

MLKMC Electronic Healthcare System

System Design Document

Date: 30-Oct-2010

Project Manager: Ryan Olson

Team: Tyler Bradovich, Cameron Harp, Matthew Kimber, Ryan Olson

Sponsor: Prof. Richard Fry

Client: Martin Luther King Memorial Clinic

Website: <http://kaizen.matthewkimber.com/>

Table of Contents

Introduction	3
Class Diagram and Interface Specification	3
Class Diagram	3
Data Types and Operation Signatures.....	4
System Architecture and System Design.....	10
Architectural Styles.....	10
Identifying Subsystems.....	11
Mapping Subsystems to Hardware.....	11
Persistent Data Storage.....	12
Network Protocol.....	12
Global Control Flow	13
Hardware Requirements.....	14
Software Requirements.....	15
Algorithms and Data Structures	16
Algorithms.....	16
Data Structures	19
User Interface	20
Appointment Scheduling.....	21
Login	22
Administrator - Create User.....	24

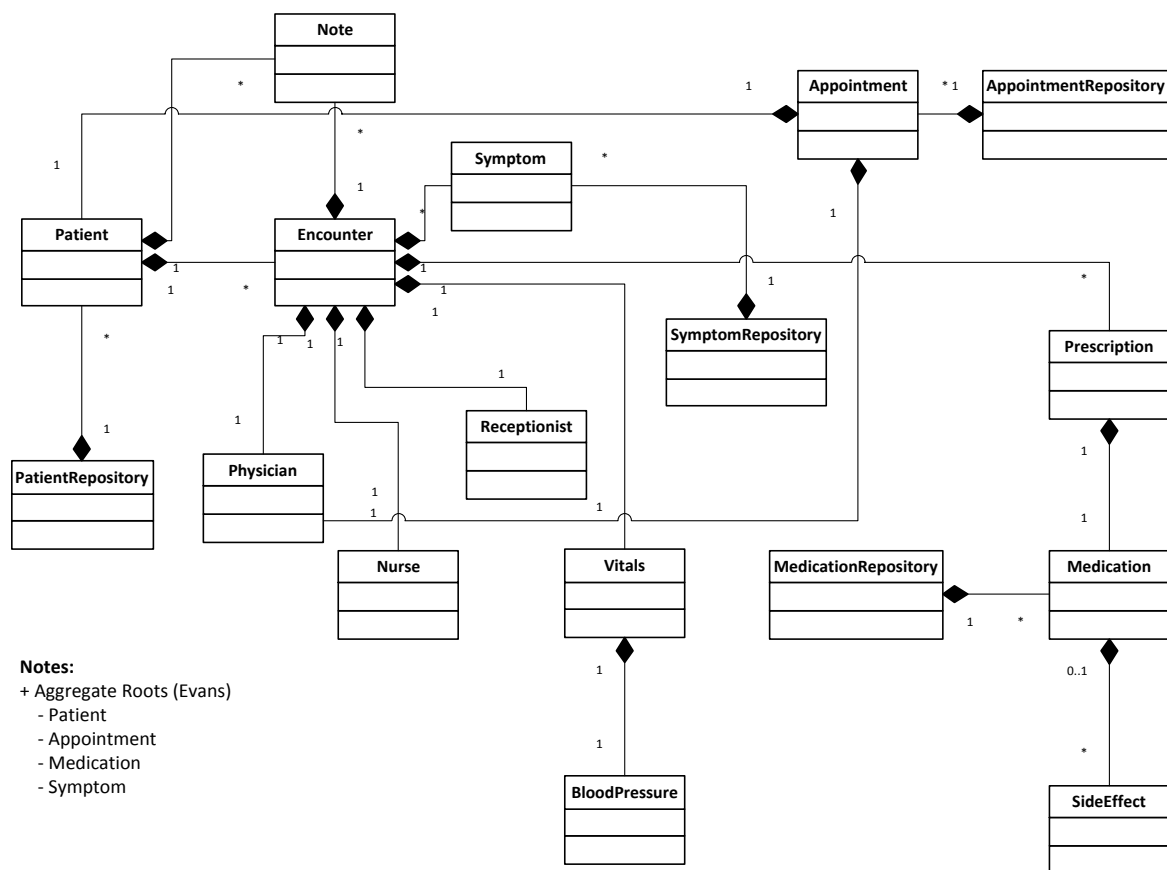
Administrator – View/Edit Users.....	25
Pharmacy Inventory Management.....	26
Clinic Supplies Inventory Management	27
Progress Report	28
Responsibility Breakdown.....	28
Plan of Work.....	28
Contributions Breakdown.....	31
Works Cited.....	32

Introduction

The Electronic Healthcare System (EHS) will be composed of the following classes and system architecture design. These requirements have been derived from the SSRS document and the new information we have received from Dr. Duah. Any new information gathered may cause further expansion or modification as we discover more requirements of the system.

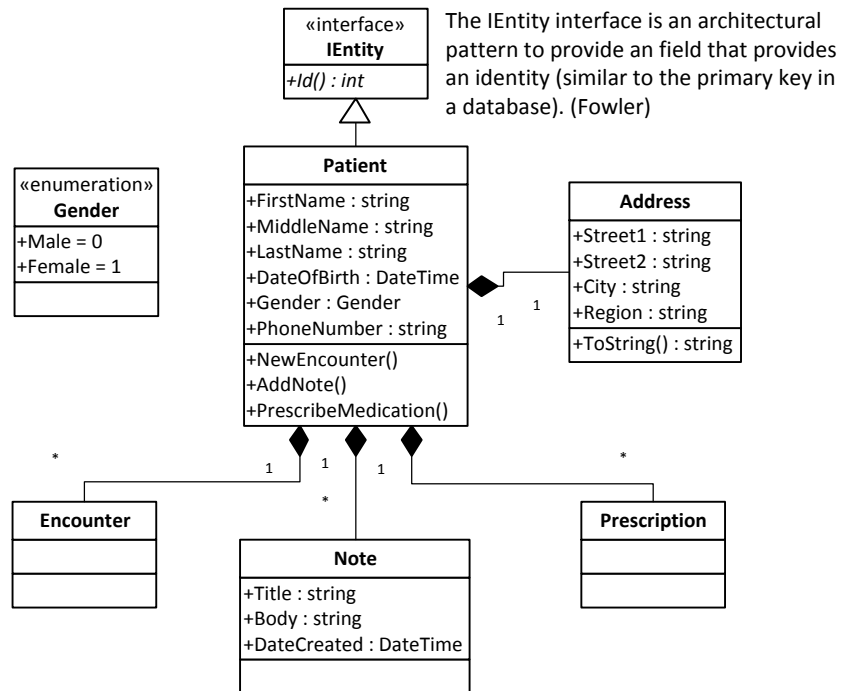
Class Diagram and Interface Specification

