PEDIATRICS ONE ELECTRONIC MEDICAL DATABASE SYSTEM

By

Erin Elizabeth Prenger

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelors degree
With Honors in

Computer Engineering

THE UNIVERSITY OF ARIZONA

May 2009

Approved by:

_____

Anthony King
College of Engineering (Interdisciplinary Advisor)

STATEMENT BY AUTHOR

I hereby grant to the University of Arizona Library the nonexclusive worldwide right to reproduce and distribute my thesis and abstract (herein, the “licensed materials”), in whole or in part, in any and all media of distribution and in any format in existence now or developed in the future. I represent and warrant to the University of Arizona that the licensed materials are my original work, that I am the sole owner of all rights in and to the licensed materials, and that none of the licensed materials infringe or violate the rights of others. I further represent that I have obtained all necessary rights to permit the University of Arizona Library to reproduce and distribute any nonpublic third party software necessary to access, display, run, or print my thesis. I acknowledge that University of Arizona Library may elect not to distribute my thesis in digital format if, in its reasonable judgment, it believes all such rights have not been secured.

Signed: _____

Roles and Responsibilities

Erin Prenger:

Team Leader – Main point of contact for sponsor and faculty advisors

Database and Back-End Work – For each of the modules developed in the project: Designed and implemented tables in a MySQL database. Designed, implemented, and tested Python code for the back-end to interact with the database, storing and retrieving all data appropriately.

Michael Schreiber:

Middleware and Integration Work – For each of the modules developed in the project: Designed, implemented, and tested the Django middleware to allow the User Interface and the Back-end to communicate with one another. Integrated the UIs and Back-ends, testing each to ensure their complete functionality.

Joseph Wahl:

User Interface – For each of the modules developed in the project: Designed, implemented, and tested the User Interface portion in Adobe Flex.

Abstract

In recent years, there has been an increase in demand within the medical community for software applications to increase the quality of patients' visit and to increase throughput, particularly in small private practices. Such a system would need to collect and store all patient data digitally, as opposed to via traditional paper-based forms. The goal of this project was to fill this need by creating a small, simple-to-use application, whose overall design closely models a patient's trajectory through the office. Specifically, the project is intended for use in small clinics, as opposed to the large hospitals for which most electronic medical systems are designed. Constraints on the implementation primarily consisted of the use of a preexisting MySQL® database for data storage. After examining a series of concepts and potential designs, it was decided to implement a web-based, three-tiered system. These tiers are the user-facing front end written using Adobe® Flex®, the aforementioned backend in MySQL®, and an intermediary layer utilizing Django® to decrease coupling between the two. In addition, modularity was designed into the system by separating each step in a doctor's office into a discrete module in the system. The final prototype produced by the team fulfills all four of the high-level functional requirements agreed upon with the customer at the beginning of the project: reduced potential for billing errors, improved time use by reducing repeated information, improved patient transitions, and improved access to medical knowledge.

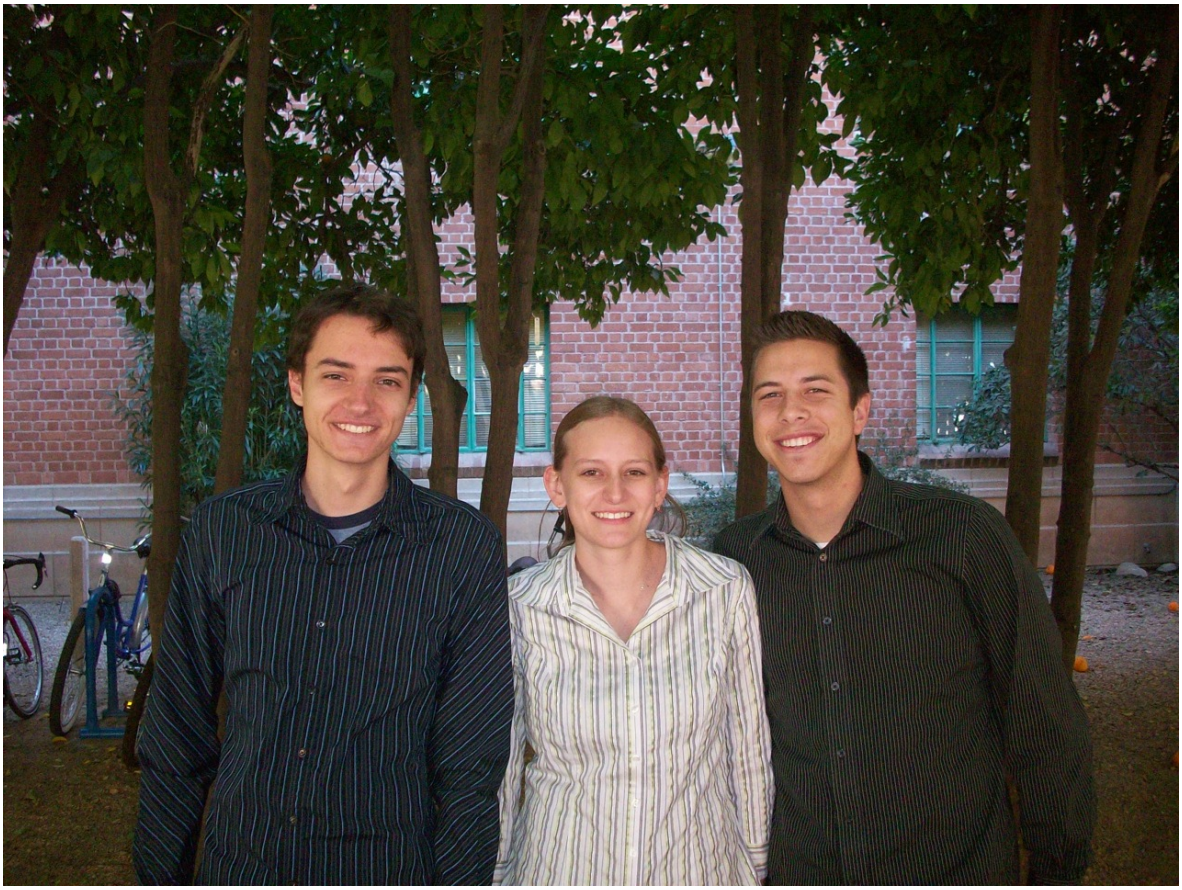
Pediatrics One: Pediatric Electronic Medical Records

Submitted by: Team 3498
Erin Prenger
Michael Schreiber
Joseph Wahl

Submitted to: Anthony King (Mentor)
Mark Ginsburg (Sponsor)

Submitted on: April 30, 2009

Interdisciplinary Engineering Design Program
University of Arizona



In recent years, there has been an increase in demand within the medical community for software applications to increase the quality of patients' visit and to increase throughput, particularly in small private practices. Such a system would need to collect and store all patient data digitally, as opposed to via traditional paper-based forms. The goal of this project was to fill this need by creating a small, simple-to-use application, whose overall design closely models a patient's trajectory through the office. Specifically, the project is intended for use in small clinics, as opposed to the large hospitals for which most electronic medical systems are designed. Constraints on the implementation primarily consisted of the use of a preexisting MySQL® database for data storage. After examining a series of concepts and potential designs, it was decided to implement a web-based, three-tiered system. These tiers are the user-facing front end written using Adobe® Flex®, the aforementioned backend in MySQL®, and an intermediary layer utilizing Django® to decrease coupling between the two. In addition, modularity was designed into the system by separating each step in a doctor's office into a discrete module in the system. The final prototype produced by the team fulfills all four of the high-level functional requirements agreed upon with the customer at the beginning of the project: reduced potential for billing errors, improved time use by reducing repeated information, improved patient transitions, and improved access to medical knowledge.

Table of Contents

I. Introduction 1
II. System Requirements 2
a. Functional Requirements 3
b. Technical Requirements 4
III. Design Concepts and Analysis 6
a. Option 1: Standalone Application 6
b. Option 2: Asynchronous JavaScript + XML (AJAX) / PHP 6
c. Selected Option: Adobe® Flex® 6
d. Concept and Decision Analysis 7
IV. Design 10
a. Functional Decomposition 10
b. Subsystem Design 11
c. Design Analysis and Validation 12
V. System Build 14
a. Subsystem Construction 14
b. Subsystem Debugging 14
c. Subsystem Testing 15
d. Subsystem Integration 15
e. Integration and Acceptance Testing 16
VI. Results 17
a. Results of Testing 17
b. Design Verification 17
c. Design Validation 18
d. Data and Research Results 18
e. System Functionality 19
VII. Conclusions 21
VIII. Recommendations 22
IX. Acknowledgements 23
X. References 24
XI. Appendix 25

List of Figures

Figure 1 - Quick Search Sequence Diagram 11
Figure 2 - Nurse Assessment Sequence Diagram 11
Figure 3 - Sick and Well Visit Sequence Diagram 12
Figure 4 – High-Level Technical Design 14

List of Tables

Table 1 - Functional Requirements 3
Table 2 - Systems Requirements 4
Table 3 - House of Quality Quality Function Deployment (QFD) Matrix 5
Table 4 - Design Matrix 10
Table 5 - Table of Ideality 12
Table 6 - Failure Design Effects Analysis 13
Table 7 - Verification Table 18
Table 8 - Implementation Table 20

I. Introduction

The objective of this project is to assist small private practices, particularly those specializing in pediatric health care, in better serving patients by increasing throughput and improving the quality of health care. This project focused on replacing and rearchitecting an existing software system, using an entirely new toolset to improve reliability and to add scalability and extendibility. The previous system, which was developed in response to a recent federal mandate for a nationwide interoperable heath information network, was designed using an inefficient and antiquated methodology, in that it is very difficult to extend or improve. Certain components and features were never completed and, due to the exceptionally poor documentation, cannot be introduced into the current system in a cost-effective manner. Additionally, because it is not modeled very closely after the actual path of a patient through the office, bottlenecks are often introduced that would not otherwise be present. The system that was developed by this project will greatly improve upon the current system's shortcomings and expand on this original design to make it more efficient and reliable. This implementation of the system, while initially custom-developed for Orange Grove Pediatrics in Tucson, AZ, has the potential to be expanded to any number of similar practices nationwide, as well as the ability to be expanded with a series of modular improvements in the future.

II. System Requirements

Based on the specified needs and requests of the customer, the following four requirements were derived as the most pertinent high-level functional requirements for this project.

1. Reduce billing errors
2. Improve use of time
3. Improve patient transitions
4. Improve access to medical knowledge

The goal of the project will be to address each of these functional requirements and implement different software modules to accomplish them based on the general specifications as well as specific features requested by the customer. The following describes in further detail each of these high level functional requirements.

Reduce Billing Errors

One of the main goals of the project was to reduce the number of billing errors that occur within private practice offices. Medical billing codes fall within two categories (Current Procedural Terminology (CPTs) and International Classification of Diseases, 9th Revision, Clinical Modification (ICD9s) Codes), which correspond to different actions taken and diagnoses given during an appointment. The majority of the errors that occur with these billing codes happen for two distinct reasons. The first cause of these errors is due to an inefficient transfer of data between the doctors/nurses and the billers. During an appointment, the doctors and nurses takes notes on a physical piece of paper, listing the details of the appointment. These details are later transcribed by the biller to billing codes, and can often be misrepresented because the handwriting of the doctors and nurses is unclear or the notes themselves are not accurately interpreted. One goal of our project was to transfer all of these paper forms to digital forms. This should eliminate the first problem of illegible notes. Another aspect of the project which helps to address this functional requirement is to auto populate fields whenever possible. By developing these modules that communicate with each other, this manual transfer of data can be eliminated and consequently this functional requirement will be fulfilled.

Improve use of time

Another primary goal of this project was to improve the use of time in the office. The main measure of success of any private practice can be defined as its "throughput," which is the number of patients that can be seen in a given amount of time. One way to facilitate an increase in throughput is to increase the efficiency of time-usage by all employees. In order to fulfill this functional requirement, our group designed several software modules to improve the efficiency of time usage. Another module that helps to fulfill this functional requirement is an automatic data converter and grapher which will take in the nurse measurements, convert them to the appropriate form and graph them automatically, thereby eliminating the time currently spent by the nurses manually graphing this data. This auto-population of data has be carried over to the doctor's evaluation portion of the appointment as well; as described in the previous section, this should not only reduce errors in data transfer but also save time since the data will no longer need to be manually transcribed by the biller. The basic idea of auto population, eliminating

manual paper to digital transfers, and adding "agent" features that allow the software to do the job of the employees wherever possible has been implemented throughout the project to successfully fulfill this requirement.

