

MLKMC Electronic Healthcare System

Specification and Design

Date: 6-Dec-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

Summary of Changes.....	3
User Requirements Definition	4
System Requirements Specification	4
Functional Requirements	4
Non-Functional Requirements.....	6
Interactions Diagrams.....	8
Registering a Patient	8
Record Patient's Vitals.....	9
Visit with Physician.....	10
Class Diagrams	11
Class Diagram	11
Data Types and Operation Signatures	12
System Architecture and System Design	19
Architectural Styles	19
Identifying Subsystems.....	20
Mapping Subsystems to Hardware	20
Persistent Data Storage.....	21
Network Protocol	21
Global Control Flow	22
Hardware Requirements.....	23
Software Requirements.....	24
Algorithms and Data Structures	25
Algorithms.....	25

Inventory System:.....	25
Data Structures	28
User Interface Design and Implementation	29
Add Patient to System.....	29
Search for Patient	32
Fulfill an Appointment	34
Take Patient Vitals	36
Issue Medications	37
Bill Patient	38
View Pharmacy Inventory Report.....	39
View Pharmacy Sales Report	40
View Clinic Income Report.....	41
Select Patient to See.....	42
View/Add Patient Records.....	44
View/Make Diagnosis.....	45
Prescribe Medicine	46
Appointment Scheduling	46
History of Work & Current Status of Implementation.....	48
Conclusions and Future Work.....	49
Bibliography	51
Approval Signatures	52

Summary of Changes

Version History

Date	Description	Author(s)
<i>2-Oct-2010</i>	Initial Draft	Everyone
<i>11-Oct-2010</i>	Use case section reformatted. Localized reports and mockups.	Matthew Kimber
<i>16-Oct-2010</i>	Added UC-13. Some various small edits. Merged in our individual contributions and plan of work.	Tyler Bradovich
<i>16-Oct-2010</i>	Added signature sheet and performed some additional formatting.	Matthew Kimber
<i>1-Dec-2010</i>	Began the merge of the two documents.	Cameron Harp
<i>5-Dec-2010</i>	Added “History of Work & Current Status” and “Conclusions & Future Work”.	Ryan Olson & Matthew Kimber
<i>6-Dec-2010</i>	Finalized the document.	Matthew Kimber

User Requirements Definition

The proposed EHS software will allow physicians, nurses, and staff to manage clinical data as well as reference it in a timely fashion. The system will give the receptionist the ability to register new patients, modify their information, and manage appointments. Nurses will be aided in recording initial encounter information such as vitals and other commonly recorded statistics. The physician's job will be improved by allowing him or her to access and record a patient's medical history. Physicians will also be able to prescribe medication to a patient and give them a printed prescription rather than a handwritten one which will reduce errors in fulfillment. The physician will also have the ability to record notes about the patients and any concerns for their well-being. He or she will be able to record a prognosis and refer back to it in follow-up appointments with that patient.

Clinical administration will receive benefits from the EHS software. Users of the system will have the ability generate reports based on a variety of data. For instance, a physician will be able to generate a report that shows the number of patients seen and get an overall view of the health of his patients. This will also help in better estimating the number of supplies and drugs that will be need on hand in the coming months. A balance sheet can be generated showing the debits and credits of the clinic thus giving an overview of the clinic's financial health.

System Requirements Specification

Functional Requirements

1. The system shall provide a user interface for physicians, nurses, and other staff members.
2. The system shall permit the scheduling of appointments.
3. The system shall allow for the scheduling of walk-in patients.
4. The system shall allow for the scheduling of follow-up appointments for patients.
5. The system shall have the ability to cancel appointments.
6. The system shall allow new patients to be added to the system.
7. The system shall allow a patient's personal information to be edited.
8. The system shall allow a patient to be removed from the system only by a physician.

9. The system shall permit the receptionist to print a new patient information sheet for the patient to fill out personal information and previous medical history.
10. The system shall allow data entry of the information given to the receptionist by the patient via the patient information sheet.
11. The system shall permit the receptionist to check-in a patient upon arrival.
12. The system shall permit the receptionist to maintain patient information at check in.
13. The system shall allow nurses to record the vitals of a patient.
14. The system shall have the ability to record a patient's medical history.
15. The system shall allow a physician to review a patient's medical history.
16. The system shall allow a physician to add information to a patient's medical history.
17. The system shall allow a physician to edit information in a patient's medical history.
18. The system shall allow a physician to remove information from a patient's medical history.
19. The system shall allow physicians to record diagnoses of patients.
20. The system shall allow physicians to record notes regarding a patient.
21. The system shall allow physicians to prescribe medication for a patient.
22. The system shall allow the staff to pull up prescription orders for a patient.
23. The system shall allow a staff member to accept payments for services provided to a patient.
24. The system shall allow a staff member to accept payments for medication sold to a patient.
25. The system shall allow a staff member to accept payments for supplies (i.e. bandages, etc.).
26. The system shall track pharmacy inventory.
27. The system shall track supply inventory.

28. The system shall have the ability to generate and print reports on pharmacy inventory.
29. The system shall have the ability to generate and print reports on supplies inventory.
30. The system shall have the ability to automatically generate weekly pharmacy inventory reports.
31. The system shall have the ability to automatically generate weekly clinical supply inventory reports.
32. The system shall allow for the generation of clinical activity reports.
33. The system shall allow for the generation of clinical income reports.
34. The system shall have the ability to automatically generate weekly activity reports.
35. The system shall have the ability to automatically generate weekly income reports.

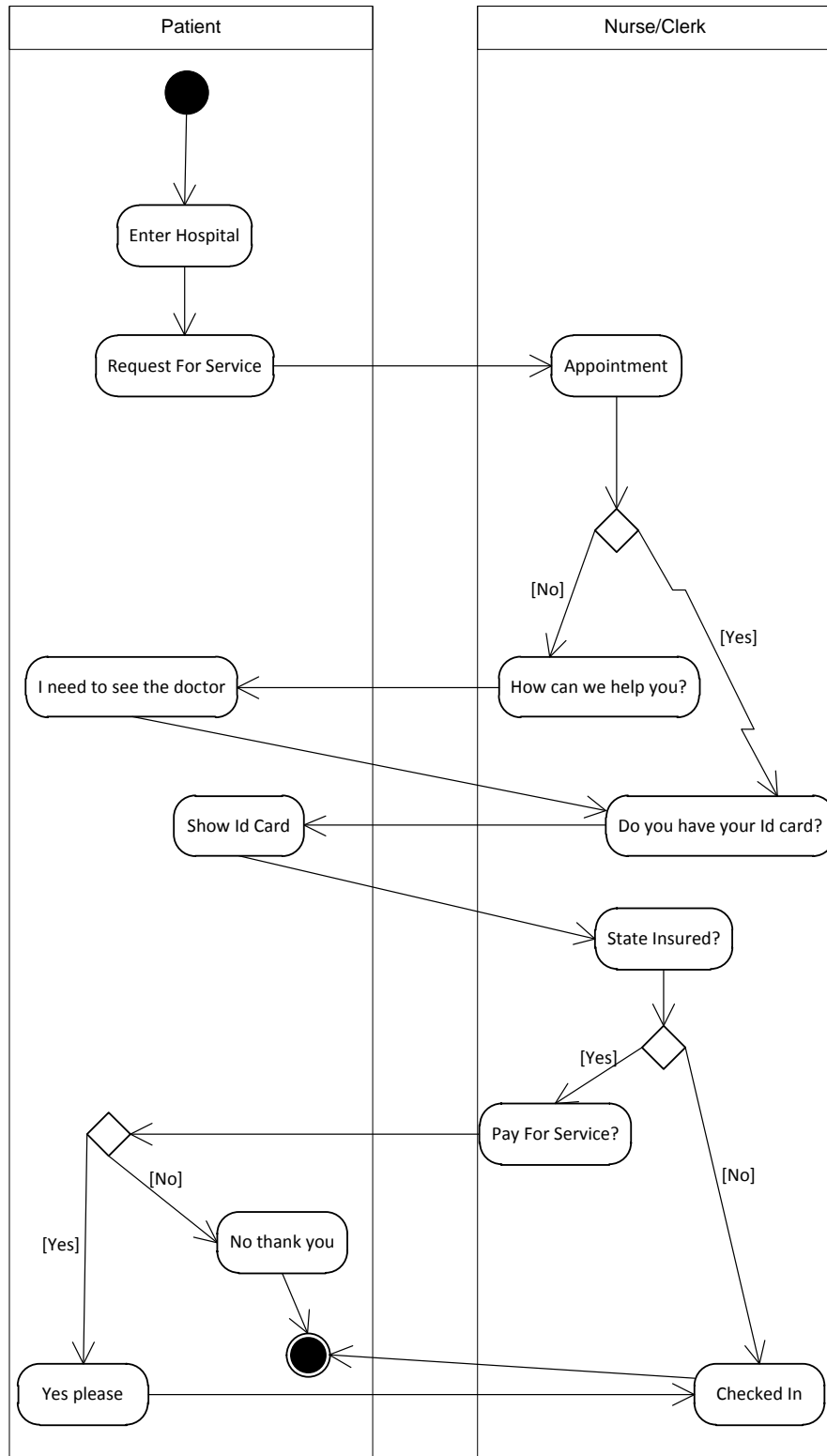
Non-Functional Requirements

1. The system shall support different security roles and permissions for the physicians, nurses, and clerical staff.
2. The system shall be designed as an *n-tier* architecture for scalability.
3. The system shall have a *database* that will be used for information storage.
4. The system shall provide a server used to store *binaries* and related data.
5. The system shall be reliable; crashes and critical errors will be rare or non-existent.
6. The system shall be easy for non-technical users to learn and use.
7. The system shall respond quickly, without *lag*.
8. The system shall have measures for ensuring data integrity in the case of *environmental* or *hardware failures*.
9. The system shall be designed to work in a networked environment of at least two computers.
10. The system shall have the ability to scale up to at least 10 *client computers*.