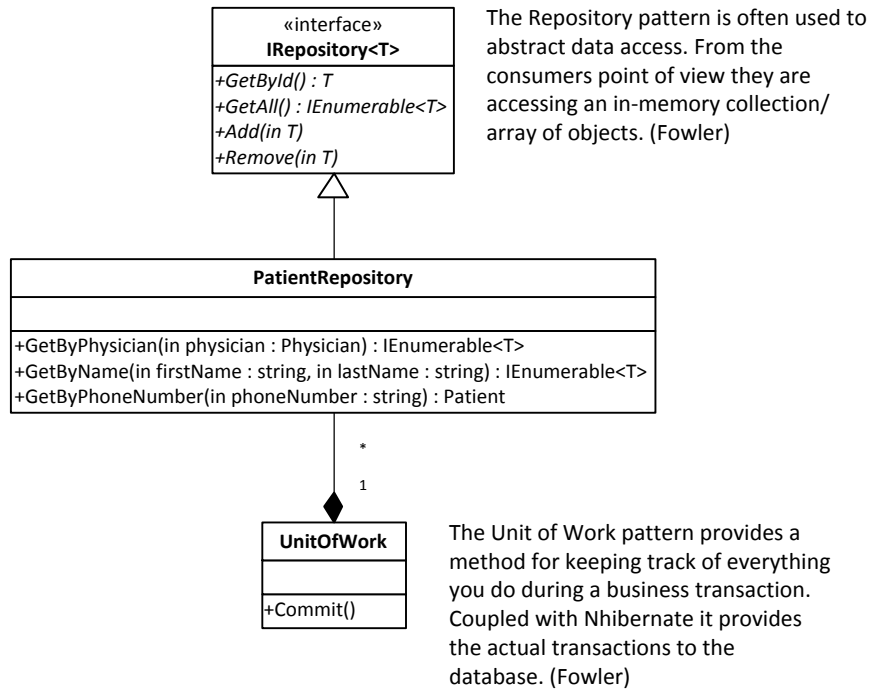
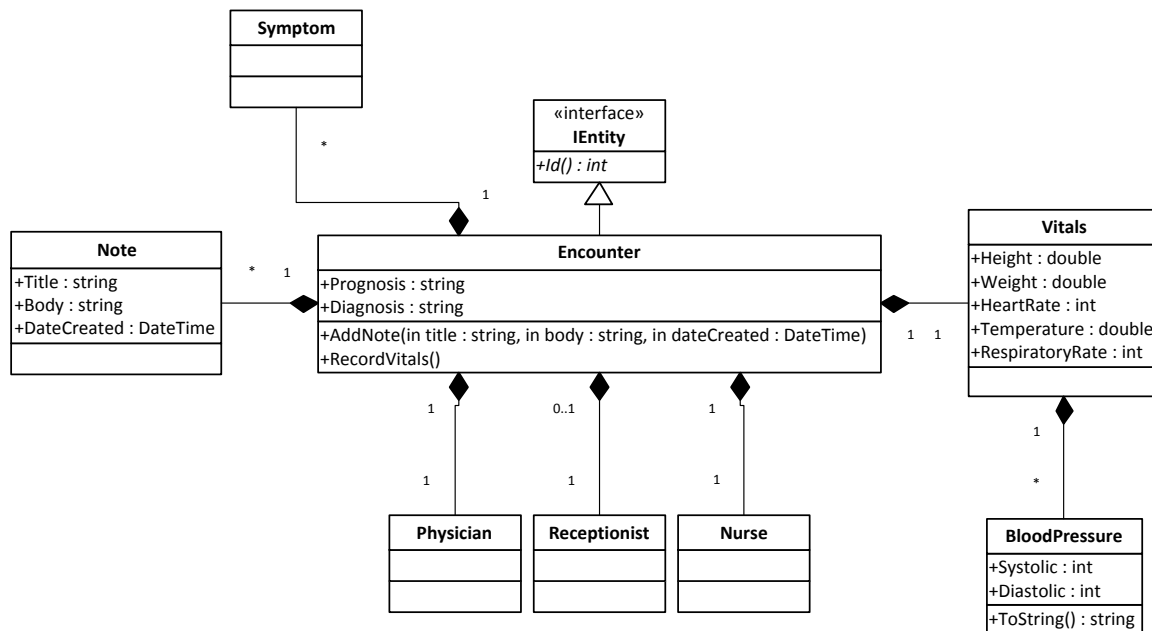
Class Diagram

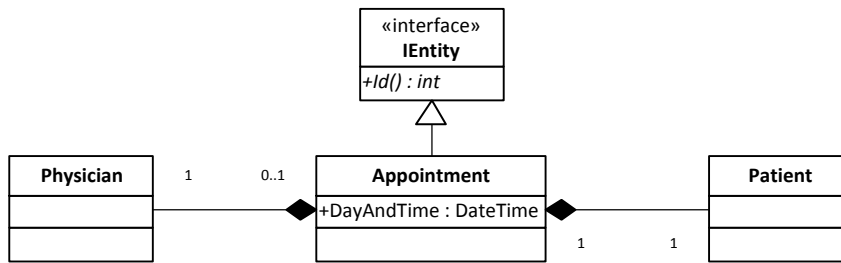
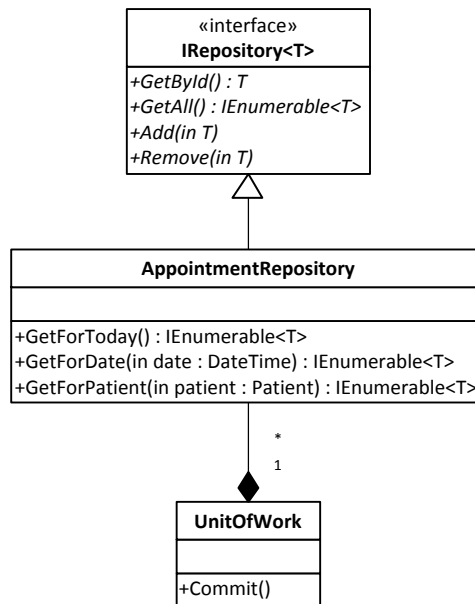


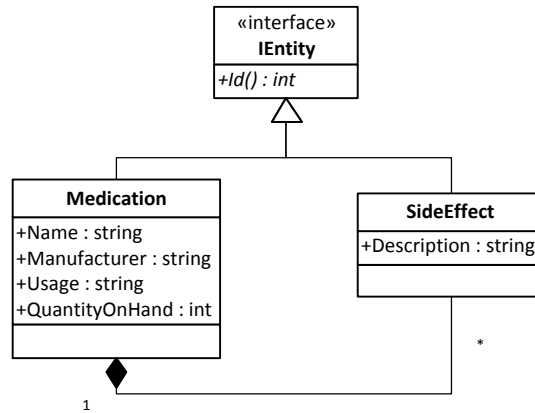
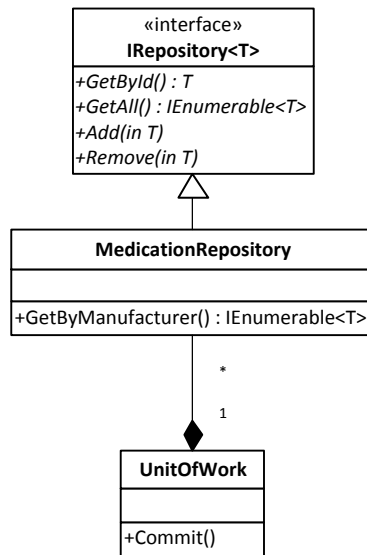
Data Types and Operation Signatures