Improve patient transitions

The third main requirement expressed by the customer is the need to improve patient transitions. In order to improve efficiency and throughput in the doctor's office, it is important that patient transitions through different stages of the appointment happen as smoothly and quickly as possible. Currently in such a hectic work environment, the communication can be lacking, which leads to doctors and nurses being unaware of the current stage of patients, and consequently patients will often wait at a given stage in an appointment simply because the next employee (doctor or nurse) does not know that they are ready to be seen. To fulfill this requirement and help improve these transitions, the customer has requested that the team implement several features. These features include a module that acts as a patient status bar, which will monitor the location of all patients in the office at any given time. Additionally this module should incorporate alert features to notify the office users (doctors, nurses, front office clerks, and billers) when a patient has entered a new stage relevant to their particular duties.

Improve access to medical knowledge

The final main requirement for the project, as directed by the sponsor, is to incorporate the idea of a nationwide interoperable health information network. As it stands, some of these networks are currently available and in use in larger hospitals and medical facilities; however, there is a new federal mandate for this network to incorporate private practices as well. So, one important requirement of this project was to incorporate this network into the implemented system. For our purposes this occurs primarily in one form: the system interfaces with the pediatric subset of the Physician's Information and Educational Resource, also known as PIER. This database of information can provide the doctors with a large amount of data about different diagnoses, their corresponding medication, and possible side effects of these medications. All of this information networking will provide the doctors with as much background information as possible so that they can make the most accurate, well-informed diagnoses which will lead to a better quality of health care overall.

a. Functional Requirements

Table 1 - Functional Requirements

Functional Requirement	Priority
Reduce billing errors	5
Improve use of time	5
Improve patient transitions	5
Improve access to medical knowledge	5

b. Technical Requirements

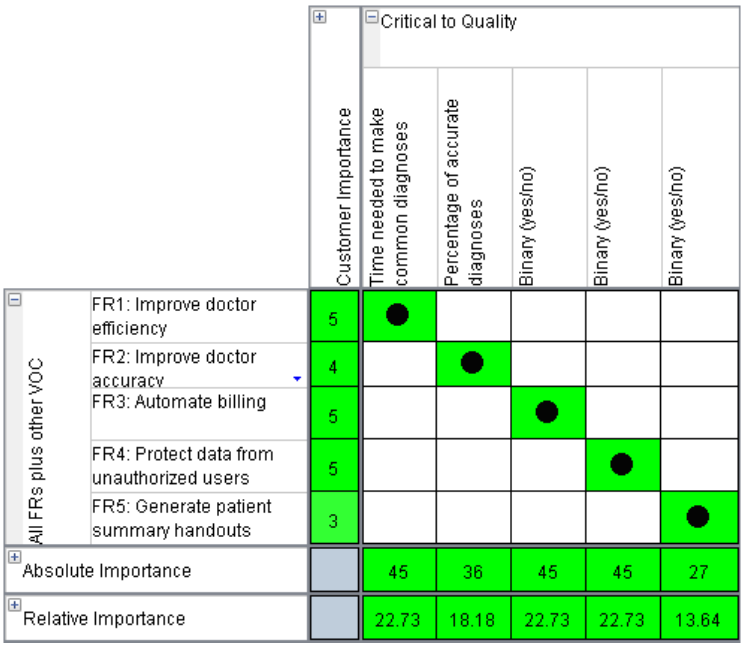
The system that has been developed will be used to allow doctors to input and retrieve patient information in a very quick and efficient manner through a web-based portal. It will also be used to assist in the record keeping and diagnosis of patient illnesses through the integration of the Pediatric subset of the Physicians' Information and Educational Resource (PIER). In order to satisfy the needs of the doctors, a set of system requirements were generated. Since the solution being developed is entirely software based, there are virtually no physical components that can be measured. Instead, the system requirements focus around the functionality of different aspects of our system and across multiple platforms. The system requirements along with constraints dictated by the customer have been described in detail in Table 2: System Requirements.

Table 2 - Systems Requirements

Number	***Category***	***Description***	***Type of Requirement***
1	Performance	Simplify patient information access by implementing a Quick Search feature which searches through the patient database using 5 different search field types	Measurable Term
2	Performance	Generate patient summary handouts indexed by the age of the patient	Measurable Term
3	Performance	Restrict access for different types of users to incorporate various administration and access levels (e.g. doctors, nurses etc)	Measurable Term
4	Performance	Aggregate reporting for all types of users such that information gathered from patients will be maintained in a common location to prevent the need for repeated questions.	Measurable Term
5	Performance	Interface with ASIIS (Arizona State Immunization Information System) such that information about any necessary immunizations related to the patients' age is readily available.	Measurable Term
6	Performance	Improve doctor efficiency by presenting possible diagnoses based on inputted symptoms	Measurable Term
7	Performance	Integrate with ACP (American College of Physicians) Pediatric subset of PIER to collect and present common information about a given diagnosis	Measurable Term

8	Performance	Interface with E-Billing Clearing House to automate the patient billing system	Measurable Term
9	Performance	Access the system via multiple web browsers, including Mozilla® Firefox® 3.x	Measurable Term
10	Platform and tools	Use the Flex® framework for the User Interface.	Constraint
11	Platform and tools	Interface with a MySQL® database	Constraint
12	Platform and tools	Use Tomcat® or Django®	Constraint
13	Platform and tools	Use JavaServer Pages (JSP) to interface with application server	Constraint

Table 3 - House of Quality Quality Function Deployment (QFD) Matrix



III. Design Concepts and Analysis

a. Option 1: Standalone Application

The first option considered when evaluating the architecture of the system featured a standalone client application that would interact with the remote server. As all of the developers on the team were most comfortable with this model of programming, it initially seemed like an obvious choice. A thin client program would be installed on all computers that will connect to the system, and all database interactions would be performed through this application. Indeed, this method of development is still heavily used internally in many corporations for a variety of tasks.

However, preliminary investigation indicated that this would not fill all of our system's requirements. In particular, most standalone applications are not cross-platform, and will only function on the system for which they were written (usually Microsoft® Windows, due to its overwhelming majority of the market). Additionally, these programs require a separate installation and update stream, which can often lead to a mix of older and newer versions across different workstations.

b. Option 2: Asynchronous JavaScript + XML (AJAX) / PHP

The second option considered was much closer to our final chosen design. Once a native application was scrapped, the project was transitioned to an entirely browser-based interface to the system. The traditional way of developing web-based applications (or "web apps") is to write the entire suite using a collection of technologies known as AJAX (short for "Asynchronous JavaScript and XML"). When coupled with the scripting language PHP (not actually an acronym!), it is possible to create dynamic and powerful web apps that are hosted at a central location and accessed by any web browser. This resolves the updating issue presented by standalone applications, as the latest version of the program is automatically downloaded at every access.

Unfortunately, this idea was also ultimately scrapped. Coding in AJAX and PHP is tedious and time-consuming, and requires enormous amounts of code to develop a system of any complexity. Using these technologies also results in considerably more time testing and debugging, as different browsers render JavaScript in very different ways (as an example, many web sites have completely disparate code bases for Microsoft® Internet Explorer and Mozilla® Firefox®, due to renderer inconsistencies).

c. Selected Option: Adobe® Flex®

The third and final option that was explored when trying to decide upon a framework in which to build our web-based application was Adobe® Flex®. Flex® allows for the development of rich web-based applications that eliminate the need for frequent Hypertext Transfer Protocol (HTTP) requests creating a "smoother" feel for the end-user. The system was architected by Adobe® Systems and the integrated development environment, or IDE, to develop in Flex® is based on the Eclipse platform.

An advantage of Flex® over the other two options is that it works with any browsers that support Flash, a ubiquitous plug-in. In addition, the interface builder provided by this IDE provides a drag-and-drop UI builder that assists in rapid prototyping of the interface. Flex® is also free to deploy for both the developer and the end user, unlike some of the previous choices. All they would need to do is ensure that the plug-in is installed and they should not have any complications accessing the system.

In addition to these positives, Flex® 10 has true cross-platform support, working properly on all three major operating systems: Microsoft® Windows®, Mac OS X® and GNU/Linux™. In addition, from a development standpoint, Flex® provides built in support for Extensible Markup Language (XML) and ECMAScript, both of which are industry standards.

Although it has many positives, Flex® is not without its drawbacks. One of these is the fact that the speed at which the program runs is based entirely upon the browser rendering speed of the user. This may lead to issues if an older browser is being used with a less powerful Flash renderer. In addition, due to the fact that Flex® is able to create very powerful components that are able to display lots of information, there may be significant loading time for the user due to how memory intensive the rendering of these "heavy" components are.

d. Concept and Decision Analysis

As described in the previous section, the decision to present the application as a web-based application is a fundamental constraint of the design. Therefore the first decision to be made is the toolkit to be used. Remember: the previous iteration made the mistake of developing a UI based solely around hand-crafted JavaScript, HTML, and AJAX, which greatly crippled the system's ability to expand in the future. At this point the available toolkits have been narrowed down to Microsoft® Silverlight™ and Adobe® Flex®. Both are powerful, versatile tools that can be used on multiple platforms and in multiple web browsers. Unfortunately, both are relatively "heavy;" that is, both sacrifice loading time in favor of offering a rich, desktop-like user interface. When neither particularly surpassed the other in this regard, certain minor issues needed to be taken into account. In particular, deployment presented a particularly difficult issue to overcome. The current system utilizes a Linux™ server (Community ENTerprise Operating System (CentOS) in particular), which is unable to host the Silverlight™ application. Instead, the customer would be forced to install copies of Windows Server 2003®, which ranges in cost from $999 to $3,999 per server. This was considered an unacceptable cost, particularly when compared to the cost of deploying Flex® (or rather, lack thereof: Flex® is completely free to license and use). Compound this with the fact that the client portion of Silverlight™ is not fully supported outside Windows® and Mac OS® X, and it becomes clear that Microsoft's solution is not adequate for this situation.

As mentioned earlier, the particular database to be used is a constraint of the project; the system must make use of a preexisting MySQL® database for all patient information storage. Flex® in its purest form is unable to communicate directly with MySQL®, and requires one or more intermediary carriers to assist in communication. The most common means of performing this interaction is using PHP code, which can be generated and parsed by Flex®, and by which databases can be queried. This is not a perfect solution, however, as PHP code tends to be very difficult to read, write, and modify over time. This approach would also require a daunting amount of initial work, as each piece of PHP needs to be custom-written. In lieu of this, a

number of third-party alternatives exist which are specifically designed to ease interaction with these databases. Many of them offer interfaces in Python, which is a powerful and flexible scripting language with pervasive use throughout industry. By eliminating the need to generate custom PHP code, more time can be spent focusing on actual system design and implementation, and less on code churn.

This leaves only the overall structure of the system. The original implementation was monolithic, without any considerable separation between modules or layers, resulting in a nightmare of interdependent code that is virtually impossible to maintain on a long-term basis. It is clear that this all-in-one design choice is not an option in this iteration. Beyond this, no other structures were considered and discarded.

After careful consideration, it has been determined that the most effective toolkit for the implementation of this system's user interface is Adobe® Flex®. This was determined after eliminating both AJAX and Microsoft® Silverlight™ as potential frameworks for the interface. In reality, Flex® is simply a convenient XML- and ActionScript™-based interface for creating interactive Flash animations in the form of event-based applications. Because Flex® is based directly on Adobe® Flash®, it can be used on any web browser, for any device, as long as it sports a free Flash plug-in. As Adobe® offers a version of the plug-in for virtually every browser on virtually every platform the end-user will encounter, any Flex® application will be almost universally compatible from day one. In fact, these rich interfaces will be even more cross-platform compatible than standard AJAX code: different web browsers on different OSs render HTML and JavaScript differently, requiring the use of many platform-specific hacks and tweaks to ensure a consistent experience. Because Flash (and Flex®, by proxy) is rendered by the plug-in, these tweaks are completely unnecessary, resulting in faster development, with considerably less room for error.