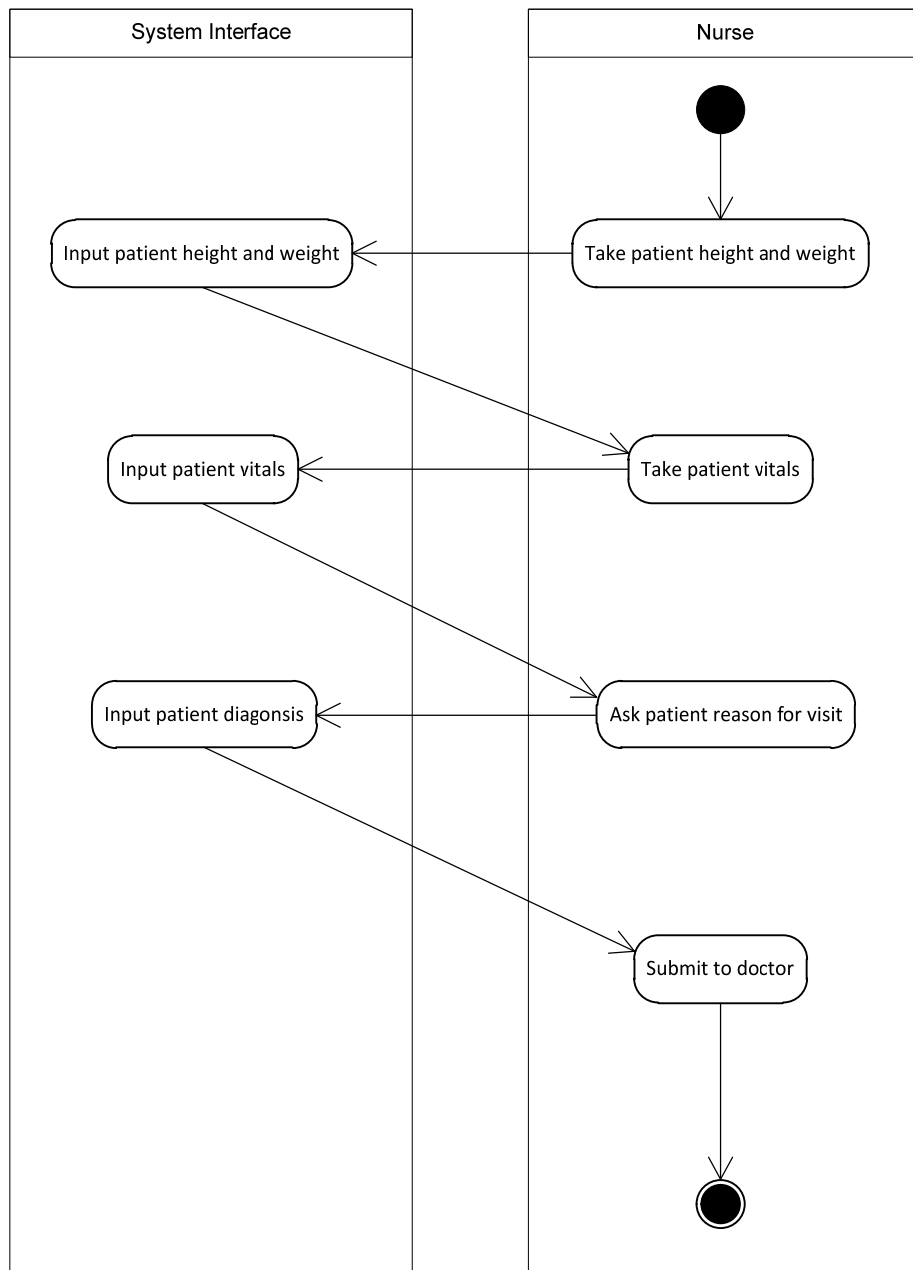
11. The system shall be compatible with an *operating system* of Windows XP or greater.
12. The system shall create a *backup* each day.