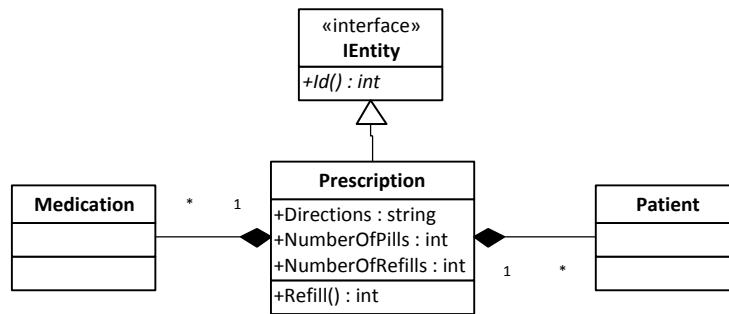
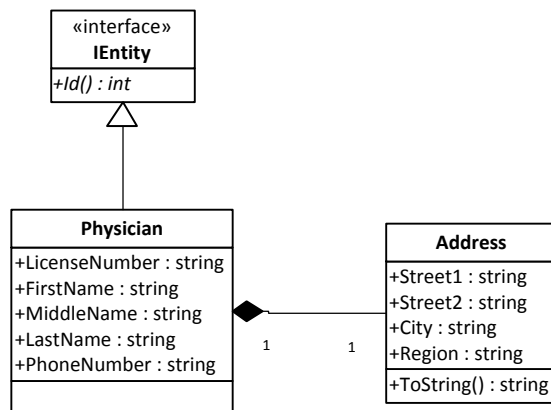
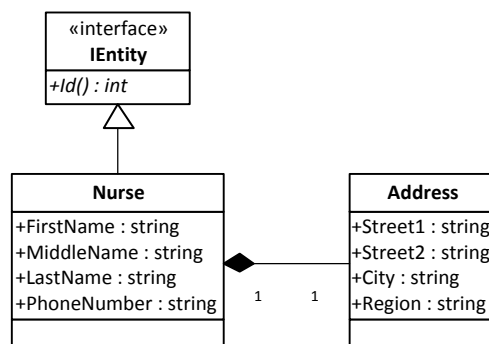
Patient Class:

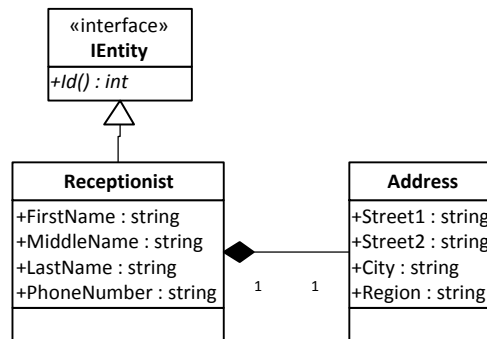
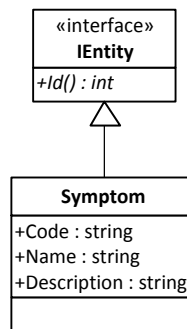
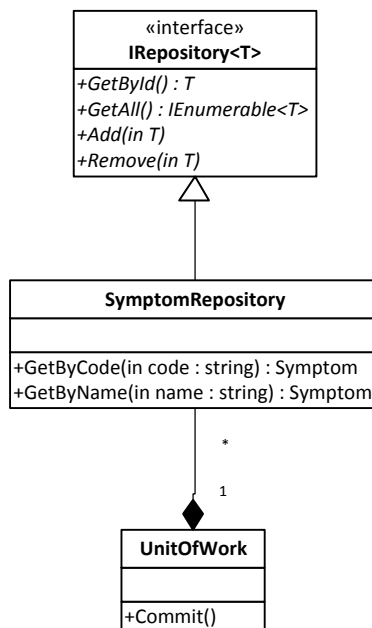


PatientRepository Class:**Encounter Class:**

Appointment Class:**AppointmentRepository Class:**

Medication Class:**MedicationRepository Class:**

Prescription Class:***Physician Class:******Nurse Class:***

Receptionist Class:**Symptom Class:****SymptomRepository Class:**

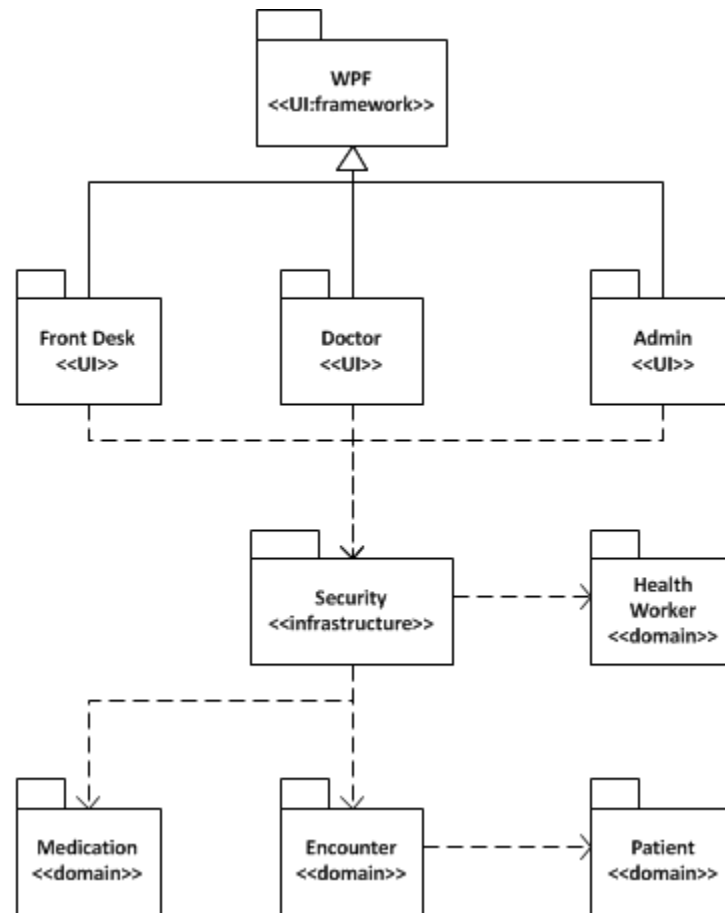
System Architecture and System Design

Architectural Styles

The MLKMC-EHS will be built using a service-oriented, n -tier architecture. Multiple client computers will connect to a local server, which will provide well-defined services for each client. These services will typically be composed of methods for retrieving data from, and changing data in the database. Client computers “consume” a service by using what looks just like a method call. For example, a client might need all the medical records for a particular patient. To do this, the client code basically performs a method call on the service, providing the Patient as a parameter, and the service returns the medical records.

The system will also be separated into different layers or tiers, “ n ” being the number of tiers in our n -tier architecture. Presentation/UI, services, domain/business logic, and data access code will reside in separate classes. Keeping these separate allows for better organization, and easier modification to any single tier. Group members can work on separate tiers at the same time and not conflict with each other.