Once Flex® and MySQL® have been put in place, the issue of communication between the two once again approaches the foreground. Fortunately, a number of third-party intermediaries exist in order to avoid the need for cumbersome and repetitive PHP or JSP pages. Investigations into this field consistently point to a single tool: Django®. Django® is a powerful framework that is written entirely in Python, and which has the ability to communicate pragmatically with a MySQL® database. With this tool, queries that may have taken many lines of obfuscated SQL syntax can be performed via a simple function call. Django® is also free to use, which is very convenient.

These three components (Flex®, MySQL®, and Django®) set the groundwork for a three-tiered application, with a rich front end, a Flex®ible middleware, and a powerful database. In this new three-tiered setup, the end user (usually a nurse, doctor, or secretary) interacts directly with the Flex® User Interface (UI), which sends messages across the web to a server. These messages are then intercepted by a Django® gateway, parsed into Python method calls, and translated to SQL queries. The results of these queries climb back up this chain, resulting in a new state being seamlessly presented to the user.

Once the overall architecture has been determined, the final design decision is the general layout of the system. Unlike the previous monolithic attempt at this problem back in 2006, our team will be implementing a highly componentized setup, with clearly defined responsibilities and duties being distributed throughout the modules. Thanks to Django®'s encapsulation of

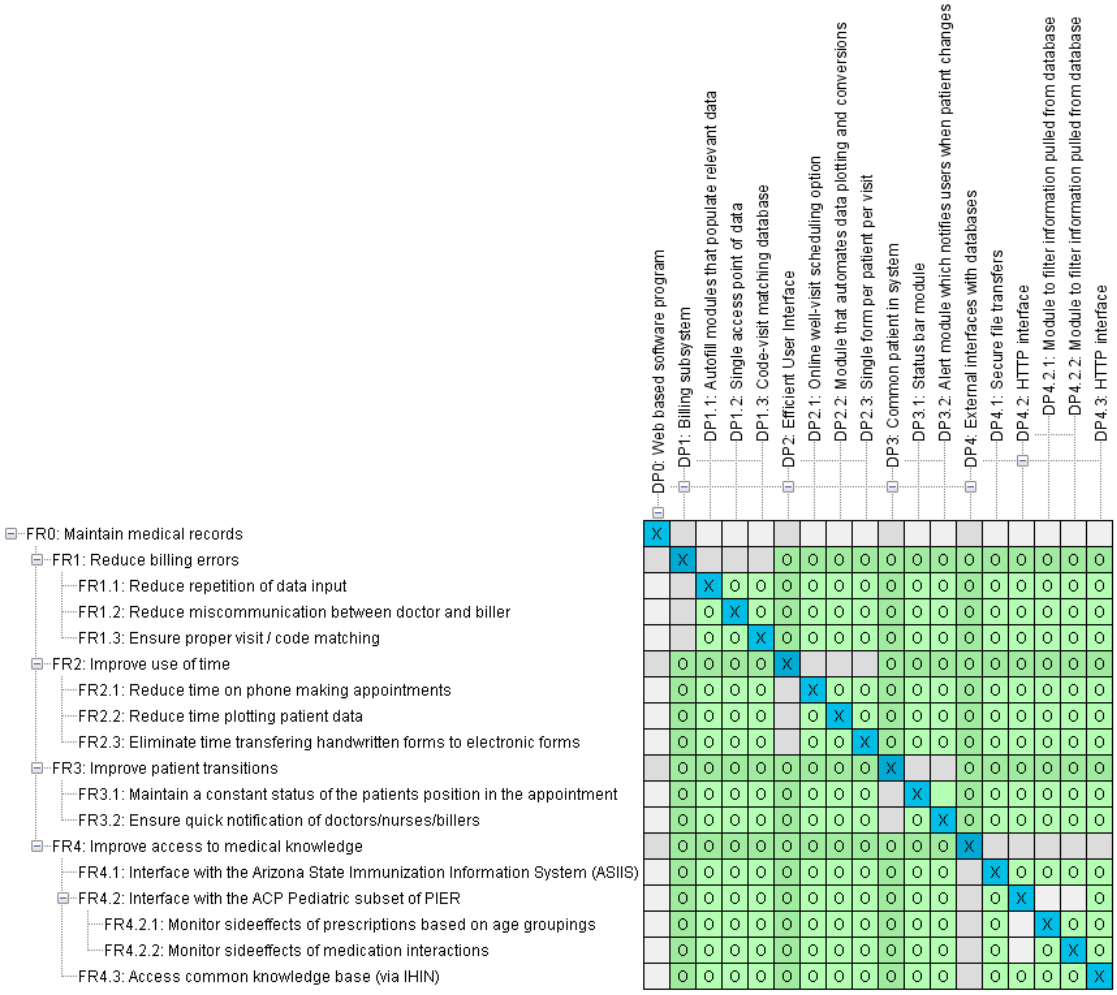
complex database calls, each component in the system can perform its own database queries without large amounts of repeated code. Each module is roughly based on a stage in a patient's trajectory through a clinic (check in, assessment, etc.). This modularization will ensure that the system is expandable, maintainable, and flexible for many years to come, without sacrificing features or simplicity of design and development. The fact that the system is modeled after a patient's trajectory should greatly improve the efficiency with which the system operates, as it will be more closely tied into the actual path of a patient through the process.

IV. Design

a. Functional Decomposition

The following table and design matrix lay out the division of the main functional requirements into smaller, more specific functional requirements and associate a design parameter with each of these requirements. The original description of these requirements lies in detail in the Systems Requirements section of this document; these tables simply enumerate the specifics of these descriptions.

Table 4 - Design Matrix



b. Subsystem Design

In order to meet these functional requirements, the system is divided into four modules. The first of these is the Quick Search module, which is responsible for responding to user queries by providing live updates pulled directly from the database. These queries may take the form of names, birthdates, or unique patient ID numbers. Once the desired patient has been selected, a portal is displayed which binds the other modules to one another.

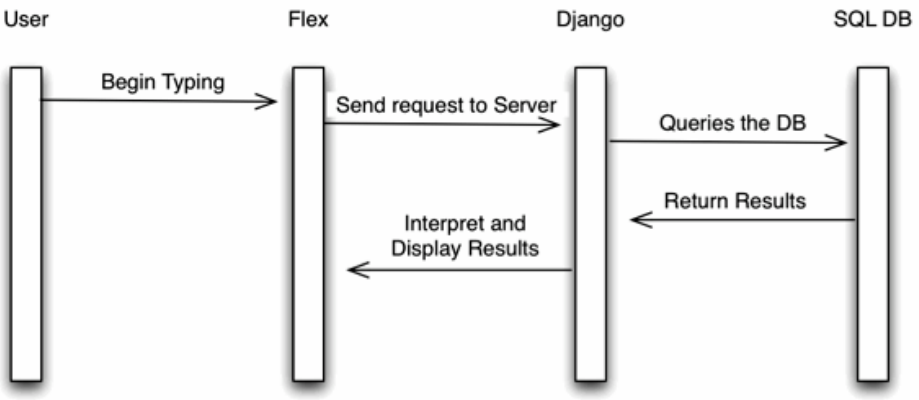


Figure 1 - Quick Search Sequence Diagram

The second module is the Nurse Assessment module. This portion of the system is used by a nurse at the beginning of every visit. It is responsible for storing standard measurements (height, weight), vitals (blood pressure, pulse), and other general questions (contact information, current medications, etc.). Notably, this module provides live graphical updates of a patient's measurements as a function of time, and compares them against official curves provided by the Center for Disease Control (CDC).

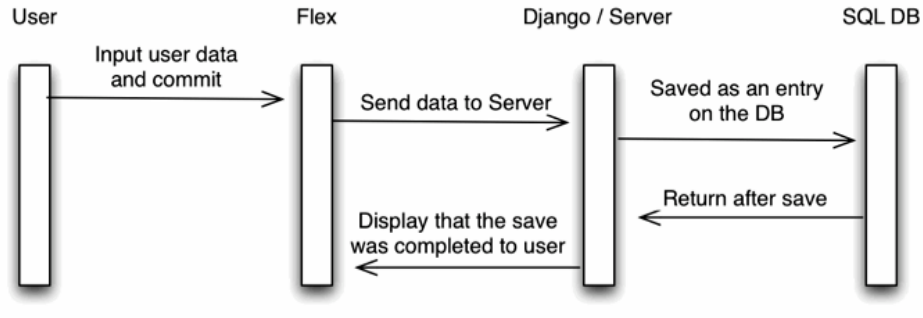


Figure 2 - Nurse Assessment Sequence Diagram

The third and fourth modules for this implementation are the closely-related Sick Visit and Well Visit. The doctor, depending on the reason for the visit, uses one of these modules per visit.

Sick Visit is responsible for helping doctors record symptoms and diagnose ailments, while Well Visit provides a dynamic list of common questions for the patient's age group. However, the two are fundamentally similar from an architectural point of view.

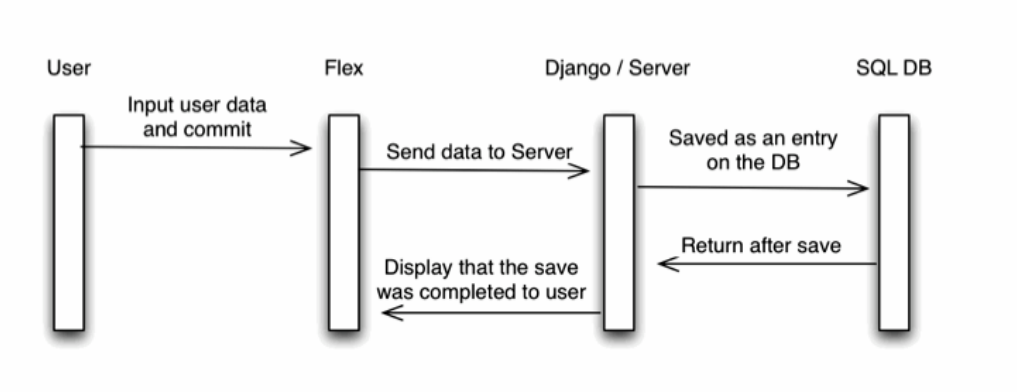


Figure 3 - Sick and Well Visit Sequence Diagram

c. Design Analysis and Validation

Table 5 - Table of Ideality

Requirement	Ideal Final Result	Our Design	Ideality (0-100)
Billing Accuracy	• No billing code errors • No repeat input of data	• Automate billing codes • Electronic forms	100
Time efficiency	• No time spent scheduling well-visits via phone • No manual data plotting	• Well-visits can be scheduled online • Plots generated automatically	80
Patient Transitions	• Move patients between stations immediately	• Indicator for patient progress	90
Access to medical knowledge	• Unified access point to online databases of medical information	• Interfaces with PIER	90
Overall Ideality:			**90**

Table 6 - Failure Design Effects Analysis

Line #	Item	Function	Potential Failure Mode	Effect	Potential Cause	Current Control
1	Logon	Restricts system access to properly authenticated users	User cannot log into system	No ability to access data	Cannot connect to database	Defaults to limited mode in absence of a secure connection
2					User is not registered on the system	Admins can add/remove users as necessary
3	Quick Patient Search	References the database to find data for a given patient	Patient cannot be located	Query is inconclusive	Typo exists in name	Current matches are supplied interactively while typing
4						Top ten suggestions are identified and supplied
5					Supplied name is not a current patient	System indicates that the specified patient cannot be located
6	Measurements	Converts standard measurement input to metric, and stores it in the database	Data cannot be converted to metric	User is unable to record measurements	Invalid input (e.g. letters in a numerical field)	System will parse input and attempt to extract relevant data
7						A descriptive error message will be presented
8	Plotting	Reads from the database, presents measurement history as a graph over time	Curves are not presented correctly	Accurate evaluations cannot be made	Data was input incorrectly initially	When measurements are inputted, the user is presented with the updated curves immediately
9						Recently added data can be removed or modified if incorrect
10					Wrong CDC plot is being used	CDC plot choice will be made automatically, avoiding human error

V. System Build

a. Subsystem Construction

This project was broken down into four subsystems, each of which contains three distinct parts. The projects main modules are Quick Search, Nurse Assessment, Sick Visit, and Well Visit. Each of these modules also is composed of three separate sections. The user interface (front end) was completed in Adobe® Flex®, and is supported by a thoroughly documented MySQL® database (backend). In order to allow these two technologies to communicate with each other, a Django®-based middleware was used. This addition allows for the user-facing portion and database to be built independently, without the need for hardwired calls between the two. By adding this additional level of abstraction to the project, should the system ever need to be modified in the future, any changes will be much easier, as opposed to requiring the entire system to be rearchitected a third time. A graphical representation of the way that the three pieces interact is shown in Figure 4.