Interactions Diagrams

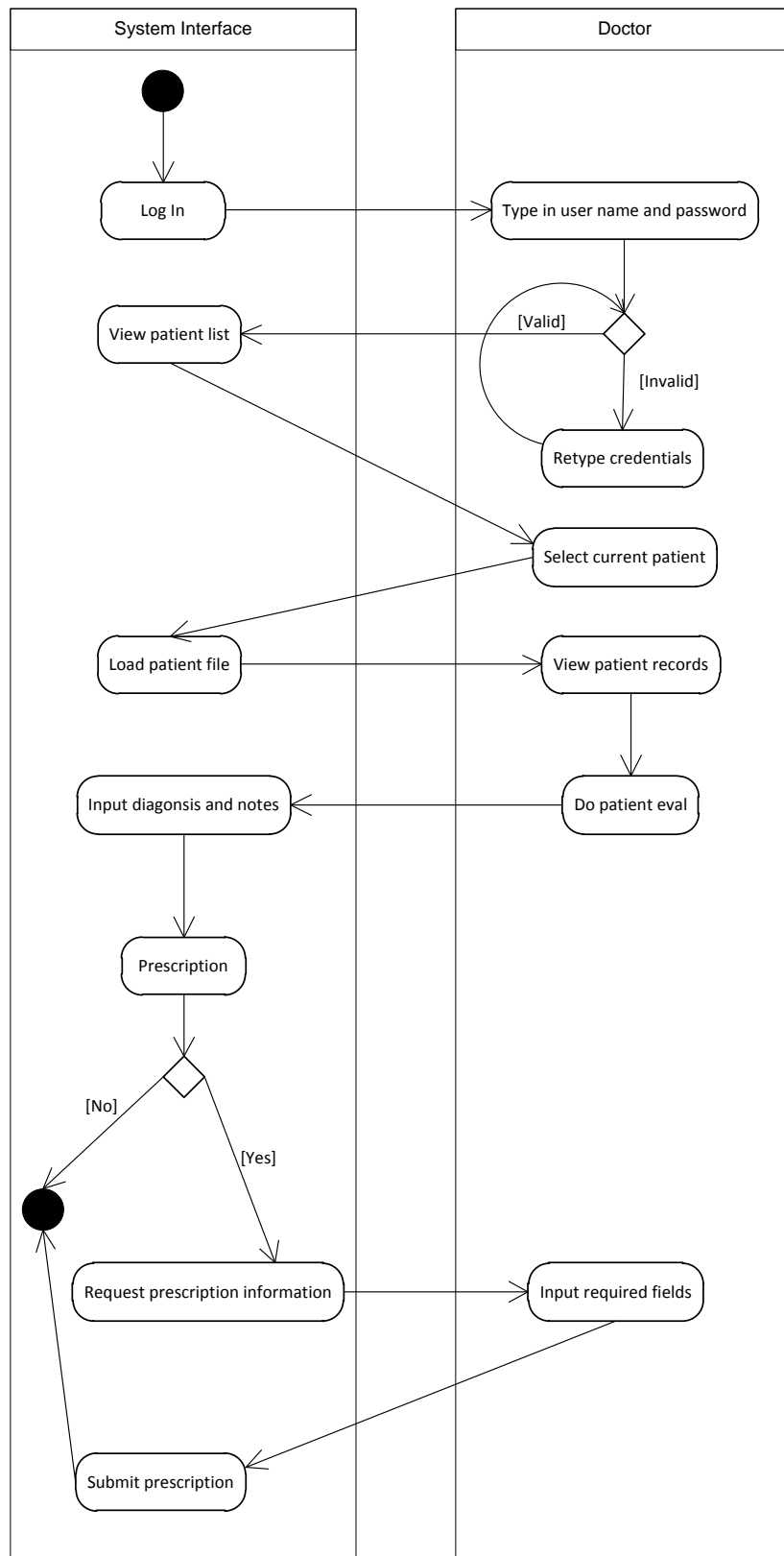
Registering a Patient



Record Patient's Vitals

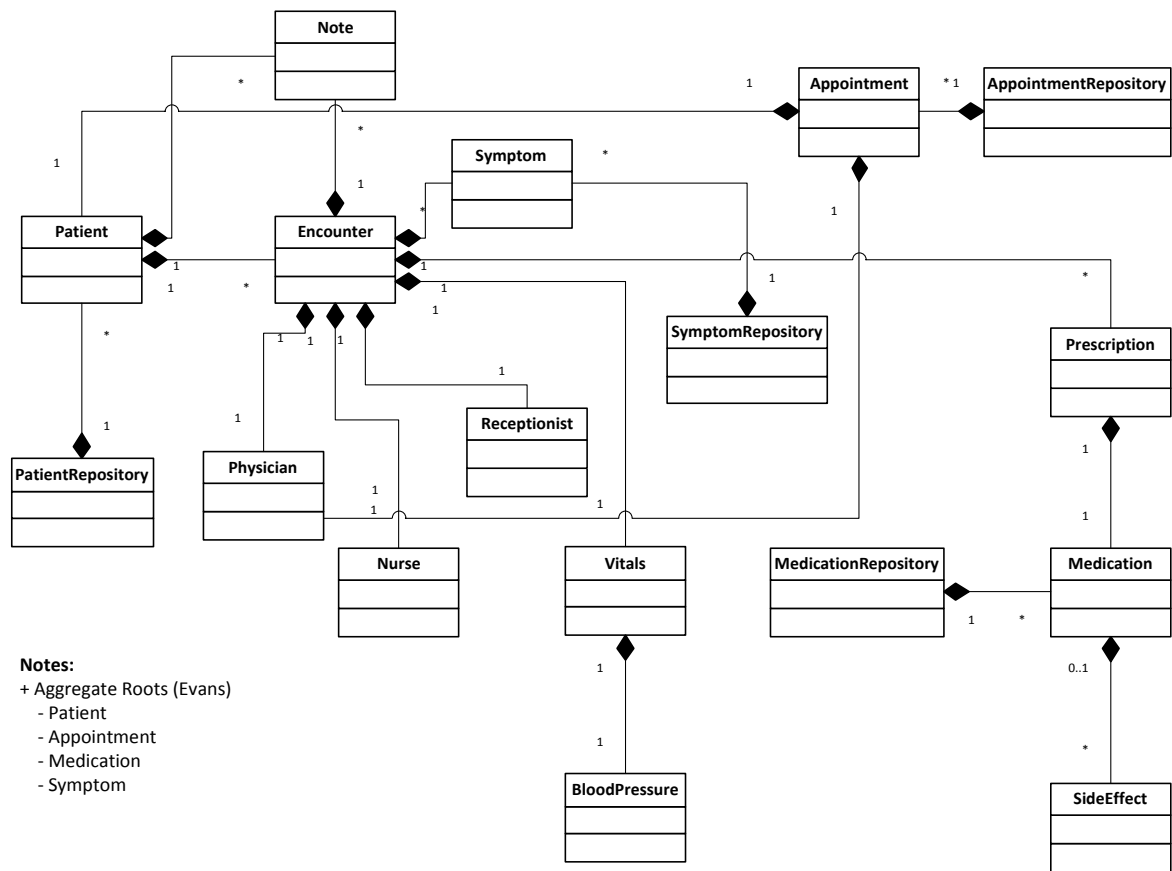


Visit with Physician



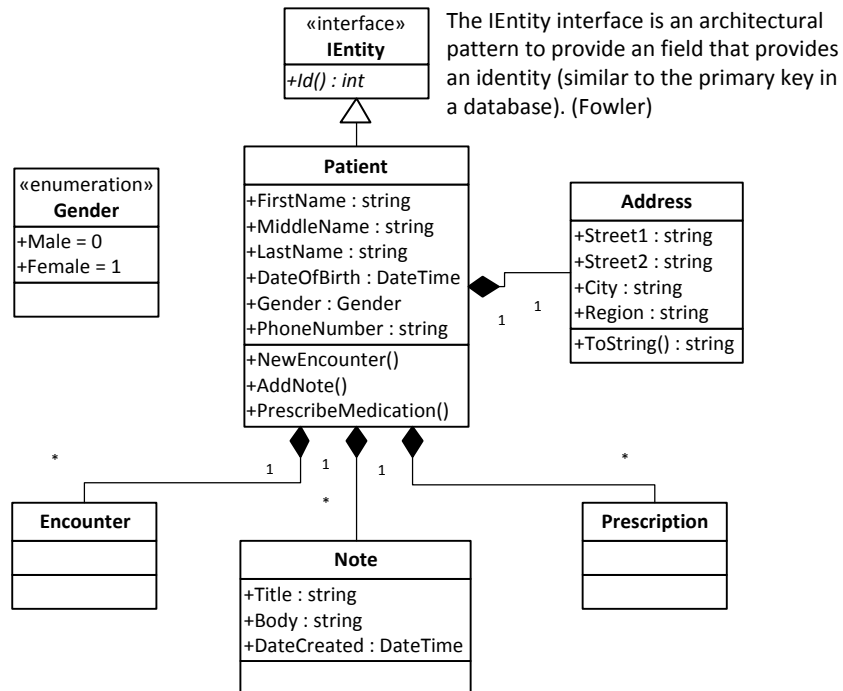
Class Diagrams

Class Diagram

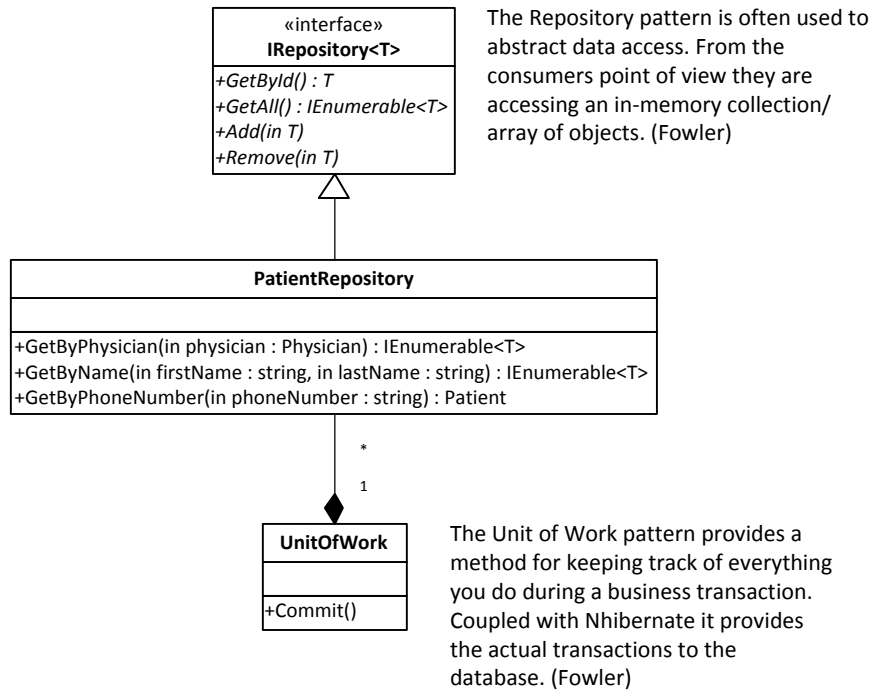


Data Types and Operation Signatures

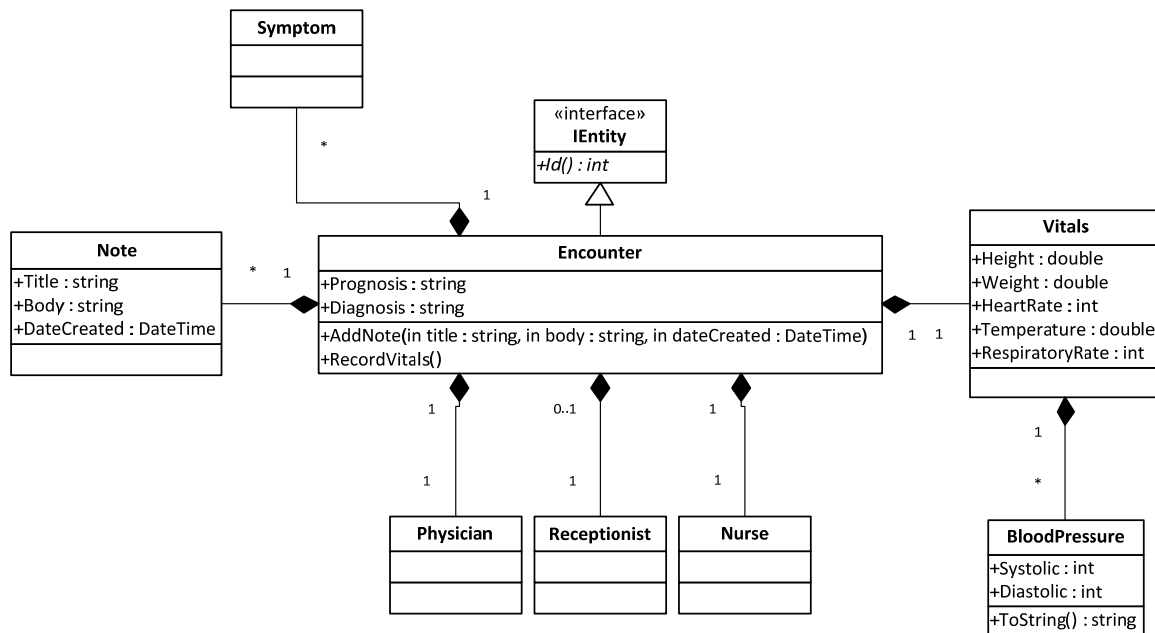
Patient Class:



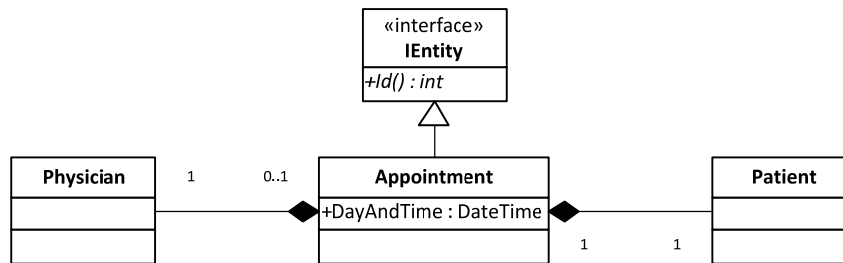
PatientRepository Class:



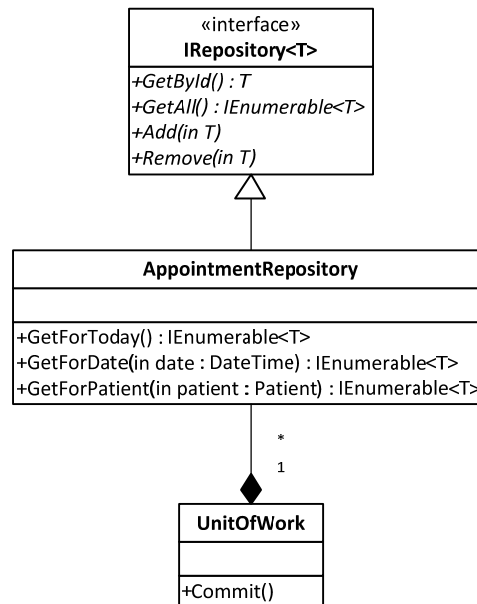
Encounter Class:



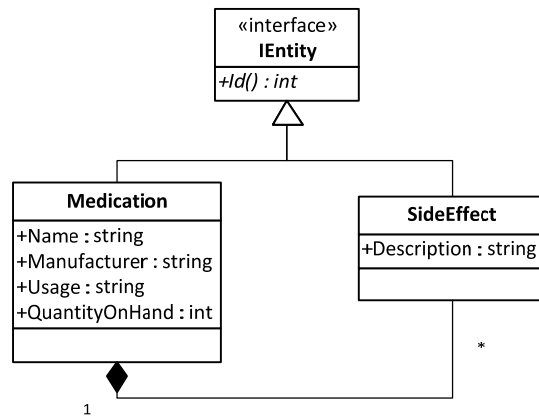
Appointment Class:



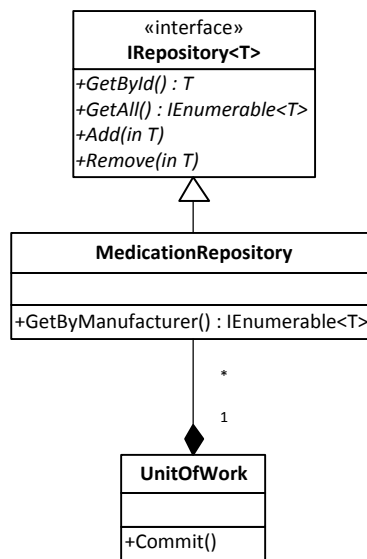
AppointmentRepository Class:



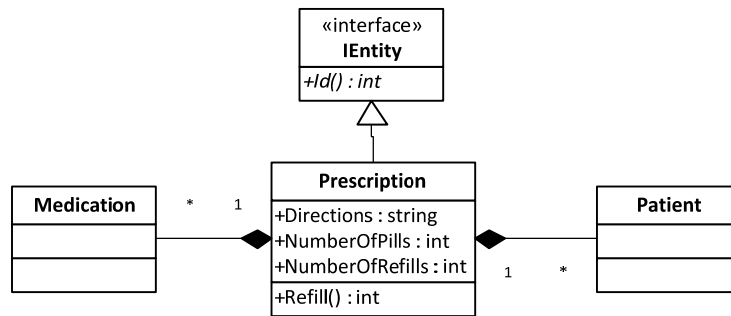
Medication Class:



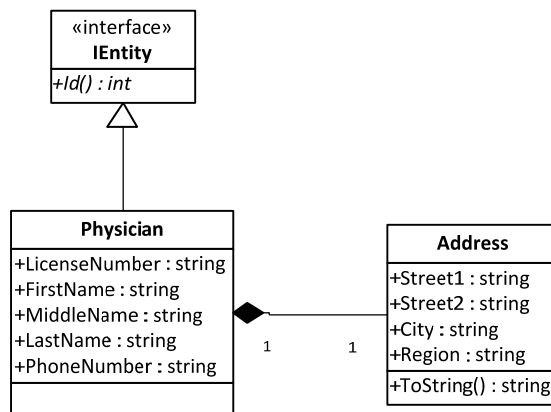
MedicationRepository Class:



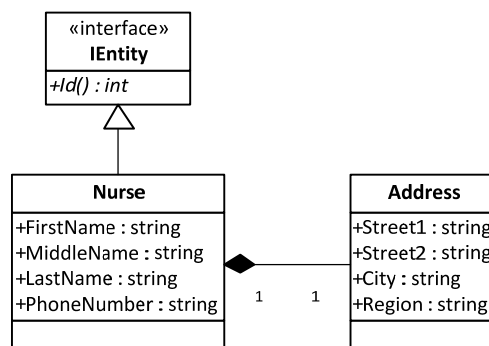
Prescription Class:



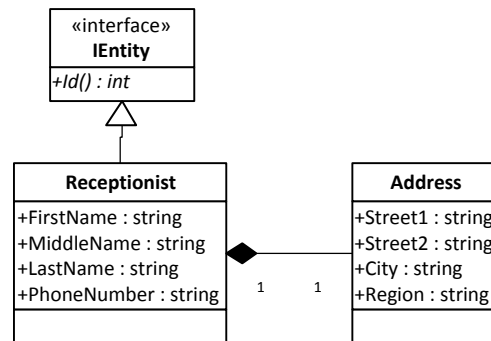
Physician Class:



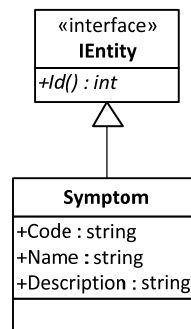
Nurse Class:



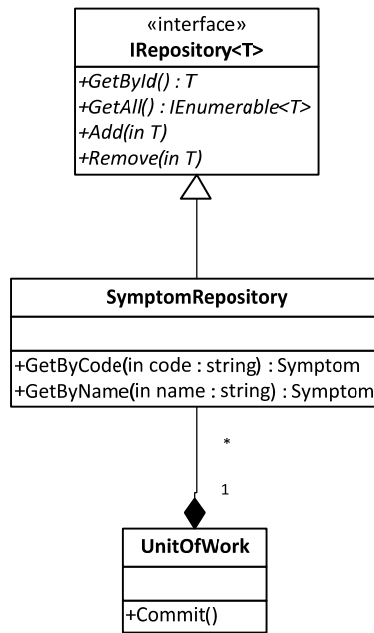
Receptionist Class:



Symptom Class:



SymptomRepository Class:



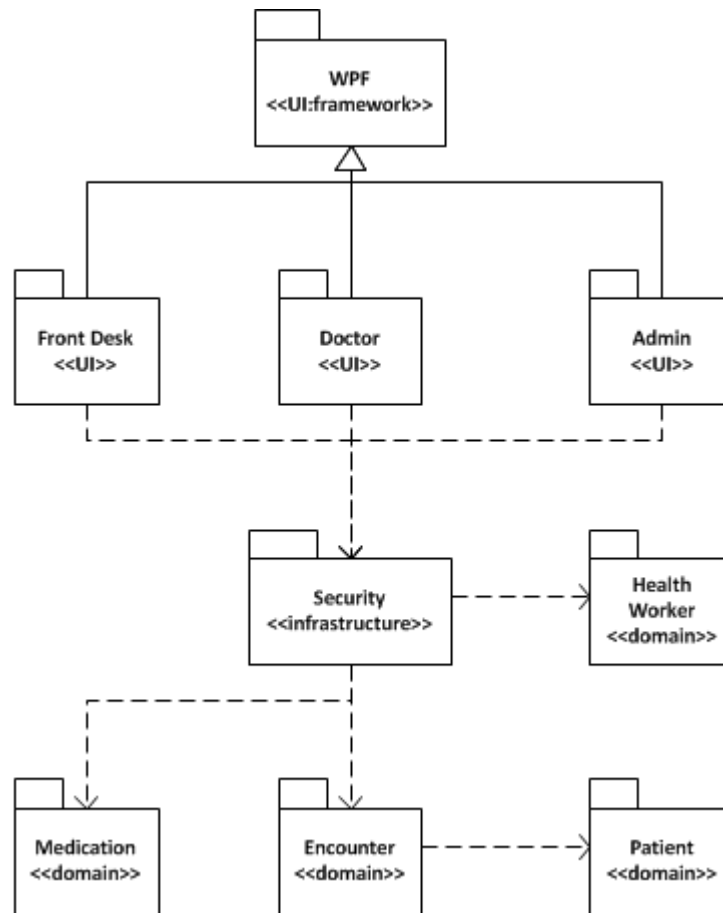
System Architecture and System Design

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

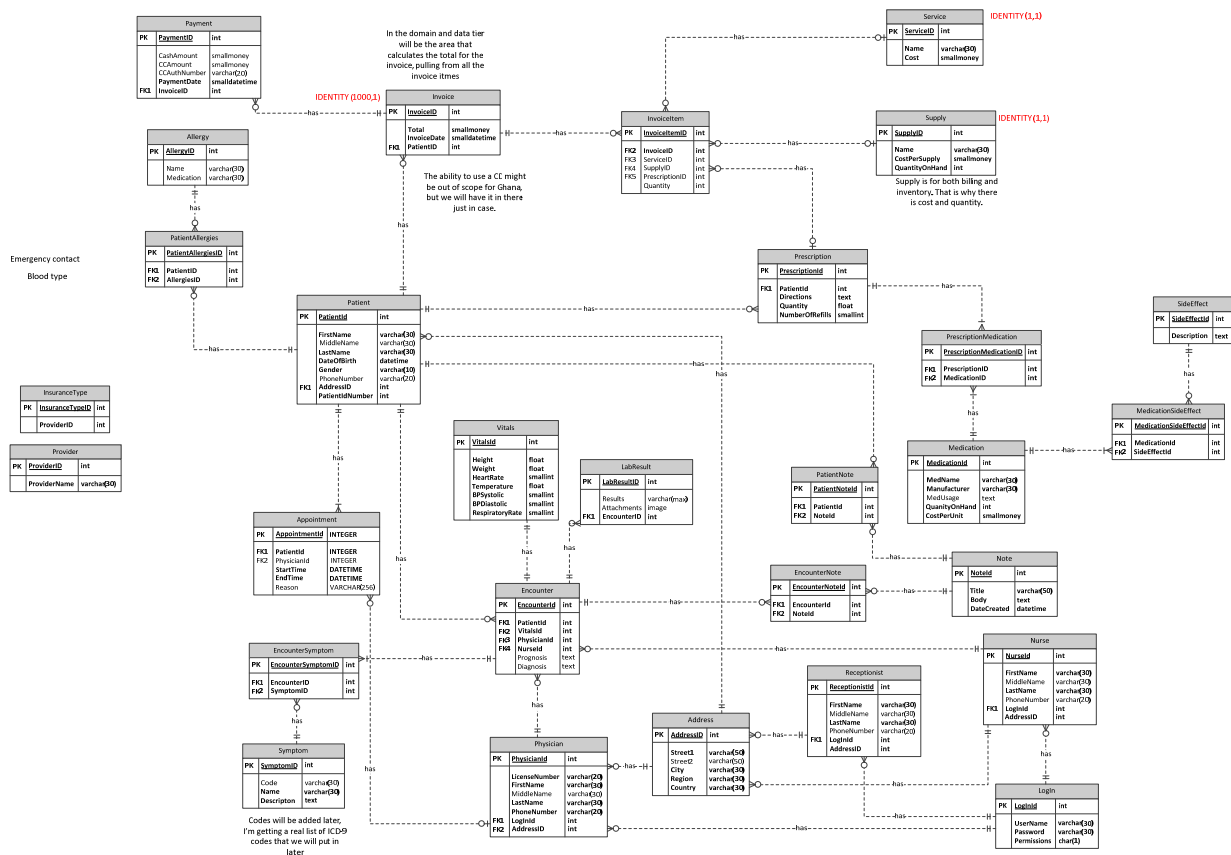


The MLKMC-EHS system is divided into subsystems as pictured above. There are three distinct user interface packages, each containing a set of files and resources to allow user interaction. Domain packages contain all other classes for the system, with closely related classes grouped together.

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.

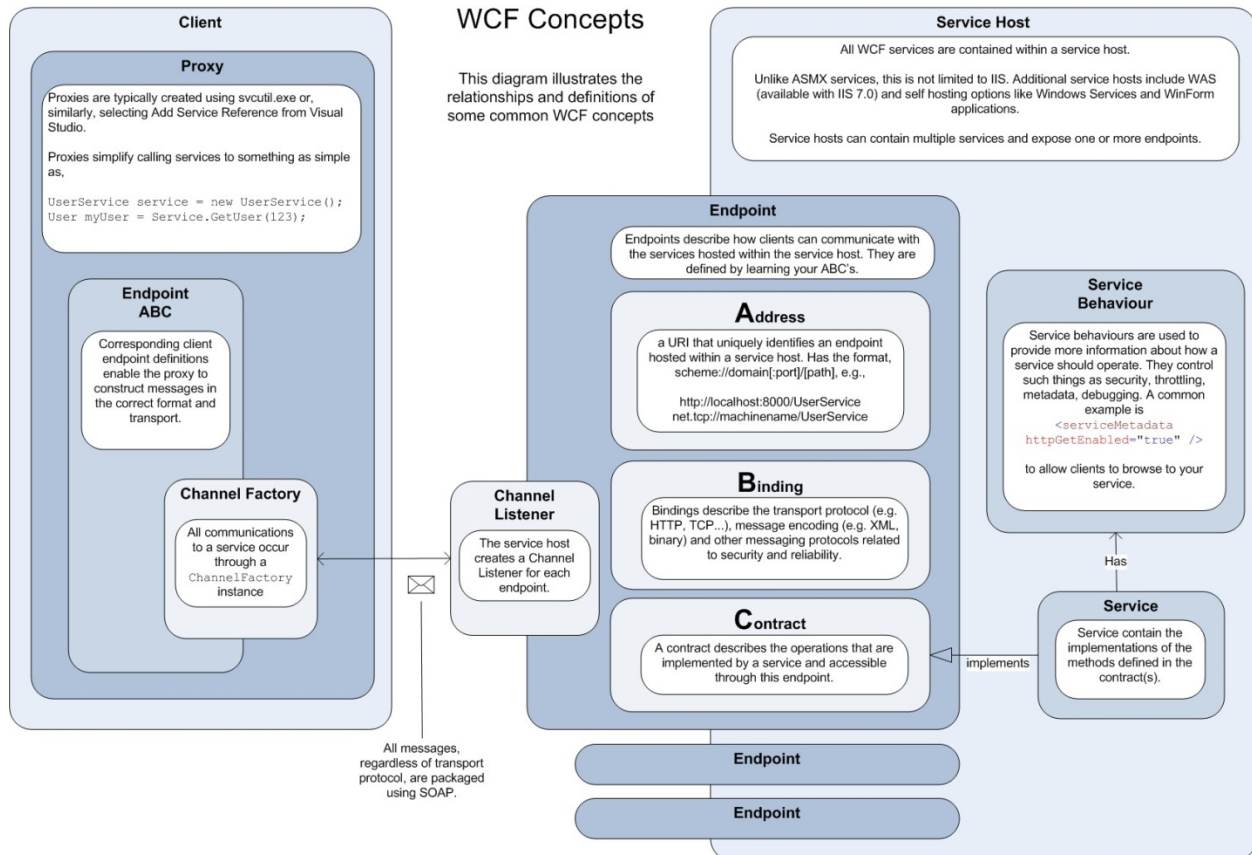
Persistent Data Storage



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.