Identifying Subsystems

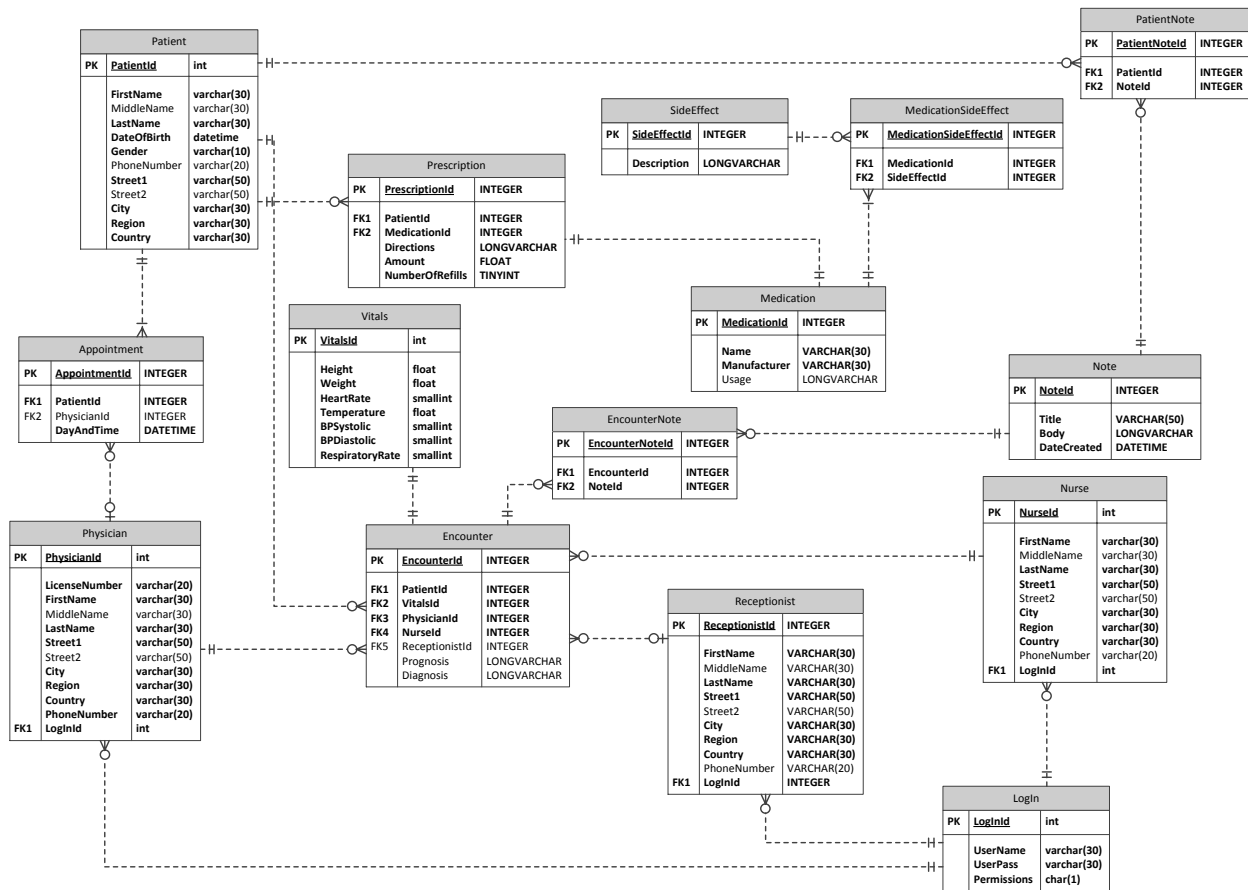


The MLKMC-EHS system is divided into subsystems as pictured above. There are three distinct user interface packages, each containing a set of files and resources to allow user interaction. Domain packages contain all other classes for the system, with closely related classes grouped together.

Mapping Subsystems to Hardware

The user interface modules will run on separate computers from the non-interface modules. The Front Desk UI will run only on clients at the front desk. The Doctor and Admin user interface subsystems will run on computers located in doctor's offices. All other domain modules will run on a local server computer, which can be accessed by any of the client computers.

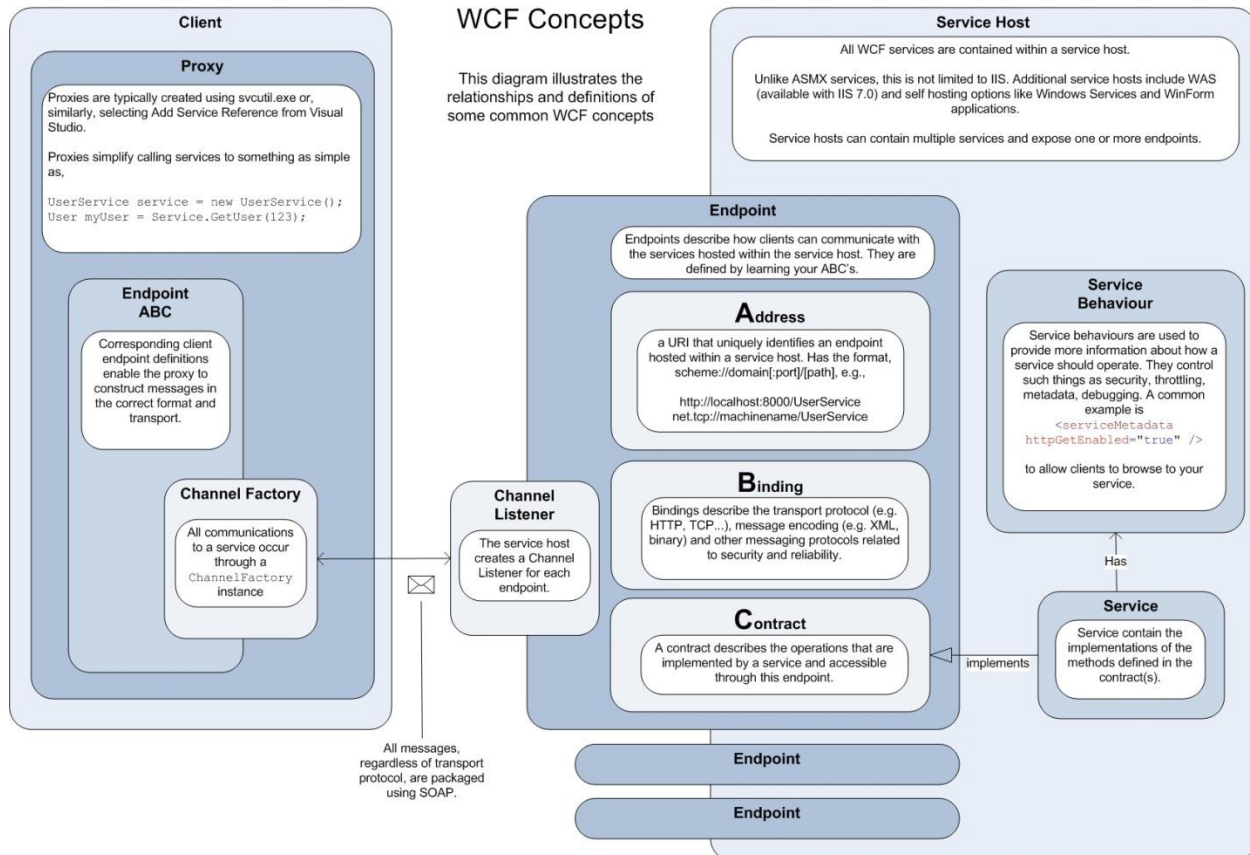
Persistent Data Storage



Network Protocol

We will be using the Windows Communication Foundation (WCF) to distribute the system over multiple computers. A central server will host the application, which will provide each client with services in order to access and modify central data. By using WCF and having it on a central server it will allow for updates to be easily pushed out to all computers on the network, minimizing having to update every computer individually.

The following diagram from andrewtokely.net describes the WCF architecture and organization.



(Tokely)

Global Control Flow

Execution Order

The system will be event-driven via a graphical user interface. Users will be able to generate actions in different orders by clicking different command buttons. The system will then respond to that event in an appropriate manner that we define.

Time Dependency

The system will be of the event-response type, with no concerns for real time. Various weekly reports will be generated however, using a date-time mechanism to generate a weekly event.

Concurrency

The methodology for handling concurrency will be optimistic concurrency. The environment in which the system will be installed has a low contention for data and therefore the need for a

pessimistic concurrency model is unnecessary. However, this approach will be evaluated as testing and integration occurs.

Hardware Requirements

The MLKMC-EHS will consist of one computer at the front desk, three computers for the physician's offices, and a computer acting as a local server. These computers will be networked using standard CAT-5 cable. Each computer will need a monitor with a resolution of *at least* 1024 x 768 pixels. In summary, the hardware requirements for the system include the following minimal setup:

- Server System
 - 1.6 GHz+ processor.
 - 512 MB of RAM.
 - 25 GB of unused hard drive space.
 - 100 Mbps network adapter
 - Monitor supporting a resolution of 1024 x 768.
- Client Systems
 - 1.6 GHz+ processor.
 - 512 MB of RAM.
 - 2 GB of unused hard drive space.
 - 100 Mbps network adapter.
 - DirectX 9 capable video card.
 - Monitor supporting a resolution of 1024 x 768.