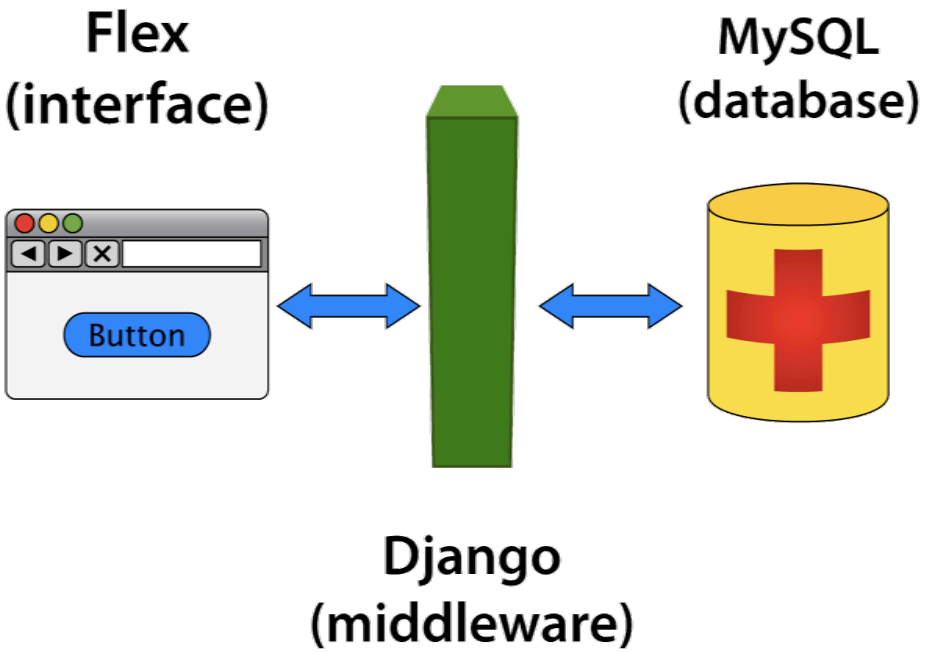


Figure 4 – High-Level Technical Design

b. Subsystem Debugging

Debugging takes place in two discrete segments: client-side and server-side. For all client debugging, the Flex® development environment provides a powerful debugger, which is functionally equivalent to the industry-standard GNU Project Debugger (GDB) for debugging C/C++ code, or JDB for Java. Server-side debugging can be considerably more difficult, though possible. Our team made use of the Python debugger PDB, for discovering flaws and errors in the code.

The most difficult portion to debug in any networked application tends to be the interaction between the client and server. The system must elegantly handle network latency and interruptions, even while debugging, making it difficult to definitively locate the source of certain problems. Often times, it would be necessary to debug both the client and the server simultaneously to identify code flaws.

c. Subsystem Testing

Due to the fact that there exist two sets of sub-systems, namely the four different modules each containing a three-layered architecture, the subsystem testing can be divided up into two sections. While it is possible to test all three portions independently, the Flex® interface is by far the easiest of the three to test. It can be run and tested for general user functionality and to ensure that when a particular series of events is generated (whether they be mouse- or keyboard-driven), an expected outcome is produced. The database backend does not require any initial testing as it is simply a database that will house the newly input data as well as the archived data for each patient. It will only be fully tested once the three pieces are integrated and integration testing is performed. The middleware is primarily tested via its integration with the UI, though some basic testing can be performed via a command-line interface.

Moving on to the four separate modules, each one is created independent of the others in a parallel fashion. They all contain a Flex® UI, MySQL® backend and Django® middleware components. Each individual module can be completely built and tested to ensure complete functionality before any integration is performed. Each module has been checked to ensure that it can save new information to the database, retrieve archival information from the database and edit said archival information and resave the edits. In addition, each of these modules has been tested for aesthetics and usability by the end-user.

d. Subsystem Integration

With regards to integrating the subsystems, there are two distinct types of integration that must occur. First the three technical elements that are being used must be integrated to work with one another to advance all of the modules to fully functional integrated subunits. Then each of these designed modules must be merged into one complete, usable system.

In order to integrate the different technical aspects, the main disconnect that must be implemented properly is the connection between the Flex® User Interface and the MySQL® database. These two technical aspects must be merged so that the user-inputted information is properly stored in the database and in return the database information can be displayed back to the user. This integration will be implemented with the use of the Django® middleware tool. This technical tool acts as an effective communication device between the Flex® front end and the MySQL® back end, allowing the two portions to successfully merge into one fully-functional system.

After the technical aspects have been integrated such that each of the individual modules have been formed and function independently, they must then be combined into one complete application. There are two main points of access that will be used to integrate the system such that the users may navigate through the different sections appropriately. The primary point of access for the system will be a patient portal. This module will employ the functionality of the Quick Search feature that the team developed in order to allow the user to search for a patient. It will then display all of the information contained in the database relevant to this patient including

their prior visits and any potential future visits. Consequently this will provide access to all the other modules in the system and allow the user to navigate through this complete system based on the desired patient. The other point of access for this system will be a Scheduler module that will display all of the information in the system based on date and time. This will also allow the users to navigate through all of the different modules (providing them access to past and future visits) only based on a given time frame and set of appointments instead of a specified patient. Both of these modules will sufficiently integrate all of the subsystems developed for this project and allow the user to navigate through the entire system in a successful manner. However, both of these integration modules are outside of the scope of this particular project; our sponsor only required the previously defined modules and claimed responsibility for the two integration modules just described. So with regards to the required contribution, our team will provide all of the overhead necessary in the developed modules for this integration to take place, and provide temporary placeholders for this integration, which will fulfill our required contribution to the integration portion of the project.

e. Integration and Acceptance Testing

The final integration and acceptance testing of this project was completed in two stages. First the team ran sample scenarios for all possible functionality of the system, ensuring that all of the expected outcomes are achieved consistently. The second level of testing was completed by presenting the finalized system to the medical team at the Orange Grove Pediatrics center. The staff performed trial runs in a real-world office situation and provides feedback and final approval for the overall subsystem integration of the complete final system.

VI. Results

a. Results of Testing

Three layers of testing were employed in order to fully test the results of this system. The first level of testing to be completed tested the functionality of all of the individual components of the system. After each module, each piece of medical data related to that module was tested to ensure that it could be both saved to the database and retrieved from the database successfully. The second level of testing that occurred within each of the modules of the system was scenario testing. Scenario testing involved running through a set of standard scenarios on the system and confirming that in each of the scenarios, the system acted as expected at all times. The final level of testing that our project endured was a real-world application test in which each of our modules was presented to the medical team at the Orange Grove Pediatrics Medical Center to be reviewed by the medical staff at this private practice. The medical staff was able to test the provided prototype in a use the field, provide us with feedback on the system, and finally given an overall approval for completion of each of the modules. The results of all three layers of testing are displayed in the following Data and Research Results section of this document.

b. Design Verification

Design verification was performed to ensure that the medical database system that was created met the requirements set out by the customer. The four high level functional requirements are listed below along with the method of testing that was implemented in order to verify the design. The main testing methods used included unit and integration testing along with module testing with the current users of the PedOne system at the Orange Grove Pediatrics Medical Center.

Table 7 - Verification Table

Requirement	Method of Verification Testing
Billing Accuracy	The current system requires multiple fields to be input in many different places. By auto-filling these fields, the potential for error will be decreased. This will be tested through observation and user testing of the new system.
Time efficiency	To improve time use in the system, graphing of patient data will be done automatically, as opposed to the current method of plotting it by hand. Unit testing of the module and user testing will perform the accuracy and efficiency of this.
Patient Transitions	In order to ensure that a users position can be tracked in the system, a patient tracking system will be written. For a given patient, it will track what states have been completed allowing that office staff to know what the next state for a patient will be. This will be tested using unit, integration and user testing. In addition to this form of tracking, each module will display an indicator as to which sub-sections of that module have been completed. Once again, unit and user testing will be performed.
Access to medical knowledge	To allow users to have better access to medical knowledge, the pediatric subset of PIER will be implemented. This will be unit tested to ensure that for a given condition, the PIER suggestion becomes available to the user. This will also be tested via user testing.

c. Design Validation

Design validation was performed in order to ensure that our design successfully accomplished the goals set out in our functional requirements. Specifically, this validated that the developed PedOne system improves billing accuracy, decreases superfluous time plotting data by hand, increases patient throughput through patient tracking, and increases the access to medical knowledge stored in the pediatric subset of PIER. These four key areas were compared against the old system and their functionality was tested individually through unit testing and collectively through integration testing. In addition, final validation was received from our sponsor and the team at the Orange Grove Pediatrics Center who tested out the final product first hand.

d. Data and Research Results

As previously described, three levels of testing were employed in order to ensure the functionality of this product both on a modular level and as a whole. The results of each of these levels will be displayed in this section. For the first level, each element of data that is associated with each module was tested to ensure that it could successfully be saved to and retrieved from the database through all the sublevels of the system. This testing was completed via confirmation by the testers, the results of which can be seen in a table located in the Appendix A. The second level of testing was scenario testing for each of the individual models as well as the integration of the entire system. A table of the res ults of this scenario testing is located in

Appendix A. The third level of testing was completed as described by providing prototypes of the software to the Orange Grove Pediatrics Medical Center, who, via our sponsor, provided us with a final level of approval for each of the modules.

e. System Functionality

The PedOne system that was developed in four separate modules that addresses specific sections that a patient may go through in a given visit. The four main modules that were developed are Quick Search, Nurse Assessment and Grapher, Sick Visit, and Well Visit. The Quick Search provides a unified access point to access a given patients medical information using either their name, date of birth or patient identification number (PID). This ensures that all members of the office staff can quickly retrieve a patient's information and access their medical history through a dynamic query of the patient database. The second module created was the Nurse Assessment and Grapher. This module is used by nurses to gather preliminary information about patients. This information is automatically converted from US to metric measurements, and height, weight and body mass index (BMI) are automatically graphed against CDC charts. The Sick Visit was created to be used by doctors to track the symptoms of a patient who comes in with an ailment. In addition, an interactive diagram is provided, which allows doctors to annotate specific regions of concern on a human body diagram. This will allow the doctors to visually represent concerns such as a rash on a patient's body or a mole that has recently appeared. In addition, if applicable, data from the pediatric subset of PIER will appear based upon the patient's symptoms. Finally, the Well Visit is provided to allow doctors to perform routine checkups on patients. This module has a common list of questions applicable to all patients, and it will also dynamically generate specific questions geared to a given patient's sex and age.

These four main modules were collectively integrated into a unified access point. This provides a clean user interface where all modules can be accessed and data can be retrieved for a given patient. In addition, a scheduling module provided by our sponsor was integrated into this main window to allow a pediatric private practice to track when patients would arrive and which doctor would be seeing them.

The high level functional requirements were satisfied through the implementation of these four modules and their unified access point through:

Table 8 - Implementation Table

Requirement	How it was implemented
Billing Accuracy	Billing accuracy was improved through the integration of the modules. When this was accomplished, the need to repeatedly input data into different sections of the forms was removed.
Time efficiency	In order to more effectively use the time of the doctors and nurses, the nurse assessment utilized auto-conversion of US to metric data and auto-calculation of BMI. In addition, to reduce possible human errors and to improve the use of time, height, weight and BMI are auto-plotted against CDC data.
Patient Transitions	Patient transitions were improved in two ways. First, well visit, sick visit and nurse assessment all contain sub-modules, which provide visual feedback to the user indicating that they have been completed. In addition, the system as a whole, through its integration, can indicate which of the four main modules have been completed.
Access to medical knowledge	The access of to medical knowledge was improved through the integration of the pediatric subset of PIER into the sick visit module.

VII. Conclusions

The purpose of this project is to develop an electronic medical records system designed to meet the specific needs of pediatric private practices. In recent years, there has developed a strong demand for well-developed electronic medical record management systems within the medical community. Our system attempts to provide an efficient, cost-effective answer to this demand. On a larger scale, the software was developed modeling the patient's trajectory in stages through an appointment. The three-tier architecture previously described (including the Flex® UI, Django® middleware, and MySQL® database) has helped to solve most of the challenges with respect to reliability, scalability, and expandability. Likewise, the system's inherent modularity will help future developers and maintainers who wish to expand or modify the features of the application without being forced to perform yet another costly rewrite.