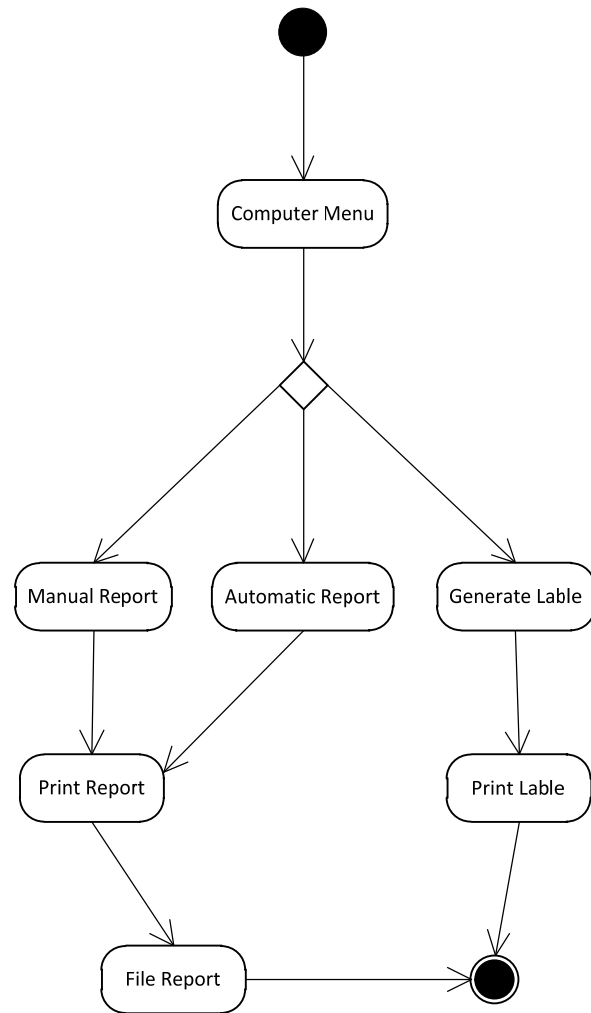
The following items are dependencies of the system to be developed and required in order to run properly:

- 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.

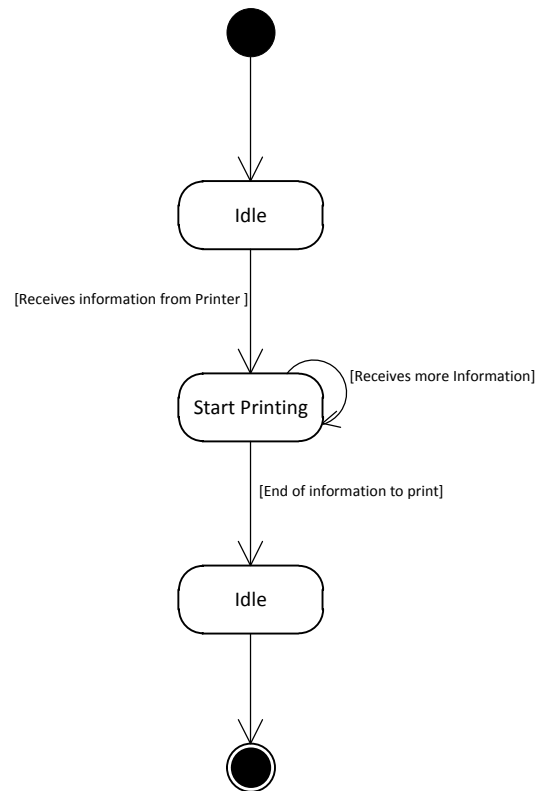
Algorithms and Data Structures

Algorithms

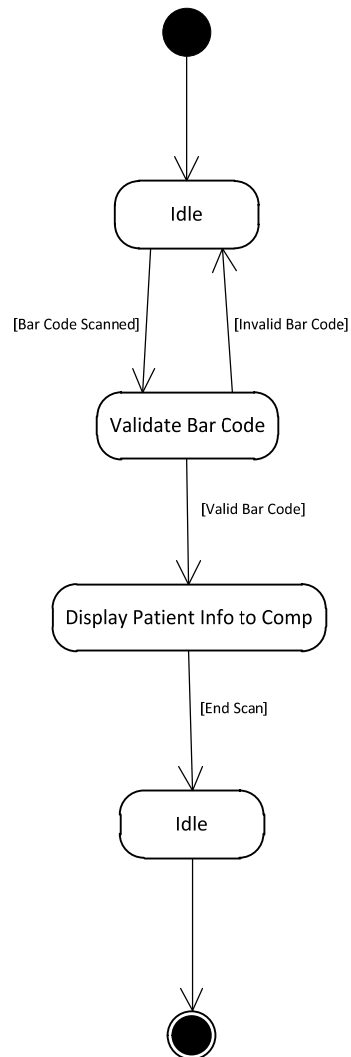
Inventory System:



Printer System:



Scanner System:



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:

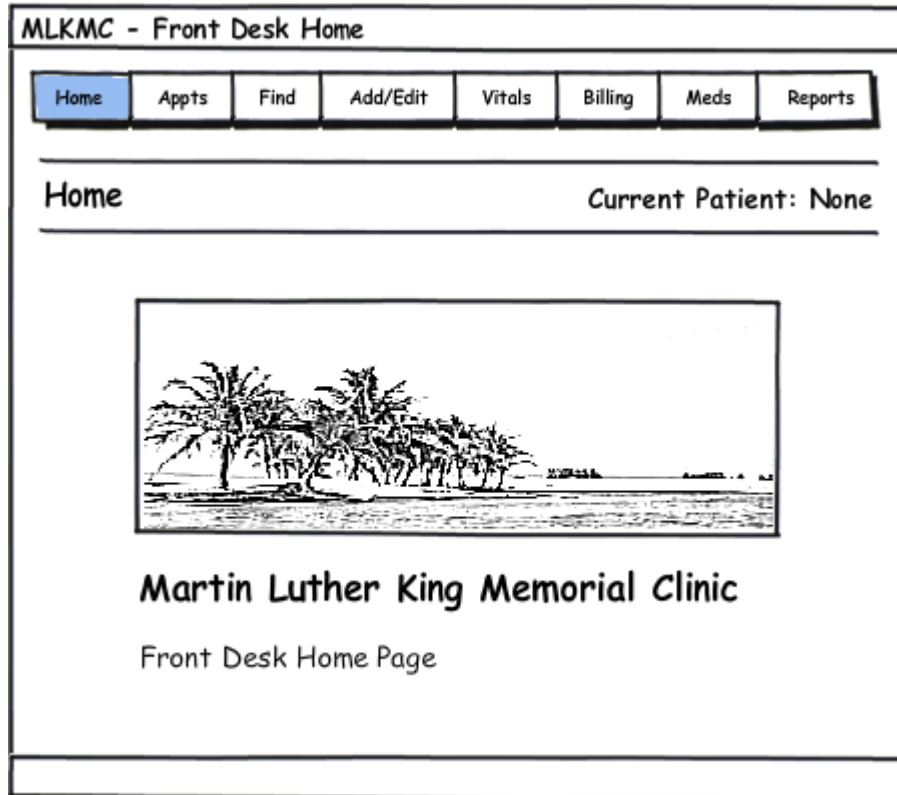
- `List<T>`
- `ObservableCollection<T>`
- `Dictionary<TKey, TValue>`
- `SortedDictionary<TKey, TValue>`
- `HashSet<T>`
- As well as classes that implement the `IEnumerable<T>` interface.

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 Design and Implementation


Add Patient to System



1. From the Front Desk Home Page, click the “Add/Edit” tab on the top of the screen.

MLKMC - Add/Edit Patient							
Home	Appts	Find	Add/Edit	Vitals	Billing	Meds	Reports
Add/Edit Patient				Current Patient: None			
<div>Add New Patient</div> <div>Edit Patient Info</div>							

2. Click on "Add New Patient".

MLKMC - Add New Patient							
Home	Appts	Find	Add/Edit	Vitals	Billing	Meds	Reports
Add New Patient				Current Patient: None			
Sex: <input type="radio"/> Male <input type="radio"/> Female							
First Name:		<input type="text"/>					
Last Name:		<input type="text"/>					
Date of Birth:		<input type="text" value="/"/> <input type="text" value="/"/> <input type="text"/>				<input type="button" value="Create Patient"/>	
Telephone:		<input type="text"/>					
House Number:		<input type="text"/>					
Area:		<input type="text"/>					
City:		<input type="text"/>					

3. Fill out all the patient information.
4. Click "Create Patient" to finish adding a new patient.

Search for Patient

MLKMC - Find Patient

Home

Appts

Find

Add/Edit

Vitals

Billing

Meds

Reports

Find PatientCurrent Patient: None

Search by card number:

OR Search by name:

Search

Recent
Patients

Name	Age	Address
John Smith	25	1234 Oak St
Sam Smith	54	4567 Cherry Rd
Gary Smith	32	7890 Pine St

Select Patient

1. Click the “Find” tab to reach the Find Patient screen.
2. Enter a card ID number OR enter a first or last name.
3. Click “Search” to bring up matching results.
4. Highlight a patient by clicking the corresponding row.
5. Click “Select Patient” to select the patient.