Common hardware that is required for the system will include:

- One router for DHCP.
- Zero to one switches.
- CAT-5 cabling.

- One or more printers.
- One laser barcode scanner.

Software Requirements

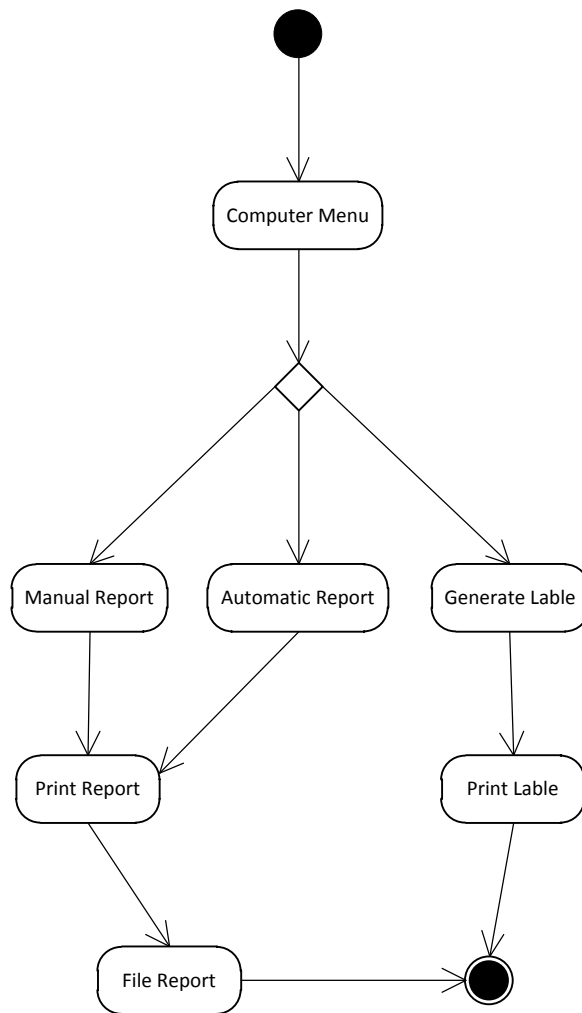
Various software packages will need to be installed on the hardware that comprises the system. The following items are dependencies of the system to be developed and required in order to run properly:

- Server
 - Microsoft Windows XP or greater.
 - Microsoft .NET Framework 4.0 or greater.
 - MySQL Community Server 5.1+
- Client
 - Microsoft Windows XP or greater.
 - Microsoft .NET Framework 4.0 or greater.

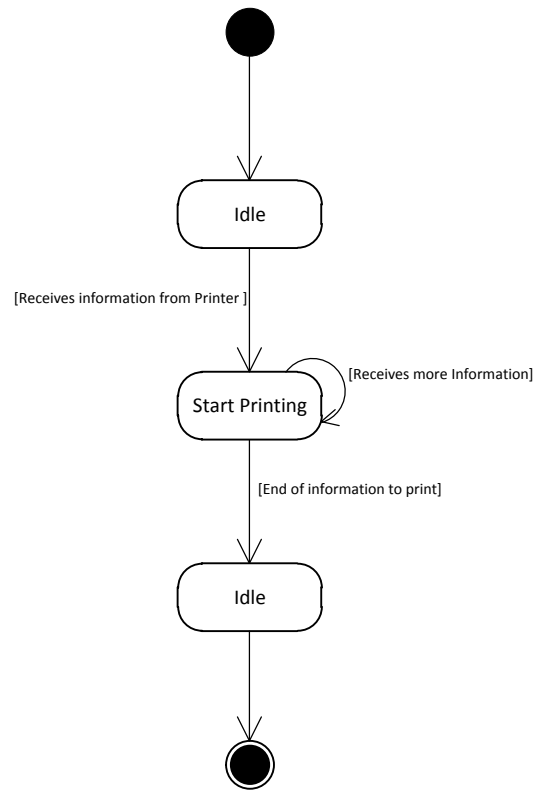
Algorithms and Data Structures

Algorithms

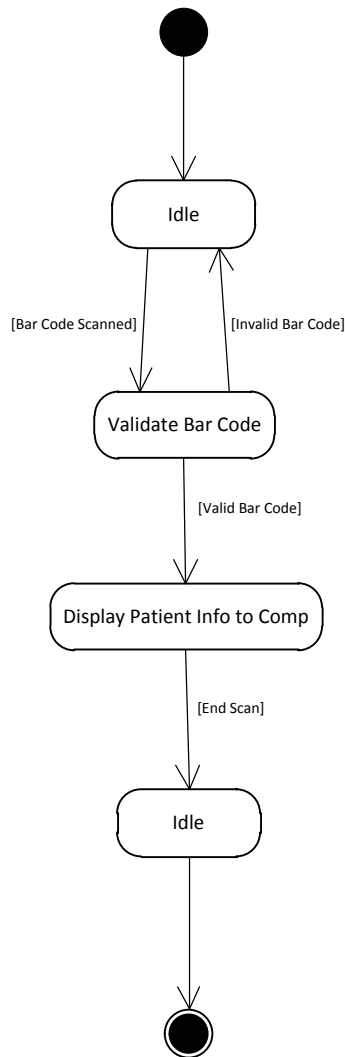
Inventory System:



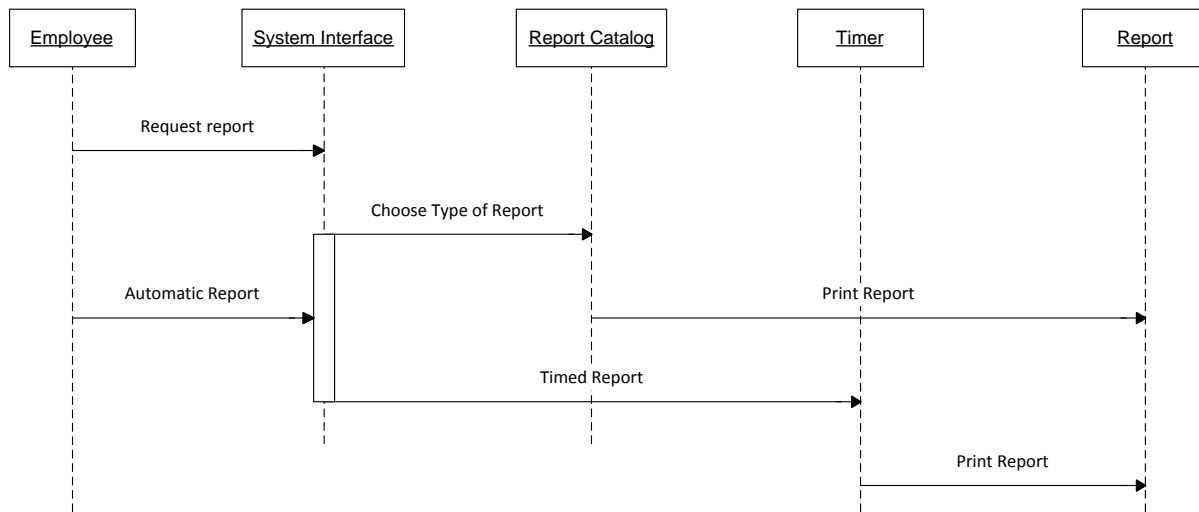
Printer System:



Scanner System:



Sequence Diagram Reports:



Data Structures

Various data structures will be used within the MLKMC-EHS most of which are derivatives of data structures already implemented in the Microsoft .NET Framework. These data structures include the following:

- List<T>
- ObservableCollection<T>
- Dictionary<TKey, TValue>
- SortedDictionary<TKey, TValue>
- HashSet<T>
- As well as classes that implement the IEnumerable<T> interface.