Of course, this application is by no means a complete system. Several features necessary for a working system, such as a way to handle billing and scheduling appointments, are not yet implemented. However, what we have provided is a solid framework, as well as a capable prototype to demonstrate the feasibility of a small, inexpensive system for small private practices. With the bulk of the most difficult development and testing completed, bringing the application to the point of deployment will be far less time- and resource-consuming.

VIII. Recommendations

Moving forward, this system can be expanded in a few ways. First, the system could be integrated with additional databases to assist in diagnosis of patients and the observation of possible medication interactions. This would allow doctors the ability to make better, more informed diagnosis in a timelier manner. In addition, with the addition of the medication interaction database, a doctor would be able to know in an instant if two medications prescribed to a patient would propose any immediate health risks.

Another possible area for advancement would be with the creation of a user community, in which doctors would be able to collaborate and share information with each other. Due to the fact that many doctors are general practitioners, this would give them a knowledge base to pull from about some less-common diagnoses. In addition, if it were created properly, if a series of particular diagnosis were noticed in a particular region, the doctors would be able to discuss possible causes for such a trend. By giving the doctors a means to communicate with each other and share information, both the practices and the patients benefit.

Finally, we could see a system such as this expanded such that no matter where a patient went or what doctor they saw, the practitioner would have access to their full medical history. This would be beneficial in many instances. The first would be if an unconscious person enters the emergency room. This system would allow the doctors and nurses to know right away what the patient is allergic to and any preexisting conditions that they should know about. In addition to this situation, it would be useful for a person who has more than one doctor. This would allow the doctors to monitor the patient's prescriptions and ensure that multiple doctor are not prescribing conflicting medications.

IX. Acknowledgements

Team 3498 would like to acknowledge our Faculty Mentor for the project, Anthony King, whose assistance throughout our project progression was greatly appreciated.

We would also like to acknowledge the medical team at the Orange Grove Pediatrics center whose doctors, nurses, and front office staff members were greatly helpful in providing input and feedback throughout the development of the product.

Finally, we would like to acknowledge Mark Ginsburg for providing us with sponsorship and constant support throughout the course of the project.

X. References

Adobe, "Adobe Flex 3.3 Language Reference." 26 Feb 2009. Adobe Systems. 25 Apr 2009 <http://livedocs.adobe.com/flex/3/langref/>

Holovaty, Adrian and Jacob Kaplan-Moss. "The Django Book: Version 1.0." 12 Dec 2009. 10 Apr 2009 <http://www.djangobook.com/en/1.0/>.

XI. Appendix

a. Testing Results

Nurse Assessment	Commit Confirmation	Retreive Confirmation
measurement_id	MBS 2/3/09	MBS 2/5/09
patient_id	MBS 2/3/09	MBS 2/5/09
visit_date	MBS 2/3/09	MBS 2/5/09
curtime	MBS 2/3/09	MBS 2/5/09
accompanied_by	MBS 2/3/09	MBS 2/5/09
visit_type	MBS 2/3/09	MBS 2/5/09
complaints	MBS 2/3/09	MBS 2/5/09
medications	MBS 2/3/09	MBS 2/5/09
allergies	MBS 2/3/09	MBS 2/5/09
allergic_reactions	MBS 2/3/09	MBS 2/5/09
bp	MBS 2/3/09	MBS 2/5/09
temp	MBS 2/3/09	MBS 2/5/09
weight_lb	MBS 2/3/09	MBS 2/5/09
weight_oz	MBS 2/3/09	MBS 2/5/09
pulse	MBS 2/3/09	MBS 2/5/09
head_circum_cm	MBS 2/3/09	MBS 2/5/09
height_in	MBS 2/3/09	MBS 2/5/09
tests	MBS 2/3/09	MBS 2/5/09
rapid_result	MBS 2/3/09	MBS 2/5/09
ua_result	MBS 2/3/09	MBS 2/5/09
hgb_result	MBS 2/3/09	MBS 2/5/09
pulseox_result	MBS 2/3/09	MBS 2/5/09
nurse_init	MBS 2/3/09	MBS 2/5/09
last_changed_by	MBS 2/3/09	MBS 2/5/09
last_changed_date	MBS 2/3/09	MBS 2/5/09
contact_tel	MBS 2/3/09	MBS 2/5/09
comments	MBS 2/3/09	MBS 2/5/09

Sick Visit Assessment	Commit Confirmation	Retrieve Confirmation
uri	EEP 3/21/09	EEP 3/22/09
viral_syndrome	EEP 3/21/09	EEP 3/22/09
strep	EEP 3/21/09	EEP 3/22/09
pharying	EEP 3/21/09	EEP 3/22/09
om	EEP 3/21/09	EEP 3/22/09
oe	EEP 3/21/09	EEP 3/22/09
om_oe_left	EEP 3/21/09	EEP 3/22/09
om_oe_right	EEP 3/21/09	EEP 3/22/09
om_oe_both	EEP 3/21/09	EEP 3/22/09
om_oe_perf	EEP 3/21/09	EEP 3/22/09
wheeze	EEP 3/21/09	EEP 3/22/09
asthma	EEP 3/21/09	EEP 3/22/09
c_exacer	EEP 3/21/09	EEP 3/22/09
pneumonia	EEP 3/21/09	EEP 3/22/09
pneumonia_r	EEP 3/21/09	EEP 3/22/09
pneumonia_l	EEP 3/21/09	EEP 3/22/09
pneumonia_ul	EEP 3/21/09	EEP 3/22/09
pneumonia_ml	EEP 3/21/09	EEP 3/22/09
pneumonia_ll	EEP 3/21/09	EEP 3/22/09
pneumonia_peribronch	EEP 3/21/09	EEP 3/22/09
bronchiolitis	EEP 3/21/09	EEP 3/22/09
croup_conjunctivitis	EEP 3/21/09	EEP 3/22/09
allergic_rhinitis	EEP 3/21/09	EEP 3/22/09
conjunct	EEP 3/21/09	EEP 3/22/09
eczema	EEP 3/21/09	EEP 3/22/09
dyshidrotic	EEP 3/21/09	EEP 3/22/09
dermatitis	EEP 3/21/09	EEP 3/22/09
viral_rash	EEP 3/21/09	EEP 3/22/09
warts	EEP 3/21/09	EEP 3/22/09
mollus	EEP 3/21/09	EEP 3/22/09
bite_bug	EEP 3/21/09	EEP 3/22/09
bite_animal	EEP 3/21/09	EEP 3/22/09
bite_human	EEP 3/21/09	EEP 3/22/09
laceration	EEP 3/21/09	EEP 3/22/09
candida	EEP 3/21/09	EEP 3/22/09
candida_oral	EEP 3/21/09	EEP 3/22/09
candida_diap	EEP 3/21/09	EEP 3/22/09
abscess	EEP 3/21/09	EEP 3/22/09
mrsa	EEP 3/21/09	EEP 3/22/09
injury_sprain	EEP 3/21/09	EEP 3/22/09
injury_fracture	EEP 3/21/09	EEP 3/22/09
injury_strain	EEP 3/21/09	EEP 3/22/09
constipation	EEP 3/21/09	EEP 3/22/09
impetigo	EEP 3/21/09	EEP 3/22/09
assessment_desc	EEP 3/21/09	EEP 3/22/09

Sick Visit Drawings	Commit Confirmation	Retrieve Confirmation
body_image_1	JDW 3/21/09	JDW 3/21/09
body_image_2	JDW 3/21/09	JDW 3/21/09
body_image_3	JDW 3/21/09	JDW 3/21/09
body_image_4	JDW 3/21/09	JDW 3/21/09
body_image_5	JDW 3/21/09	JDW 3/21/09
body_image_6	JDW 3/21/09	JDW 3/21/09
body_image_7	JDW 3/21/09	JDW 3/21/09
body_image_8	JDW 3/21/09	JDW 3/21/09
body_image_9	JDW 3/21/09	JDW 3/21/09
body_image_10	JDW 3/21/09	JDW 3/21/09
eyes_image_1	JDW 3/21/09	JDW 3/21/09
eyes_image_2	JDW 3/21/09	JDW 3/21/09
eyes_image_3	JDW 3/21/09	JDW 3/21/09
eyes_image_4	JDW 3/21/09	JDW 3/21/09
eyes_image_5	JDW 3/21/09	JDW 3/21/09
eyes_image_6	JDW 3/21/09	JDW 3/21/09
eyes_image_7	JDW 3/21/09	JDW 3/21/09
eyes_image_8	JDW 3/21/09	JDW 3/21/09
eyes_image_9	JDW 3/21/09	JDW 3/21/09
eyes_image_10	JDW 3/21/09	JDW 3/21/09
ear_l_image_1	JDW 3/21/09	JDW 3/21/09
ear_l_image_2	JDW 3/21/09	JDW 3/21/09
ear_l_image_3	JDW 3/21/09	JDW 3/21/09
ear_l_image_4	JDW 3/21/09	JDW 3/21/09
ear_l_image_5	JDW 3/21/09	JDW 3/21/09
ear_l_image_6	JDW 3/21/09	JDW 3/21/09
ear_l_image_7	JDW 3/21/09	JDW 3/21/09
ear_l_image_8	JDW 3/21/09	JDW 3/21/09
ear_l_image_9	JDW 3/21/09	JDW 3/21/09
ear_l_image_10	JDW 3/21/09	JDW 3/21/09
ear_r_image_1	JDW 3/21/09	JDW 3/21/09
ear_r_image_2	JDW 3/21/09	JDW 3/21/09