MLKMC - Confirm Patient							
Home	Appts	Find	Add/Edit	Vitals	Billing	Meds	Reports
Confirm Patient				Current Patient: John Smith			
<div><p>The current patient is:</p><p>John Smith Age 25 1234 Oak St Suburb A, Accra 555-555-1234 Card #: 123456</p></div>				<p>Any function may now be used applying to the current patient selected.</p> <p>Not the right patient? Go Back</p>			

6. The current patient is now set to the user's selection. All system tasks, when performed, will be applied to the current patient listed on the upper portion of the screen.
7. If the wrong patient was selected, click "Go Back" to return to the "Find Patient" screen.

Fulfill an Appointment

MLKMC - Appointments

Home Appts Find Add/Edit Vitals Billing Meds Reports

Appointments Current Patient: None

Create New Appointment

View Appointments

1. Click the “Appts” tab to reach the Appointments screen.
2. Click “View Appointments”.

MLKMC - Appointments

Home

Appts

Find

Add/Edit

Vitals

Billing

Meds

Reports

View Appointments

Current Patient: None

View appointments for date:

/

/

Show

Name	Card Number	Date	Time
John Smith	111111	12/11/2010	11:15am
Jane Doe	121212	12/11/2010	12:30pm
Rich Jones	454545	12/11/2010	2:00pm

Select Patient

Cancel Appointment

- Today's unfulfilled (to be seen) appointments are automatically displayed. To view another day's appointments, enter the date and click "Show".
- Click on a patient and then click "Select Patient" to take the patient off the unfulfilled appointments list. This also sets the current patient for other tasks to apply to.

Take Patient Vitals

MLKMC - Take Vitals							
Home	Appts	Find	Add/Edit	Vitals	Billing	Meds	Reports
Take Vitals				Current Patient: John Smith			
Height: ____							
Weight: ____							
Blood Pressure: ____							
Heart Rate: ____							
Respiratory Rate: ____							
Tempurature: ____							
<div>Reason for visit:</div>					<div>Submit</div>		

1. Click the "Vitals" tab to reach the Take Vitals screen.
2. Enter all the vital statistics, pressing Tab or clicking to reach the next field.
3. Enter a brief description of the reason for the patients visit if necessary.
4. Click "Submit" to save the information, which the physician may see now from his computer.

Issue Medications

MLKMC - Medications

Home

Appts

Find

Add/Edit

Vitals

Billing

Meds

Reports

MedicationsCurrent Patient: John Doe

Medications Prescribed

Drug	Type	Quantity	Cost	Date Issued
Aspirin	Pill	10	2.25	2/11/2010 (Today)
Tetracycline	Pill	5	3.50	2/11/2010 (Today)

Issue and add selected to bill

Issue and add all medications to bill

1. Click the “Meds” tab to reach the Medications screen.
2. A list of medications prescribed by the physician will be listed.
3. Click on a medication to highlight it, then click “Issue and add selected to bill” after medication has been filled. This takes the quantity of drugs out from the inventory, and also adds the cost of the drugs to the patient’s bill to be paid.
4. Or click “Issue and add all medications to bill”.

Bill Patient

MLKMC - Billing

Home

Appts

Find

Add/Edit

Vitals

Billing

Meds

Reports

BillingCurrent Patient: John Doe

Total patient bill: 12.00
Today's bill: 12.00
Unpaid bills: 0.00

Pay Full Amount

Partial Payment

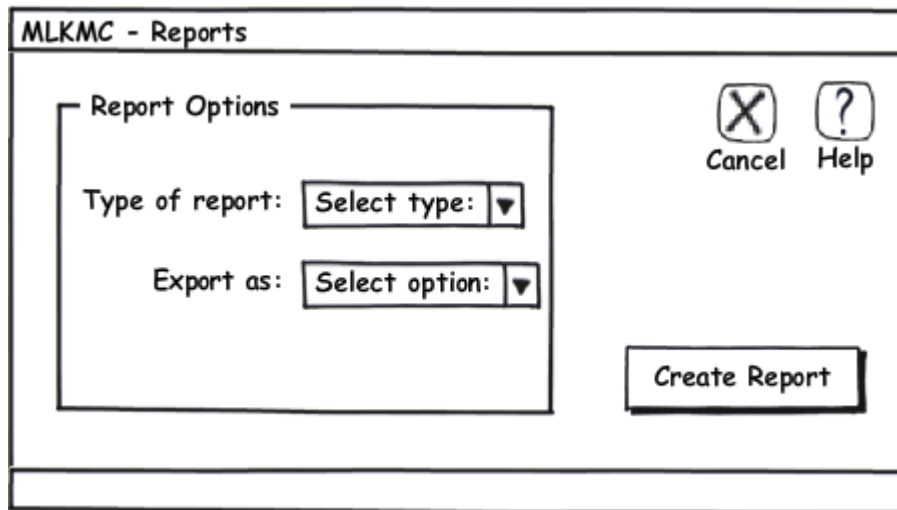
Pay Partial

Bill Details

Description	Date	Cost
Patient Care	02/11/2010	6.25
Aspirin	02/11/2010	2.25
Tetracycline	02/11/2010	3.50

1. Click the "Billing" tab to reach the Billing screen.
2. Click "Pay Full Amount" if the patient has the money to pay the total bill.
3. Or enter an amount for partial payment, and click "Pay Partial".

View Pharmacy Inventory Report



The image shows a dialog box titled "MLKMC - Reports". Inside, there is a section labeled "Report Options" which contains two dropdown menus: "Type of report:" with a "Select type:" label and a downward arrow, and "Export as:" with a "Select option:" label and a downward arrow. To the right of the "Report Options" section are two buttons: "Cancel" (with an 'X' icon) and "Help" (with a '?' icon). Below the "Report Options" section is a "Create Report" button.

1. Click the “Reports” tab to reach the Reports screen.
2. Choose “Pharmacy Inventory Report” from the drop down list.
3. Choose a desired export option.
4. Click “Create Report”.

The report generated will be in the following form.

Pharmacy Inventory

As of 06/11/2010

Name	Type	Remaining	Last Ordered
Aspirin	Pill	22	11/10/2010
Tetracycline	Pill	45	Never
Penicillin	Pill	64	19/10/2010
Ampicillin	Pill	12	Never
Quinacrine	Shot	5	16/09/2010

View Pharmacy Sales Report

MLKMC - Reports

Report Options

Type of report: Select type: ▼

Export as: Select option: ▼

Cancel Help

Create Report

1. Click the “Reports” tab to reach the Reports screen.
2. Choose “Pharmacy Sales Report” from the drop down list.
3. Choose a desired export option.
4. Click “Create Report”.

The report generated will be in the following form.

Pharmacy Sales

01/10/2010 - 31/10/2010

Total Sales: €30.00

Date	Patient	Drug	Quantity	Sale Amount
02/10/2010	William Smith	Tetracycline	10	€4.50
03/10/2010	Julie Roberts	Aspirin	8	€1.25
03/10/2010	Julie Roberts	Ibuprofen	5	€0.75
05/10/2010	Sam Jackson	Penicillin	15	€3.00

