserver/2008/en/us/default.aspx

[Nielsen09] Jakob Nielsen's Website
Available: http://www.useit.com/

APPENDIX A

Table A-1 ARM Reminder Categories' Physician Orders

Treatment Procedures	Notes	Diagnostic Studies	Medications	Consults	Laboratories	Discharge	Admission
Discontinue	Notes	Diagnostic	Drip	Consult	Laboratory	Discharge	Admission
Activity		Physical Exam	Medicine				
Diet		X-Rays					
Intake & Output							
Restraint							
Therapy							
Vital Sign							

APPENDIX B

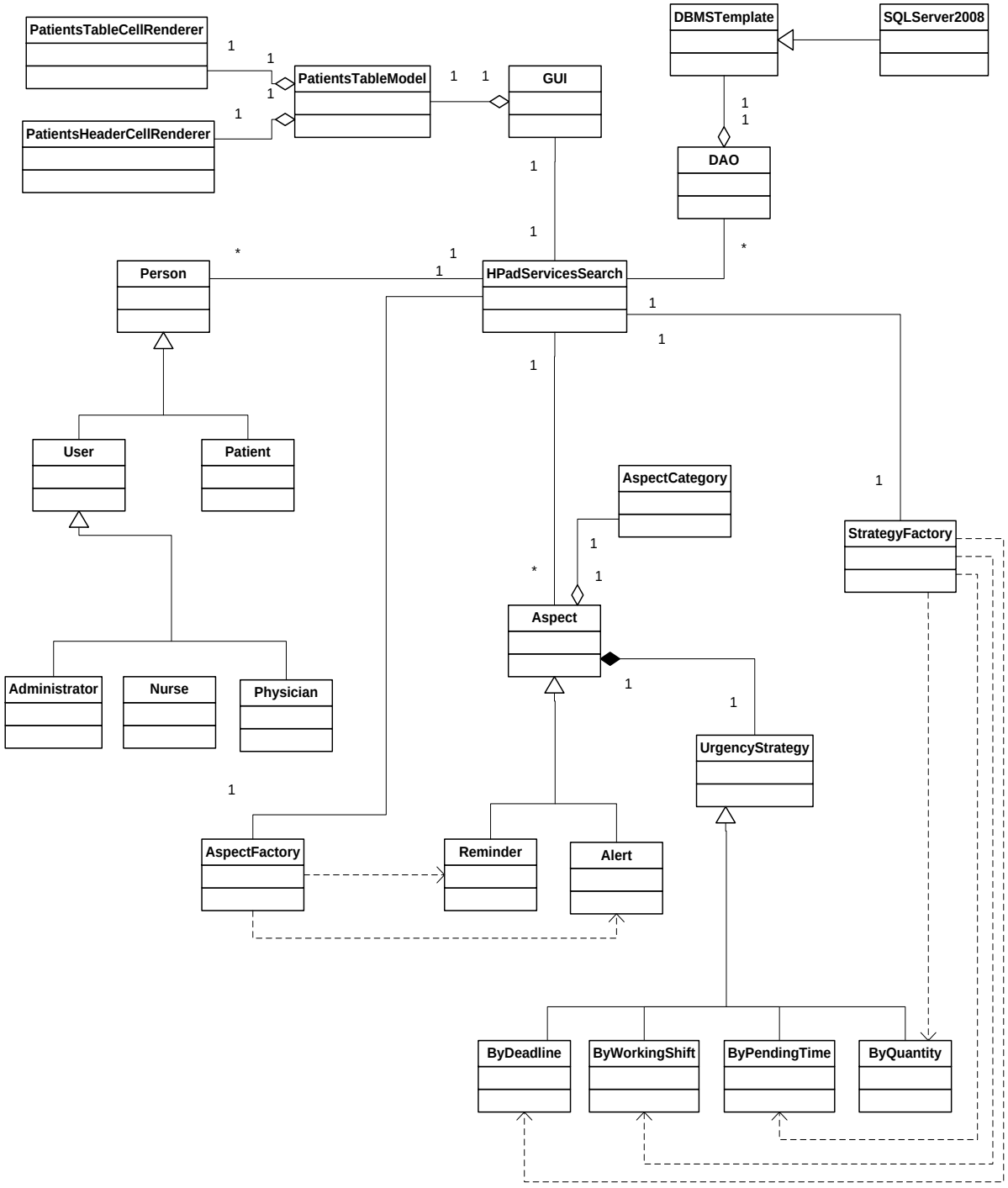

Figure B-1 Simplified UML Class Diagram of the Redesigned ARM

APPENDIX C

HPad System 2009

Monthly Report: May/2009

Patients Charts

Patient:Rosa Del Campo

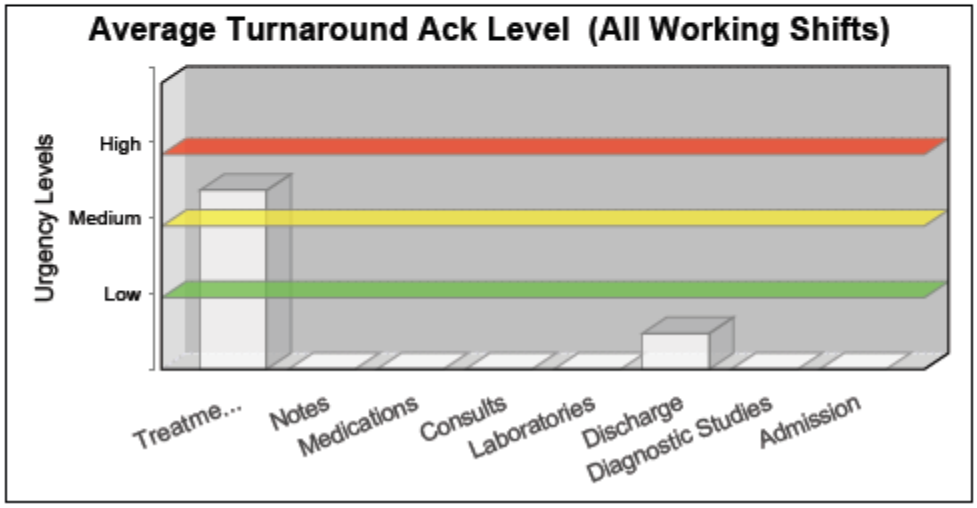

Figure C-1 Example of First Page of a Monthly Report