ear_r_image_3	JDW 3/21/09	JDW 3/21/09
ear_r_image_4	JDW 3/21/09	JDW 3/21/09
ear_r_image_5	JDW 3/21/09	JDW 3/21/09
ear_r_image_6	JDW 3/21/09	JDW 3/21/09
ear_r_image_7	JDW 3/21/09	JDW 3/21/09
ear_r_image_8	JDW 3/21/09	JDW 3/21/09
ear_r_image_9	JDW 3/21/09	JDW 3/21/09
ear_r_image_10	JDW 3/21/09	JDW 3/21/09
nose_image_1	JDW 3/21/09	JDW 3/21/09
nose_image_2	JDW 3/21/09	JDW 3/21/09
nose_image_3	JDW 3/21/09	JDW 3/21/09
nose_image_4	JDW 3/21/09	JDW 3/21/09
nose_image_5	JDW 3/21/09	JDW 3/21/09
nose_image_6	JDW 3/21/09	JDW 3/21/09
nose_image_7	JDW 3/21/09	JDW 3/21/09
nose_image_8	JDW 3/21/09	JDW 3/21/09
nose_image_9	JDW 3/21/09	JDW 3/21/09
nose_image_10	JDW 3/21/09	JDW 3/21/09
mouth_image_1	JDW 3/21/09	JDW 3/21/09
mouth_image_2	JDW 3/21/09	JDW 3/21/09
mouth_image_3	JDW 3/21/09	JDW 3/21/09
mouth_image_4	JDW 3/21/09	JDW 3/21/09
mouth_image_5	JDW 3/21/09	JDW 3/21/09
mouth_image_6	JDW 3/21/09	JDW 3/21/09
mouth_image_7	JDW 3/21/09	JDW 3/21/09
mouth_image_8	JDW 3/21/09	JDW 3/21/09
mouth_image_9	JDW 3/21/09	JDW 3/21/09
mouth_image_10	JDW 3/21/09	JDW 3/21/09
neck_image_1	JDW 3/21/09	JDW 3/21/09
neck_image_2	JDW 3/21/09	JDW 3/21/09
neck_image_3	JDW 3/21/09	JDW 3/21/09
neck_image_4	JDW 3/21/09	JDW 3/21/09
neck_image_5	JDW 3/21/09	JDW 3/21/09
neck_image_6	JDW 3/21/09	JDW 3/21/09
neck_image_7	JDW 3/21/09	JDW 3/21/09
neck_image_8	JDW 3/21/09	JDW 3/21/09
neck_image_9	JDW 3/21/09	JDW 3/21/09
neck_image_10	JDW 3/21/09	JDW 3/21/09
heart_image_1	JDW 3/21/09	JDW 3/21/09
heart_image_2	JDW 3/21/09	JDW 3/21/09
heart_image_3	JDW 3/21/09	JDW 3/21/09
heart_image_4	JDW 3/21/09	JDW 3/21/09
heart_image_5	JDW 3/21/09	JDW 3/21/09
heart_image_6	JDW 3/21/09	JDW 3/21/09
heart_image_7	JDW 3/21/09	JDW 3/21/09
heart_image_8	JDW 3/21/09	JDW 3/21/09
heart_image_9	JDW 3/21/09	JDW 3/21/09
heart_image_10	JDW 3/21/09	JDW 3/21/09
lungs_image_1	JDW 3/21/09	JDW 3/21/09
lungs_image_2	JDW 3/21/09	JDW 3/21/09
lungs_image_3	JDW 3/21/09	JDW 3/21/09
lungs_image_4	JDW 3/21/09	JDW 3/21/09
lungs_image_5	JDW 3/21/09	JDW 3/21/09
lungs_image_6	JDW 3/21/09	JDW 3/21/09
lungs_image_7	JDW 3/21/09	JDW 3/21/09
lungs_image_8	JDW 3/21/09	JDW 3/21/09
lungs_image_9	JDW 3/21/09	JDW 3/21/09
lungs_image_10	JDW 3/21/09	JDW 3/21/09
abdonmal_image_1	JDW 3/21/09	JDW 3/21/09
abdonmal_image_2	JDW 3/21/09	JDW 3/21/09
abdonmal_image_3	JDW 3/21/09	JDW 3/21/09
abdonmal_image_4	JDW 3/21/09	JDW 3/21/09
abdonmal_image_5	JDW 3/21/09	JDW 3/21/09
abdonmal_image_6	JDW 3/21/09	JDW 3/21/09
abdonmal_image_7	JDW 3/21/09	JDW 3/21/09
abdonmal_image_8	JDW 3/21/09	JDW 3/21/09
abdonmal_image_9	JDW 3/21/09	JDW 3/21/09
abdonmal_image_10	JDW 3/21/09	JDW 3/21/09
gu_image_1	JDW 3/21/09	JDW 3/21/09
gu_image_2	JDW 3/21/09	JDW 3/21/09
gu_image_3	JDW 3/21/09	JDW 3/21/09
gu_image_4	JDW 3/21/09	JDW 3/21/09
gu_image_5	JDW 3/21/09	JDW 3/21/09
gu_image_6	JDW 3/21/09	JDW 3/21/09
gu_image_7	JDW 3/21/09	JDW 3/21/09
gu_image_8	JDW 3/21/09	JDW 3/21/09
gu_image_9	JDW 3/21/09	JDW 3/21/09
gu_image_10	JDW 3/21/09	JDW 3/21/09

Sick Visit Medications	**Commit Confirmation**	**Retrieve Confirmatio**
amox	EEP 3/26/09	EEP 3/26/09
keflex	EEP 3/26/09	EEP 3/26/09

augment	EEP 3/26/09	EEP 3/26/09	xopenex		EEP 3/26/09	EEP 3/26/09
augment_200_5	EEP 3/26/09	EEP 3/26/09	mdi		EEP 3/26/09	EEP 3/26/09
augment_400_5	EEP 3/26/09	EEP 3/26/09	neb		EEP 3/26/09	EEP 3/26/09
augment_500	EEP 3/26/09	EEP 3/26/09	alb_xopenex_mdi_neb_strength		EEP 3/26/09	EEP 3/26/09
augment_875	EEP 3/26/09	EEP 3/26/09	alb_xopenex_mdi_neb_freq		EEP 3/26/09	EEP 3/26/09
augment_tsp_pill	EEP 3/26/09	EEP 3/26/09	alb_xopenex_mdi_neb_continue_pm		EEP 3/26/09	EEP 3/26/09
augment_bid_x_10d	EEP 3/26/09	EEP 3/26/09	alb_xopenex_mdi_neb_continue_untilcoughresolves		EEP 3/26/09	EEP 3/26/09
omnicef	EEP 3/26/09	EEP 3/26/09	home_neb_hasathome		EEP 3/26/09	EEP 3/26/09
omnicef_125_5	EEP 3/26/09	EEP 3/26/09	home_neb_ordered		EEP 3/26/09	EEP 3/26/09
omnicef_250_5	EEP 3/26/09	EEP 3/26/09	home_neb_given		EEP 3/26/09	EEP 3/26/09
omnicef_300	EEP 3/26/09	EEP 3/26/09	prelone		EEP 3/26/09	EEP 3/26/09
omnicef_tsp_pill	EEP 3/26/09	EEP 3/26/09	prednisone		EEP 3/26/09	EEP 3/26/09
omnicef_qd	EEP 3/26/09	EEP 3/26/09	prelone_prednisone_dose		EEP 3/26/09	EEP 3/26/09
omnicef_bid_x_5_10d	EEP 3/26/09	EEP 3/26/09	prelone_prednisone_qdx_days		EEP 3/26/09	EEP 3/26/09
bactrim_ds_susp	EEP 3/26/09	EEP 3/26/09	rhinocort		EEP 3/26/09	EEP 3/26/09
bactrim_ds_susp_pill_tsp	EEP 3/26/09	EEP 3/26/09	veramist		EEP 3/26/09	EEP 3/26/09
bactrim_ds_susp_bid_x_10d	EEP 3/26/09	EEP 3/26/09	flonase		EEP 3/26/09	EEP 3/26/09
ciprodex	EEP 3/26/09	EEP 3/26/09	nasonex		EEP 3/26/09	EEP 3/26/09
floxin_drops	EEP 3/26/09	EEP 3/26/09	rhi_ver_flo_nas_spray		EEP 3/26/09	EEP 3/26/09
ciprox_floxin_bid_x_5_7d	EEP 3/26/09	EEP 3/26/09	rhi_ver_flo_nas_qs		EEP 3/26/09	EEP 3/26/09
polytrim	EEP 3/26/09	EEP 3/26/09	claritin		EEP 3/26/09	EEP 3/26/09
vigamox	EEP 3/26/09	EEP 3/26/09	zyrtec		EEP 3/26/09	EEP 3/26/09
bactroban_oint	EEP 3/26/09	EEP 3/26/09	claritin_zyrtec_5mg		EEP 3/26/09	EEP 3/26/09
bactroban_bid	EEP 3/26/09	EEP 3/26/09	claritin_zyrtec_10mg		EEP 3/26/09	EEP 3/26/09
bactroban_hc_1_2p5	EEP 3/26/09	EEP 3/26/09	claritin_zyrtec_pill		EEP 3/26/09	EEP 3/26/09
bactroban_tac_0p1	EEP 3/26/09	EEP 3/26/09	claritin_zyrtec_syrup		EEP 3/26/09	EEP 3/26/09
bactroban_metrogel_1_bid_pm	EEP 3/26/09	EEP 3/26/09	allegra		EEP 3/26/09	EEP 3/26/09
mag_citrate	EEP 3/26/09	EEP 3/26/09	allegra_30		EEP 3/26/09	EEP 3/26/09
mag_citrate_bottle_bid_2	EEP 3/26/09	EEP 3/26/09	allegra_60		EEP 3/26/09	EEP 3/26/09
miralax	EEP 3/26/09	EEP 3/26/09	allegra_180		EEP 3/26/09	EEP 3/26/09
miralax_oz_day	EEP 3/26/09	EEP 3/26/09	allegra_qd		EEP 3/26/09	EEP 3/26/09
pulmicort	EEP 3/26/09	EEP 3/26/09	allegra_bid		EEP 3/26/09	EEP 3/26/09
pulmicort_inh	EEP 3/26/09	EEP 3/26/09	medications_desc		EEP 3/26/09	EEP 3/26/09
pulmicort_resp	EEP 3/26/09	EEP 3/26/09				
qvar	EEP 3/26/09	EEP 3/26/09	**Sick Visit Objective**		**Commit Confirmation**	**Retrieve Confirmation**
advair	EEP 3/26/09	EEP 3/26/09	patient_id		EEP 3/28/09	JDW 3/28/09
pulmicort_qvar_advair_strength	EEP 3/26/09	EEP 3/26/09	head_nl		EEP 3/28/09	JDW 3/28/09
pulmicort_qvar_advair_strength_vial	EEP 3/26/09	EEP 3/26/09	skin_nl		EEP 3/28/09	JDW 3/28/09
pulmicort_qvar_advair_strength_puff	EEP 3/26/09	EEP 3/26/09	extrem_nl		EEP 3/28/09	JDW 3/28/09
pulmicort_qvar_advair_freq	EEP 3/26/09	EEP 3/26/09	rash_mole_color		EEP 3/28/09	JDW 3/28/09
alb	EEP 3/26/09	EEP 3/26/09	light_dark		EEP 3/28/09	JDW 3/28/09

size	EEP 3/28/09	JDW 3/28/09	eyes_r_cbl	EEP 3/28/09	JDW 3/28/09
papular	EEP 3/28/09	JDW 3/28/09	eyes_r_cbl_severity	EEP 3/28/09	JDW 3/28/09
macular	EEP 3/28/09	JDW 3/28/09	eyes_r_desc	EEP 3/28/09	JDW 3/28/09
crusty	EEP 3/28/09	JDW 3/28/09	ears_l_nl	EEP 3/28/09	JDW 3/28/09
scaley	EEP 3/28/09	JDW 3/28/09	ears_l_canal_erythem	EEP 3/28/09	JDW 3/28/09
diffuse	EEP 3/28/09	JDW 3/28/09	ears_l_canal_erythem_severity	EEP 3/28/09	JDW 3/28/09
scattered	EEP 3/28/09	JDW 3/28/09	ears_l_canal_edema	EEP 3/28/09	JDW 3/28/09
patchy	EEP 3/28/09	JDW 3/28/09	ears_l_canal_edema_severity	EEP 3/28/09	JDW 3/28/09
confluent	EEP 3/28/09	JDW 3/28/09	ears_l_canal_exudate	EEP 3/28/09	JDW 3/28/09
bruised	EEP 3/28/09	JDW 3/28/09	ears_l_canal_exudate_severity	EEP 3/28/09	JDW 3/28/09
edema	EEP 3/28/09	JDW 3/28/09	ears_l_canal_pus	EEP 3/28/09	JDW 3/28/09
bruised_edema_severity	EEP 3/28/09	JDW 3/28/09	ears_l_canal_pus_severity	EEP 3/28/09	JDW 3/28/09
rom_full	EEP 3/28/09	JDW 3/28/09	ears_l_canal_wax	EEP 3/28/09	JDW 3/28/09
rom_limited	EEP 3/28/09	JDW 3/28/09	ears_l_canal_wax_severity	EEP 3/28/09	JDW 3/28/09
rom_desc	EEP 3/28/09	JDW 3/28/09	ears_l_canal_foreignbody	EEP 3/28/09	JDW 3/28/09
tender_severity	EEP 3/28/09	JDW 3/28/09	ears_l_canal_foreignbody_severity	EEP 3/28/09	JDW 3/28/09
tender_desc	EEP 3/28/09	JDW 3/28/09	ears_l_tm_eryth	EEP 3/28/09	JDW 3/28/09
eyes_nl	EEP 3/28/09	JDW 3/28/09	ears_l_tm_eryth_severity	EEP 3/28/09	JDW 3/28/09
eyes_l_inject	EEP 3/28/09	JDW 3/28/09	ears_l_tm_pet	EEP 3/28/09	JDW 3/28/09
eyes_l_inject_severity	EEP 3/28/09	JDW 3/28/09	ears_l_tm_fluid	EEP 3/28/09	JDW 3/28/09
eyes_l_disch_clr	EEP 3/28/09	JDW 3/28/09	ears_l_tm_fluid_desc	EEP 3/28/09	JDW 3/28/09
eyes_l_disch_crust	EEP 3/28/09	JDW 3/28/09	ears_l_tm_obstructed	EEP 3/28/09	JDW 3/28/09
eyes_l_disch_yellow	EEP 3/28/09	JDW 3/28/09	ears_l_tm_bullus	EEP 3/28/09	JDW 3/28/09
eyes_l_disch_green	EEP 3/28/09	JDW 3/28/09	ears_l_desc	EEP 3/28/09	JDW 3/28/09
eyes_l_lids_edema	EEP 3/28/09	JDW 3/28/09	ears_r_nl	EEP 3/28/09	JDW 3/28/09
eyes_l_lids_severity	EEP 3/28/09	JDW 3/28/09	ears_r_canal_erythem	EEP 3/28/09	JDW 3/28/09
eyes_l_dennes	EEP 3/28/09	JDW 3/28/09	ears_r_canal_erythem_severity	EEP 3/28/09	JDW 3/28/09
eyes_l_dennes_severity	EEP 3/28/09	JDW 3/28/09	ears_r_canal_edema	EEP 3/28/09	JDW 3/28/09
eyes_l_cbl	EEP 3/28/09	JDW 3/28/09	ears_r_canal_edema_severity	EEP 3/28/09	JDW 3/28/09
eyes_l_cbl_severity	EEP 3/28/09	JDW 3/28/09	ears_r_canal_exudate	EEP 3/28/09	JDW 3/28/09
eyes_l_desc	EEP 3/28/09	JDW 3/28/09	ears_r_canal_exudate_severity	EEP 3/28/09	JDW 3/28/09
eyes_r_inject	EEP 3/28/09	JDW 3/28/09	ears_r_canal_pus	EEP 3/28/09	JDW 3/28/09
eyes_r_inject_severity	EEP 3/28/09	JDW 3/28/09	ears_r_canal_pus_severity	EEP 3/28/09	JDW 3/28/09
eyes_r_disch_clr	EEP 3/28/09	JDW 3/28/09	ears_r_canal_wax	EEP 3/28/09	JDW 3/28/09
eyes_r_disch_crust	EEP 3/28/09	JDW 3/28/09	ears_r_canal_wax_severity	EEP 3/28/09	JDW 3/28/09
eyes_r_disch_yellow	EEP 3/28/09	JDW 3/28/09	ears_r_canal_foreignbody	EEP 3/28/09	JDW 3/28/09
eyes_r_disch_green	EEP 3/28/09	JDW 3/28/09	ears_r_canal_foreignbody_severity	EEP 3/28/09	JDW 3/28/09
eyes_r_lids_edema	EEP 3/28/09	JDW 3/28/09	ears_r_tm_eryth	EEP 3/28/09	JDW 3/28/09
eyes_r_lids_severity	EEP 3/28/09	JDW 3/28/09	ears_r_tm_eryth_severity	EEP 3/28/09	JDW 3/28/09
eyes_r_lids_dennes	EEP 3/28/09	JDW 3/28/09	ears_r_tm_pet	EEP 3/28/09	JDW 3/28/09
eyes_r_dennes_severity	EEP 3/28/09	JDW 3/28/09	ears_r_tm_fluid	EEP 3/28/09	JDW 3/28/09