For now these data structures will be the only data structures used in the implementation. However, custom data structures may be implemented in the future and will be fully documented here.

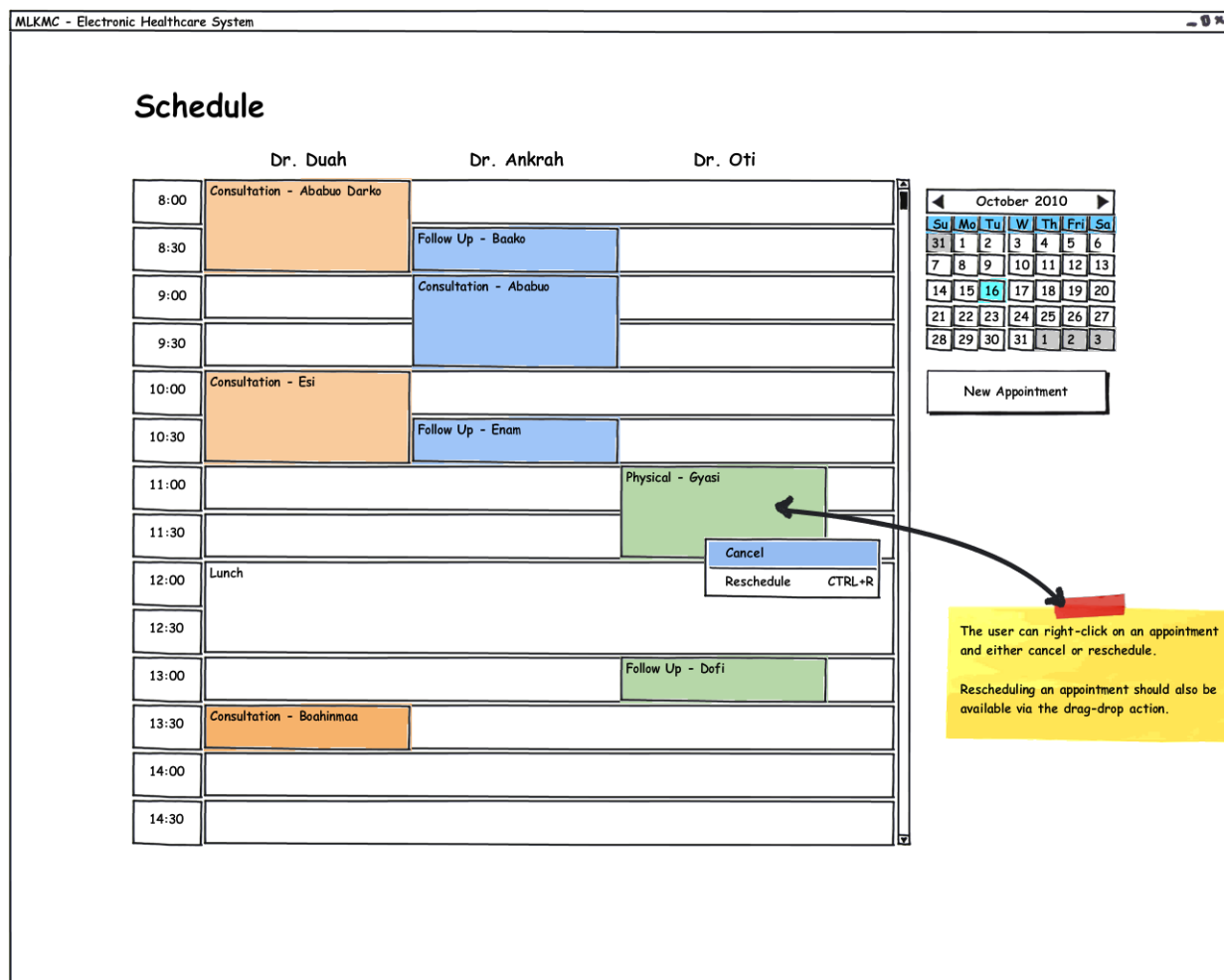
User Interface

The initial screen mock-ups developed for *Report #1* have been modified in the following ways:

1. New method of working with appointment scheduling.
2. Login mock-ups created.
3. Administrative capabilities mock-ups created.
4. Mock-up created for pharmacy inventory management.
5. Mock-up created for clinic supply inventory management.

Appointment Scheduling

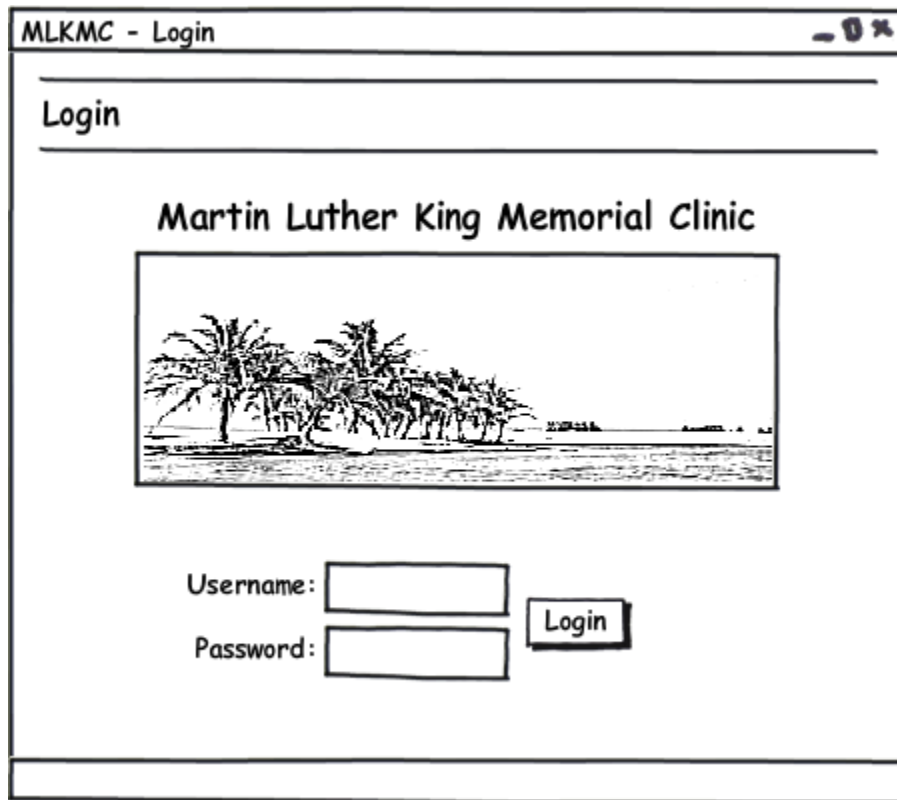
Instead of the clerk clicking *Appointments*, then clicking *View Appointments* or *Make Appointment*, all the appointment functionality will be available from the home screen for the front desk. It is easy for the clerk to see upcoming appointments, and when each doctor has an opening for a new appointment. From this home screen, existing appointments can be easily canceled or rescheduled by right clicking the appointment block, or by drag and drop to reschedule. This is a significant improvement in ease of use over the old scheduling design.