ears_r_tm_fluid_desc	EEP 3/28/09	JDW 3/28/09
ears_r_tm_obstructed	EEP 3/28/09	JDW 3/28/09
ears_r_tm_bullus	EEP 3/28/09	JDW 3/28/09
ears_r_desc	EEP 3/28/09	JDW 3/28/09
nose_nl	EEP 3/28/09	JDW 3/28/09
nose_turbs_edema	EEP 3/29/09	JDW 3/29/09
nose_turbs_edema_severity	EEP 3/29/09	JDW 3/29/09
nose_turbs_eryth	EEP 3/29/09	JDW 3/29/09
nose_turbs_eryth_severity	EEP 3/29/09	JDW 3/29/09
nose_turbs_pearly	EEP 3/29/09	JDW 3/29/09
nose_turbs_pearly_severity	EEP 3/29/09	JDW 3/29/09
nose_rr	EEP 3/29/09	JDW 3/29/09
nose_desc	EEP 3/29/09	JDW 3/29/09
mouth_nl	EEP 3/29/09	JDW 3/29/09
mouth_tonsils_one	EEP 3/29/09	JDW 3/29/09
mouth_tonsils_two	EEP 3/29/09	JDW 3/29/09
mouth_tonsils_three	EEP 3/29/09	JDW 3/29/09
mouth_tonsils_four	EEP 3/29/09	JDW 3/29/09
mouth_tonsils_erythem	EEP 3/29/09	JDW 3/29/09
mouth_tonsils_erythem_severity	EEP 3/29/09	JDW 3/29/09
mouth_tonsils_exudate	EEP 3/29/09	JDW 3/29/09
mouth_soft_palate_erythem	EEP 3/29/09	JDW 3/29/09
mouth_soft_palate_erythem_severity	EEP 3/29/09	JDW 3/29/09
mouth_soft_palate_petechiae	EEP 3/29/09	JDW 3/29/09
mouth_post_pharynx_erythem	EEP 3/29/09	JDW 3/29/09
mouth_post_pharynx_erythem_severity	EEP 3/29/09	JDW 3/29/09
mouth_post_pharynx_cobblestoning	EEP 3/29/09	JDW 3/29/09
mouth_desc	EEP 3/29/09	JDW 3/29/09
neck_nl	EEP 3/29/09	JDW 3/29/09
neck_ln_enlarged	EEP 3/29/09	JDW 3/29/09
neck_ln_lessThan1p5	EEP 3/29/09	JDW 3/29/09
neck_ln_cm	EEP 3/29/09	JDW 3/29/09
neck_desc	EEP 3/29/09	JDW 3/29/09
heart_rrr	EEP 3/29/09	JDW 3/29/09
heart_tach_nl_r	EEP 3/29/09	JDW 3/29/09
heart_no_murm	EEP 3/29/09	JDW 3/29/09
heart_desc	EEP 3/29/09	JDW 3/29/09
lungs_nl	EEP 3/29/09	JDW 3/29/09
lungs_decreasing_breathsounds	EEP 3/29/09	JDW 3/29/09
lungs_wheeze	EEP 3/29/09	JDW 3/29/09
lungs_crackle	EEP 3/29/09	JDW 3/29/09
lungs_ae	EEP 3/29/09	JDW 3/29/09
lungs_ae_desc	EEP 3/29/09	JDW 3/29/09
lungs_forced_expiration_nl	EEP 3/29/09	JDW 3/29/09
lungs_forced_expiration_prolonged	EEP 3/29/09	JDW 3/29/09
lungs_forced_expiration_wheeze	EEP 3/29/09	JDW 3/29/09
lungs_desc	EEP 3/29/09	JDW 3/29/09
abdomen_nl	EEP 3/29/09	JDW 3/29/09
abdomen_s	EEP 3/29/09	JDW 3/29/09
abdomen_nd	EEP 3/29/09	JDW 3/29/09
abdomen_no_hsm	EEP 3/29/09	JDW 3/29/09
abdomen_bs_nl	EEP 3/29/09	JDW 3/29/09
abdomen_bs_increasing	EEP 3/29/09	JDW 3/29/09
abdomen_bs_decreasing	EEP 3/29/09	JDW 3/29/09
abdomen_bs_absent	EEP 3/29/09	JDW 3/29/09
abdomen_tender	EEP 3/29/09	JDW 3/29/09
abdomen_desc	EEP 3/29/09	JDW 3/29/09
gu_labia_min	EEP 3/29/09	JDW 3/29/09
gu_labia_maj	EEP 3/29/09	JDW 3/29/09
gu_urethra	EEP 3/29/09	JDW 3/29/09
gu_vagina	EEP 3/29/09	JDW 3/29/09
gu_anus_rect	EEP 3/29/09	JDW 3/29/09
gu_breast_left	EEP 3/29/09	JDW 3/29/09
gu_breast_right	EEP 3/29/09	JDW 3/29/09
gu_breast_both	EEP 3/29/09	JDW 3/29/09
gu_breast_desc	EEP 3/29/09	JDW 3/29/09
gu_penis_nl	EEP 3/29/09	JDW 3/29/09
gu_penis_desc	EEP 3/29/09	JDW 3/29/09
gu_testes	EEP 3/29/09	JDW 3/29/09
gu_tan	EEP 3/29/09	JDW 3/29/09

Plans / Handouts / Follow up	**Commit Confirmation**	**Retrieve Confirmation**
plan_sympt_car	MBS 3/26/09	MBS 3/26/09
plan_tx_provided	MBS 3/26/09	MBS 3/26/09
plan_desc	MBS 3/26/09	MBS 3/26/09
handouts_uri	MBS 3/26/09	MBS 3/26/09
handouts_uri_less2	MBS 3/26/09	MBS 3/26/09
handouts_uri_greater2	MBS 3/26/09	MBS 3/26/09
handouts_wheezing	MBS 3/26/09	MBS 3/26/09
handouts_croup	MBS 3/26/09	MBS 3/26/09
handouts_warts	MBS 3/26/09	MBS 3/26/09
handouts_oe	MBS 3/26/09	MBS 3/26/09

handouts_constipation	MBS 3/26/09	MBS 3/26/09
handouts_ocps	MBS 3/26/09	MBS 3/26/09
handouts_bronchiolitis	MBS 3/26/09	MBS 3/26/09
followup_pm	MBS 3/26/09	MBS 3/26/09
followup_ifworsens	MBS 3/26/09	MBS 3/26/09
followup_ifsymptomschange	MBS 3/26/09	MBS 3/26/09
followup_ifnotresolving	MBS 3/26/09	MBS 3/26/09
followup_desc	MBS 3/26/09	MBS 3/26/09

Sick Visit Subjective	**Commit Confirmation**	**Retrieve Confirmation**
fever	EEP 3/30/09	JDW 3/30/09
activity_nl	EEP 3/30/09	JDW 3/30/09
activity_ok	EEP 3/30/09	JDW 3/30/09
activity_down	EEP 3/30/09	JDW 3/30/09
activity_desc	EEP 3/30/09	JDW 3/30/09
rn	EEP 3/30/09	JDW 3/30/09
congest	EEP 3/30/09	JDW 3/30/09
rn_desc	EEP 3/30/09	JDW 3/30/09
st	EEP 3/30/09	JDW 3/30/09
ha	EEP 3/30/09	JDW 3/30/09
cough_wet	EEP 3/30/09	JDW 3/30/09
cough_dry	EEP 3/30/09	JDW 3/30/09
cough_barky	EEP 3/30/09	JDW 3/30/09
cough_tight	EEP 3/30/09	JDW 3/30/09
cough_day	EEP 3/30/09	JDW 3/30/09
cough_night	EEP 3/30/09	JDW 3/30/09
cough_with_activity	EEP 3/30/09	JDW 3/30/09
cough_desc	EEP 3/30/09	JDW 3/30/09
eyes_red	EEP 3/30/09	JDW 3/30/09
eyes_hurt	EEP 3/30/09	JDW 3/30/09
eyes_itchy	EEP 3/30/09	JDW 3/30/09
eyes_drainage	EEP 3/30/09	JDW 3/30/09
eyes_left	EEP 3/30/09	JDW 3/30/09
eyes_right	EEP 3/30/09	JDW 3/30/09
eyes_both	EEP 3/30/09	JDW 3/30/09
eyes_desc	EEP 3/30/09	JDW 3/30/09
ear_left	EEP 3/30/09	JDW 3/30/09
ear_right	EEP 3/30/09	JDW 3/30/09
ear_both	EEP 3/30/09	JDW 3/30/09
ear_pinna	EEP 3/30/09	JDW 3/30/09
ear_drainage	EEP 3/30/09	JDW 3/30/09
ear_swimming	EEP 3/30/09	JDW 3/30/09
ear_desc	EEP 3/30/09	JDW 3/30/09
stomach_general	EEP 3/30/09	JDW 3/30/09
stomach_ruq	EEP 3/30/09	JDW 3/30/09
stomach_luq	EEP 3/30/09	JDW 3/30/09
stomach_rlq	EEP 3/30/09	JDW 3/30/09
stomach_llq	EEP 3/30/09	JDW 3/30/09
stomach_periumb	EEP 3/30/09	JDW 3/30/09
stomach_epigast	EEP 3/30/09	JDW 3/30/09
stomach_desc	EEP 3/30/09	JDW 3/30/09
appetite_nl	EEP 3/30/09	JDW 3/30/09
appetite_off_and_on	EEP 3/30/09	JDW 3/30/09
appetite_fair	EEP 3/30/09	JDW 3/30/09
appetite_poor	EEP 3/30/09	JDW 3/30/09
stomach_nausea	EEP 3/30/09	JDW 3/30/09
stomach_vomit	EEP 3/30/09	JDW 3/30/09
appetite_desc	EEP 3/30/09	JDW 3/30/09
stool_nl	EEP 3/30/09	JDW 3/30/09
stool_watery	EEP 3/30/09	JDW 3/30/09
stool_loosestool_hard	EEP 3/30/09	JDW 3/30/09
stool_tiny_balls	EEP 3/30/09	JDW 3/30/09
stool_last_time	EEP 3/30/09	JDW 3/30/09
stool_blood	EEP 3/30/09	JDW 3/30/09
stool_no_blood	EEP 3/30/09	JDW 3/30/09
stool_desc	EEP 3/30/09	JDW 3/30/09
urine_burning	EEP 3/30/09	JDW 3/30/09
urine_frequency	EEP 3/30/09	JDW 3/30/09
urine_accidents	EEP 3/30/09	JDW 3/30/09
urine_desc	EEP 3/30/09	JDW 3/30/09
skin_dry	EEP 3/30/09	JDW 3/30/09
skin_itchy	EEP 3/30/09	JDW 3/30/09
skin_rash	EEP 3/30/09	JDW 3/30/09
skin_papular	EEP 3/30/09	JDW 3/30/09
skin_macular	EEP 3/30/09	JDW 3/30/09
skin_desc	EEP 3/30/09	JDW 3/30/09
lesion_size	EEP 3/30/09	JDW 3/30/09
lesion_number	EEP 3/30/09	JDW 3/30/09
menses_nl	EEP 3/30/09	JDW 3/30/09
menses_q	EEP 3/30/09	JDW 3/30/09
menses_d	EEP 3/30/09	JDW 3/30/09
menses_w	EEP 3/30/09	JDW 3/30/09

flow_nl	EEP 3/30/09	JDW 3/30/09
flow_heavy	EEP 3/30/09	JDW 3/30/09
cramping_none	EEP 3/30/09	JDW 3/30/09
cramping_mild	EEP 3/30/09	JDW 3/30/09
cramping_severe	EEP 3/30/09	JDW 3/30/09
cramping_desc	EEP 3/30/09	JDW 3/30/09

Sick Visit Tests/Procedures	Commit Confirmation	Retrieve Confirmation
rs_plus	MBS 4/2/09	JDW 4/2/09
rs_minua	MBS 4/2/09	JDW 4/2/09
rs_cx_sent	MBS 4/2/09	JDW 4/2/09
rs_cx_desc	MBS 4/2/09	JDW 4/2/09
ua	MBS 4/2/09	JDW 4/2/09
ua_cx_sent	MBS 4/2/09	JDW 4/2/09
ua_cx_desc	MBS 4/2/09	JDW 4/2/09
silver_nitrate	MBS 4/2/09	JDW 4/2/09
fluorescein	MBS 4/2/09	JDW 4/2/09
hcg	MBS 4/2/09	JDW 4/2/09
wound	MBS 4/2/09	JDW 4/2/09
wound_cx	MBS 4/2/09	JDW 4/2/09
wound_cx_desc	MBS 4/2/09	JDW 4/2/09
neb_tx_desc	MBS 4/2/09	JDW 4/2/09
alb	MBS 4/2/09	JDW 4/2/09
pulm_half	MBS 4/2/09	JDW 4/2/09
pulm_one	MBS 4/2/09	JDW 4/2/09
xop	MBS 4/2/09	JDW 4/2/09
cxr	MBS 4/2/09	JDW 4/2/09
xray	MBS 4/2/09	JDW 4/2/09
cxr_xray_desc	MBS 4/2/09	JDW 4/2/09
pulse_ox	MBS 4/2/09	JDW 4/2/09
cerumen_removal_manual	MBS 4/2/09	JDW 4/2/09
cerumen_removal_irrigation	MBS 4/2/09	JDW 4/2/09
suture_removal	MBS 4/2/09	JDW 4/2/09
staple_removal	MBS 4/2/09	JDW 4/2/09
suture_staple_removal_number	MBS 4/2/09	JDW 4/2/09
wart_removal_cryo	MBS 4/2/09	JDW 4/2/09
wart_removal_canth	MBS 4/2/09	JDW 4/2/09
wart_removal_cryo_canth_number	MBS 4/2/09	JDW 4/2/09
splint	MBS 4/2/09	JDW 4/2/09
ace	MBS 4/2/09	JDW 4/2/09

wound_dressing	MBS 4/2/09	JDW 4/2/09
splint_ace_wound_dressing_desc	MBS 4/2/09	JDW 4/2/09
exam_after_tx	MBS 4/2/09	JDW 4/2/09
meds_given_in_office	MBS 4/2/09	JDW 4/2/09

Well Visit	Commit Confirmation	Retrieve Confirmation
dental_num_teeth	MBS 4/16/09	MBS 4/16/09
dental_condition	MBS 4/16/09	MBS 4/16/09
dental_drkg_method	MBS 4/16/09	MBS 4/16/09
dental_bottle__bed	MBS 4/16/09	MBS 4/16/09
dental_pacifier_use	MBS 4/16/09	MBS 4/16/09
dental_care_type	MBS 4/16/09	MBS 4/16/09
dental_flosses	MBS 4/16/09	MBS 4/16/09
dental_flouride_toothpaste	MBS 4/16/09	MBS 4/16/09
dental_flouride_supplement	MBS 4/16/09	MBS 4/16/09
dental_flouride_dose	MBS 4/16/09	MBS 4/16/09
dental_dentist	MBS 4/16/09	MBS 4/16/09
dental_last_visit	MBS 4/16/09	MBS 4/16/09
elimation_void_no_concerns	MBS 4/16/09	MBS 4/16/09
elimation_void_wet_diapers_pday	MBS 4/16/09	MBS 4/16/09
elimation_void_concerns	MBS 4/16/09	MBS 4/16/09
elimation_void_desc	MBS 4/16/09	MBS 4/16/09
elimation_stool_no_concerns	MBS 4/16/09	MBS 4/16/09
elimation_stool_pday	MBS 4/16/09	MBS 4/16/09
elimation_stool_type	MBS 4/16/09	MBS 4/16/09
elimation_stool_use_of	MBS 4/16/09	MBS 4/16/09
elimation_stool_concerns	MBS 4/16/09	MBS 4/16/09
elimation_stool_desc	MBS 4/16/09	MBS 4/16/09
elimation_void_stool_frequency	MBS 4/16/09	MBS 4/16/09
elimation_void_stool_comments	MBS 4/16/09	MBS 4/16/09
elimation_tt_no	EEP 4/16/09	JDW 4/16/09
elimation_tt__process	EEP 4/16/09	JDW 4/16/09
elimation_tt_day	EEP 4/16/09	JDW 4/16/09
elimation_tt_night	EEP 4/16/09	JDW 4/16/09
elimation_tt_void_only	EEP 4/16/09	JDW 4/16/09
elimation_tt_stool_only	EEP 4/16/09	JDW 4/16/09
elimation_tt_void_and_stool	EEP 4/16/09	JDW 4/16/09
menses_none	EEP 4/16/09	JDW 4/16/09
menses_age_at_onset	EEP 4/16/09	JDW 4/16/09
menses_regular	EEP 4/16/09	JDW 4/16/09

menses_every_days	EEP 4/16/09	JDW 4/16/09
menses_irregular	EEP 4/16/09	JDW 4/16/09
menses_irregular_desc	EEP 4/16/09	JDW 4/16/09
menses_cramps	EEP 4/16/09	JDW 4/16/09
menses_cramps_severity	EEP 4/16/09	JDW 4/16/09
menses_relieved_with	EEP 4/16/09	JDW 4/16/09
nutrition_milk	EEP 4/16/09	JDW 4/16/09
nutrition_milk_type	EEP 4/16/09	JDW 4/16/09
nutrition_juice	EEP 4/16/09	JDW 4/16/09
nutrition_soda	EEP 4/16/09	JDW 4/16/09
nutrition_gatorade	EEP 4/16/09	JDW 4/16/09
nutrition_drk_quanitity	EEP 4/16/09	JDW 4/16/09
nutrition_dairy	EEP 4/16/09	JDW 4/16/09
nutrition_dairy_spd	EEP 4/16/09	JDW 4/16/09
nutrition_fruits_vegs	EEP 4/16/09	JDW 4/16/09
nutrition_fruits_vegs_spd	EEP 4/16/09	JDW 4/16/09
nutrition_red_meat	EEP 4/16/09	JDW 4/16/09
nutrition_other_meat	EEP 4/16/09	JDW 4/16/09
nutrition_meat_spd	EEP 4/16/09	JDW 4/16/09
nutrition_mv	EEP 4/16/09	JDW 4/16/09
nutrition_calcium	EEP 4/16/09	JDW 4/16/09
nutrition_fe	EEP 4/16/09	JDW 4/16/09
nutrition_other	EEP 4/16/09	JDW 4/16/09
nutrition_vitam_desc	EEP 4/16/09	JDW 4/16/09
nutrition_vitam_quanitity	EEP 4/16/09	JDW 4/16/09
pmh_majorDx	EEP 4/16/09	JDW 4/16/09
pmh_surgery	EEP 4/16/09	JDW 4/16/09
pmh_allergy_reaction_to_meds	EEP 4/16/09	JDW 4/16/09
pmh_reaction_to_immunization	EEP 4/16/09	JDW 4/16/09
pmh_medications	EEP 4/16/09	JDW 4/16/09
pmh_social_history	EEP 4/16/09	JDW 4/16/09
pmh_tobacco_use__family	EEP 4/16/09	JDW 4/16/09
parent_patient_concerns	EEP 4/16/09	JDW 4/16/09
optho_no_concerns	EEP 4/16/09	JDW 4/16/09
optho_concerns	EEP 4/16/09	JDW 4/16/09
ophto_description	EEP 4/16/09	JDW 4/16/09
optho_vision_screen_at_school	EEP 4/16/09	JDW 4/16/09
optho_vision_screen_optho_optom	EEP 4/16/09	JDW 4/16/09
optho_vision_screen_last_visit	EEP 4/16/09	JDW 4/16/09
daycare_school_daycare	EEP 4/16/09	JDW 4/16/09
daycare_school_daycare_type	EEP 4/16/09	JDW 4/16/09

daycare_school_daycare_where	EEP 4/16/09	JDW 4/16/09
daycare_school_school	EEP 4/16/09	JDW 4/16/09
daycare_school_school_where	EEP 4/16/09	JDW 4/16/09
daycare_school_school_k12_grade	EEP 4/16/09	JDW 4/16/09
daycare_school_high_school	EEP 4/16/09	JDW 4/16/09
daycare_school_college	EEP 4/16/09	JDW 4/16/09
daycare_school_college_year	EEP 4/16/09	JDW 4/16/09
daycare_school_work	EEP 4/16/09	JDW 4/16/09
daycare_school_no_concerns	EEP 4/16/09	JDW 4/16/09
daycare_school_concerns	EEP 4/16/09	JDW 4/16/09
daycare_school_description	EEP 4/16/09	JDW 4/16/09
sleep_no_concerns	EEP 4/16/09	JDW 4/16/09
sleep_duration_through_night	EEP 4/16/09	JDW 4/16/09
sleep_duration_wakes	EEP 4/16/09	JDW 4/16/09
sleep_duration_timespday	EEP 4/16/09	JDW 4/16/09
sleep_naps_value	EEP 4/16/09	JDW 4/16/09
sleep_naps_timespday	EEP 4/16/09	JDW 4/16/09
sleep_naps_hourspday	EEP 4/16/09	JDW 4/16/09
sleep_hourspnnight	EEP 4/16/09	JDW 4/16/09
sleep_problems_with	EEP 4/16/09	JDW 4/16/09

Scenario	Confirmation
Quick Search for patients that do not exist yet	MBS
Quick Search for patients by all possible fields	MBS
Commit all sections of Nurse Assessment in different orders	JDW
Graph data points including changing previous points	MBS
Access Grapher before any points have been recorded yet	JDW
Switch between normal Nurse Assessment and Grapher	JDW
Commit all sections of Sick Visit in different orders	EEP
Commit and retrieve drawings after using the Undo feature	EEP
Commit all sections of Well Visit in different orders	EEP
Switch between all states in all orders in the overall system	MBS
Check for invalid entries in all fields	JDW