Login

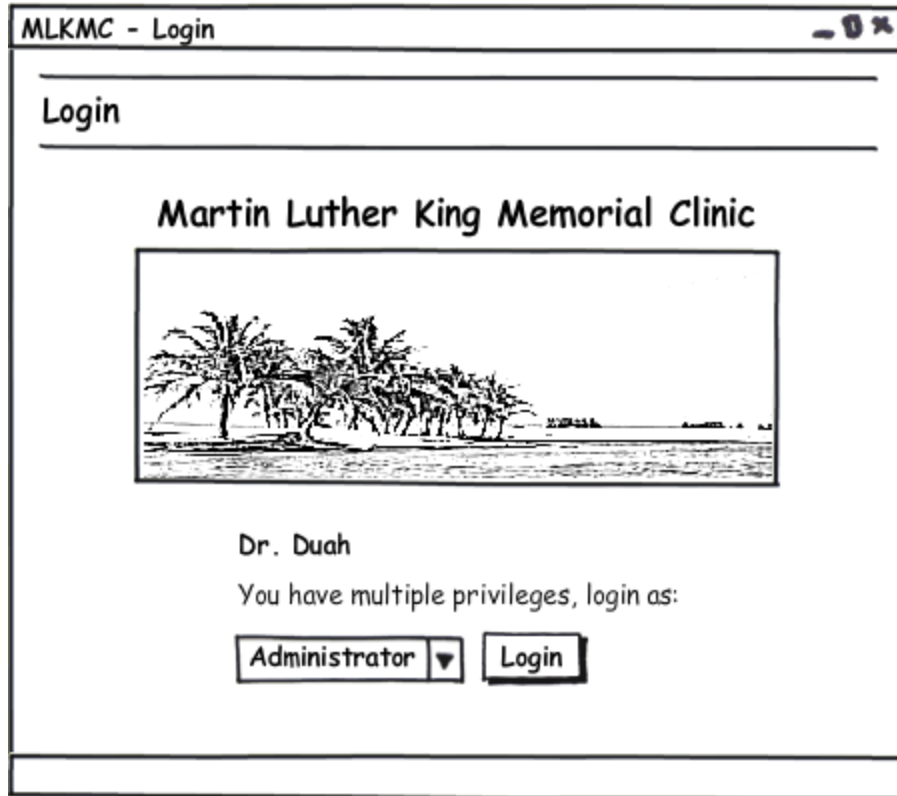
To be able to use the system, every user must log-in with a unique username and a password.

These usernames and passwords can be managed by a user with administrative privileges for the system.



The screenshot shows a window titled "MLKMC - Login". Inside the window, the word "Login" is displayed at the top. Below it is the text "Martin Luther King Memorial Clinic". Underneath the text is a rectangular image of a tropical beach with palm trees and a body of water. At the bottom of the window, there are two input fields: "Username:" and "Password:". To the right of the "Password:" field is a button labeled "Login".

1. Enter a username (case-insensitive) and a password (case-sensitive).
2. Click "Login" to submit the username and password.



1. If a user has multiple privileges for the system, such as Doctor and Administrator, then the options above will be presented.
2. Choose a login type from the drop-down list.
3. Click "Login" to enter the system under the chosen role view.

Administrator - Create User

Only users with administrative rights can create new users for the system.

The screenshot shows a web application window titled "MLKMC - Admin: Create User". At the top, there are two tabs: "Create User" (which is highlighted in blue) and "View/Edit Users". Below the tabs, the heading "Admin: Create User" is displayed. The main content area is divided into two sections. The "Login Info" section on the left contains two input fields: "Username:" and "Password:". The "User Privileges" section on the right contains a list of roles with checkboxes: "Receptionist" (checked), "Nurse" (unchecked), "Doctor" (unchecked), and "Administrator" (unchecked). At the bottom right of the form, there is a "Create User" button.

1. Click the "Create User" tab from the menu bar if not already selected.
2. Enter the desired username and password of the new user.
3. Choose at least one user privilege to define what system functions the user may perform.

Administrator – View/Edit Users

Users with administrative rights can view and edit information for other system users.

Username	Password	Privileges
Asare	hi123	nurse
Ansere	bye321	doctor
Oppong	yes99no	receptionist
Duah	pass456	admin-doctor

1. Click the “View/Edit Users” tab from the menu bar.
2. Select a user from the list by clicking the row that contains their username.
3. Click one of the three buttons to perform the action on the selected user.

Pharmacy Inventory Management

The pharmacy inventory can be viewed, adjusted, and managed using the MLKMC system.

Drug Name	Type	Quantity
Tetracycline	Pill	25
Ampicillin	Pill	40
Ambien	Pill	57
Quinacrine	Shot	5

Quantities can be adjusted directly on the grid.

1. Click the “Pharmacy Inventory” tab if not already selected.
2. Click inside any quantity field and type a new quantity to adjust its value.
3. Click “Add New Drug Name” or “Delete a Drug” to add/delete entire items on the list.

Clinic Supplies Inventory Management

The clinic supplies inventory can be viewed, adjusted, and managed using the MLKMC system.

Supply Name	Quantity
Hypodermic Needle	31
Large Bandage	12
Small Bandage	27
Neosporin Cream	9

Add New Supply Name

Delete a Supply

Quantities can be adjusted directly on the grid.

1. Click the “Clinic Supplies Inventory” tab from the menu bar.
2. Click inside any quantity field and type a new quantity to adjust its value.
3. Click “Add New Supply Name” or “Delete a Supply” to add/delete entire items on the list.

Progress Report

Currently, all we have implemented in our project, so far, is we have created the skeleton of our project and added some of the classes for the data tier and domain tier. In the near future, we will implement more functionality that would allow appointment viewing/setting and patient lookup/creation.

Responsibility Breakdown

Modules	Ryan	Tyler	Matt	Cameron
Data Tier	20%	20%	30%	30%
Repositories	20%	20%	40%	20%
UnitOfWork	10%	10%	70%	10%
Business Tier	30%	30%	20%	20%
Services Tier	25%	25%	25%	25%
Presentation Tier	25%	25%	25%	25%

Plan of Work

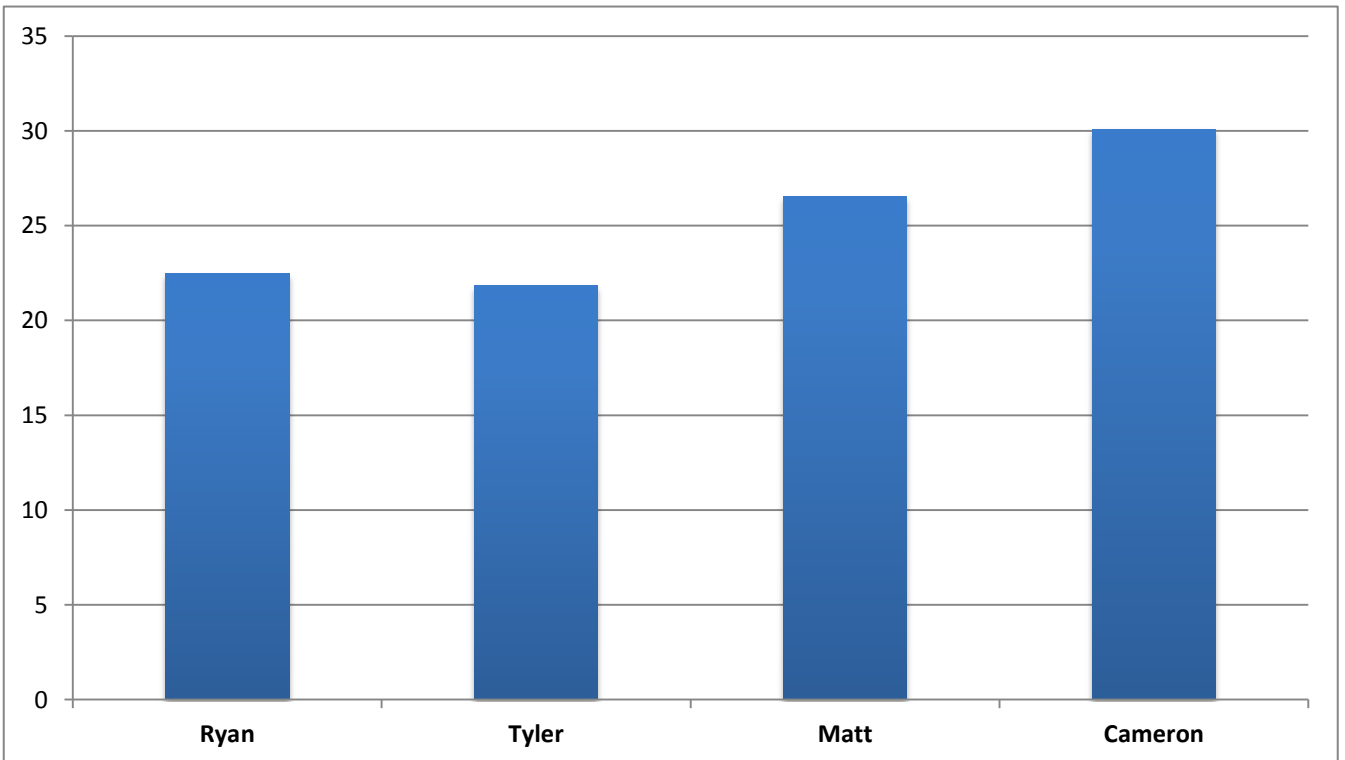
Task Name	Duration	Start	Finish	Predecessors
Analysis/Software Requirements	4.53 days?	Wed 9/1/10	Tue 9/7/10	
Conduct needs elicitation and analysis	10 hrs	Wed 9/1/10	Thu 9/2/10	
Draft preliminary software specifications	10 hrs	Thu 9/2/10	Fri 9/3/10	2
Develop delivery timeline	2 hrs	Mon 9/6/10	Mon 9/6/10	2,3,5
Review software specifications	0.5 hrs	Fri 9/3/10	Fri 9/3/10	3
Obtain approval to proceed	0.25 hrs	Tue 9/7/10	Tue 9/7/10	2,3,4,5
Analysis Complete	0 days	Tue 9/7/10	Tue 9/7/10	6
Design	55 days	Mon 9/20/10	Mon 12/6/10	1
Review software specifications	2 hrs	Mon 9/20/10	Mon 9/20/10	
Decide on SRS template	0.5 hrs	Mon 9/20/10	Mon 9/20/10	10
Develop functional specifications	2 hrs	Tue 9/28/10	Tue 9/28/10	11
Develop nonfunctional specifications	2 hrs	Tue 9/28/10	Tue 9/28/10	11
Develop prototype based on functional requirements	10 hrs	Tue 9/28/10	Wed 9/29/10	12,13,17,18,19
Review functional specifications	2 hrs	Wed 9/29/10	Wed 9/29/10	12,13
Determine plan of work	1 hr	Mon 9/20/10	Mon 9/20/10	11
Develop use cases	4 hrs	Mon 9/20/10	Mon 9/20/10	10
Develop use case specifications	1 hr	Tue 9/21/10	Tue 9/21/10	17

Design UI Mockups	10 hrs	Mon 9/20/10	Tue 9/21/10	10
Fill out contributions breakdown	1 hr	Tue 9/21/10	Tue 9/21/10	18
Report #1 (First draft)	0 days	Sat 10/2/10	Sat 10/2/10	
Schedule meeting	0.5 hrs	Mon 10/4/10	Mon 10/4/10	21
Meeting with fry	0.5 hrs	Wed 10/6/10	Wed 10/6/10	22
Edit as specified from meeting	10 hrs	Wed 10/6/10	Thu 10/7/10	23
Report #1 (Final)	0 days	Sat 10/16/10	Sat 10/16/10	24
Decide on SSD template	0.5 hrs	Mon 10/18/10	Mon 10/18/10	25
Determine interface specification	1 hr	Mon 10/18/10	Mon 10/18/10	26
Design architecture	3 hrs	Mon 10/18/10	Mon 10/18/10	25
Design UML	10 hrs	Mon 10/18/10	Tue 10/19/10	25
Define our algorithms and data structures	1 hr	Mon 10/18/10	Mon 10/18/10	26
Decide how we are going to network the computers	1 hr	Mon 10/18/10	Mon 10/18/10	25
Create progress report	2 hrs	Mon 10/18/10	Mon 10/18/10	30
Hardware requirements	1 hr	Mon 10/18/10	Mon 10/18/10	26
Design database ERD	10 hrs	Mon 10/18/10	Tue 10/19/10	25
Report #2	0 days	Sat 10/30/10	Sat 10/30/10	34,33,32,31,30,29,28
Design product brochure	1 hr	Mon 11/15/10	Mon 11/15/10	35
Prepare for Demo #1 (slides, script, etc.)	2 hrs	Mon 11/15/10	Mon 11/15/10	
Prepare program for in class demonstration	2 hrs	Mon 11/15/10	Mon 11/15/10	
Obtain approvals to proceed	0.25 hrs	Mon 11/15/10	Mon 11/15/10	36,37,38
Demo #1	0 days	Sat 11/20/10	Sat 11/20/10	35
Edit report #1 and report #2 and merge them	2 hrs	Mon 11/22/10	Mon 11/22/10	
Add history of work and current status	0.5 hrs	Wed 11/24/10	Wed 11/24/10	
Add conclusions and plan for future work	0.5 hrs	Wed 11/24/10	Wed 11/24/10	
Generate references that were used in our project	0.5 hrs	Wed 11/24/10	Wed 11/24/10	
Get client to sign it off	0.25 hrs	Wed 11/24/10	Wed 11/24/10	41,42,43,44
Report #3	0 days	Mon 12/6/10	Mon 12/6/10	45
Design Complete	0 days	Mon 12/6/10	Mon 12/6/10	46
Development	13 days	Sun 1/3/10	Thu 1/21/10	9
Testing	18.63 days	Wed 1/5/11	Mon 1/31/11	48
Training	3.5 days	Wed 2/2/11	Mon 2/7/11	56
Documentation	14.31 days	Wed 2/9/11	Tue 3/1/11	73

Pilot	6.88 days	Mon 3/7/11	Tue 3/15/11	81
Deployment	20 days	Sun 2/20/11	Sat 3/19/11	91
Post Implementation Review	1.13 days	Wed 4/27/11	Thu 4/28/11	98
SDLC complete	0 days	Thu 4/28/11	Thu 4/28/11	109

Contributions Breakdown

Tasks	Ryan	Tyler	Matt	Cameron	Allocation
Project Management	100%	0%	0%	0%	100%
Interaction Diagrams	15%	15%	15%	55%	100%
Class Diagrams and Specs	15%	25%	35%	25%	100%
System Architecture and Design	10%	15%	60%	15%	100%
Network Protocol	10%	40%	25%	25%	100%
ERD Diagrams	0%	0%	50%	50%	100%
User Interface Design	10%	75%	15%	0%	100%
Plan of Work	40%	20%	20%	20%	100%
References	25%	25%	25%	25%	100%
Possible Points	22.5	21.85	26.55	30.1	



Works Cited

Evans, Eric. Domain-Driven Design. Boston: Addison-Wesley, 2004.

Fowler, Martin. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2003.

Kuaté, Pierre Henri, et al. NHibernate in Action. Greenwich: Manning, 2009.

Tokely, Andrew. Thoughts on the Software Development Industry. 31 July 2008. 22 October 2010
<<http://andrewtokeley.net/archive/2008/07/31/wcf-simplified-to-a-single-diagram.aspx>>.