View Clinic Income Report

MLKMC - Reports

Report Options

Type of report: Select type: ▼

Export as: Select option: ▼

Cancel Help

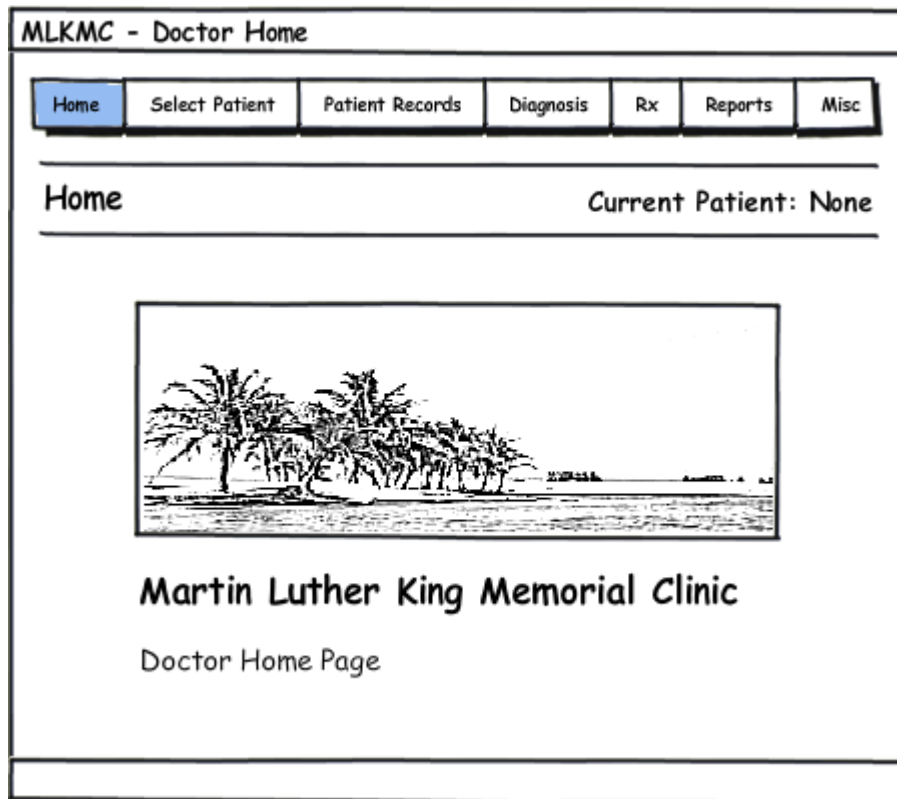
Create Report

1. Click the “Reports” tab to reach the Reports screen.
2. Choose “Clinic Income Report” from the drop down list.
3. Choose a desired export option.
4. Click “Create Report”.

The report generated will be in the following form.

<h2>Clinic Income</h2>			
01/10/2010 - 31/10/2010			
Total income from patients: €80.25			
From patient care: €50.25			
From Rx sales: €30.00			
Date	Name	Income Type	Amount
02/10/2010	William Smith	Care	€5.50
02/10/2010	William Smith	Rx	€4.50
03/10/2010	Julie Roberts	Rx	€2.00
05/10/2010	Thomas Jones	Care	€7.25

Select Patient to See



1. From the Physician Home Page, click the "Select Patient" tab on the top of the screen.

MLKMC - Select Patient

Home

Select Patient

Patient Records

Diagnosis

Rx

Reports

Misc

Select Patient

Current Patient: None

Patients Waiting

Name	Time Admitted	Reason For Visit
Jane Doe	11:30am	Stomach pains and fever
Steve Wilson	11:39am	Return checkup
Ashley Jenson	11:42am	Treatment for condition

Select Patient

Search Other Patients

- The patient waiting the longest amount of time will be shown and selected automatically on the top of the list.
- Another patient may be highlighted for selection by clicking the row corresponding to their name.

Click "Select Patient" to select the highlighted patient, and begin using other system functions on them as the Current Patient.

View/Add Patient Records

MLKMC - Patient Records

Home Select Patient **Patient Records** Diagnosis Rx Reports Misc

Patient Records Current Patient: Jane Doe

Records

- 27/11/2010
- 21/08/2010**
- 16/01/2010
- 07/09/2009
- 28/02/2009

Jane Doe - 21/08/2010

Notes:
Patient notes displayed here...

Treatment:
Treatment description goes here...

Prescriptions:
Prescriptions displayed here...

Add New Record **Edit Record** **Delete Record**

1. Click the “Patient Records” tab to reach the Patient Records screen.
2. Click the record to view from the list of dates.
3. To add a new record, click “Add New Record”.
4. Enter notes for each section of the patient record.
5. Click “Save Record” to add a new record on today’s date for the patient.

View/Make Diagnosis

MLKMC - Diagnosis

Home

Select Patient

Patient Records

Diagnosis

Rx

Reports

Misc

DiagnosisCurrent Patient: Jane Doe

Add new diagnosis:

Choose or type a condition ▼

Add

Previous diagnosis

Condition	Date Diagnosed	Cured (y/n)
Strep Throat	30/09/2010	No
Influenza	06/08/2010	Yes
Broken Wrist	15/01/2009	Yes

Remove Diagnosis

Mark Cured

1. Click the “Diagnosis” tab to reach the Diagnosis screen.
2. Any previous diagnosis can be removed, or toggled between cured and not cured with the two lower buttons.
3. To select a new diagnosis, choose a condition or disease from the list, or type the name of the condition or disease if it is not in the list.
4. Click “Add” to add the selected diagnosis to the patient’s record.

Prescribe Medicine

MLKMC - Prescriptions

Home Select Patient Patient Records Diagnosis **Rx** Reports Misc

Prescriptions Current Patient: Jane Doe

Prescribe: Select Medicine: ▼

Quantity: 1 ▲ ▼

Refill Date: / /

Issue Prescription

Past Prescriptions

Drug	Date Issued	Quantity	Refill Date
Aspirin	21/09/2010	12	21/10/2010
Penicilin	16/07/2010	6	n/a

1. Click the “Rx” tab to reach the Prescriptions screen.
2. Select a medicine to prescribe from the clinic inventory by clicking In the drop down list.
3. Choose a quantity of the medicine to prescribe.
4. Select a refill date for the prescription.
5. Click “Issue Prescription” to issue the prescription to the patient. The front desk will now be able to see this prescription when the patient comes to receive it.

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.

MLKMC - Electronic Healthcare System

Schedule

Dr. Duah

Dr. Ankrah

Dr. Oti

8:00	Consultation - Ababuo Darko		
8:30		Follow Up - Baako	
9:00		Consultation - Ababuo	
9:30			
10:00	Consultation - Esi		
10:30		Follow Up - Enam	
11:00			Physical - Gyasi
11:30			
12:00	Lunch		
12:30			
13:00			Follow Up - Dofi
13:30	Consultation - Boahinmaa		
14:00			
14:30			

October 2010

Su	Mo	Tu	W	Th	Fri	Sa
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

New Appointment

Cancel

Reschedule CTRL+R

The user can right-click on an appointment and either cancel or reschedule.

Rescheduling an appointment should also be available via the drag-drop action.

History of Work & Current Status of Implementation

So far, the team has partially implemented a few of the main components needed in the final project. The two features that were primarily focused on in the prototype were:

- Patient Management
- Appointment Scheduling

In the patient management screen exists the option to search or create a new patient in the system. Once the user has searched and obtained a list of patients, they are able to select the patient and either edit or delete the record from the system. The team implemented a feature in the search functionality that exhibits the behavior of instant results as the user types, akin to Google's "Google Instant". (Google) Within the Appointment Scheduling screen the team implemented the ability to add, edit, and cancel appointments for a given day. When scheduling an appointment the user chooses an attending physician, the start time, the end time, and the patient to be seen. The user interface also allows the user to enter a brief note about the reason for the visit to help the physicians and nurses prepare for the scheduled appointment.

The milestones set by the team were mostly met on-time with the exception of some fairly major modifications mid-way through the first iteration. Unfortunately, some of the design had to be modified to reach the goal of a functioning prototype. These modifications are only temporary and will be replaced in future iterations of the software.

Conclusions and Future Work

During the process of implementing some of the components that were designed this semester into a prototype had its fair share of technical challenges. First of all, the team decided to use NHibernate, the repository pattern, and domain driven development to simplify CRUD (create, read, update, and delete) operations with the database and make interacting with the domain entities as simple as possible. (Nilsson)

When implementing the repository pattern and NHibernate, there was a steeper than expected learning curve that we battled to overcome for the first prototype. Using both of these patterns has a lot of advantages, but the disadvantage of using them was the implementation with having very little team experience using either. However, once the fundamentals for using these “technologies” are learned and put into action, dealing with the interaction between the application and the database will be fairly easy and makes the development process a lot cleaner in the end. (Maulo)

Another technical challenge that arose during the development process was, once again, the steep learning curve of a particular technology. This time it was the windowing framework WPF (Windows Presentation Foundation). Our initial foray into this space was plagued with data binding problems due to our own inexperience. However, our approach to this part of the application, the GUI, is shifting rather radically. For instance we will be implementing a presentation pattern called Model-View-ViewModel. (Smith) This pattern allows for the decoupling of a particular visualization and the logic that drives it. It also has the potential to easily allow the team to implement modular development of GUI screens and avoid breaking existing functionality in other parts of the application. And, last but not least this pattern allows the dynamic loading and unloading of “screens” or “views” from the GUI at runtime.

The software development methods that have been learned from this class have helped us gain an understanding of important processes that one must go through in the requirements gathering and design phases to have an organized and successful project. Some of the techniques that helped address these changes were things like: how to interact between modules is approached, unit testing to verify business logic, application of patterns, and the use of alternate approaches to finish the product on-schedule.

With the technical problems we encountered, we ended up having to go a different route in the data tier of our application. Our initial design went from implementing NHibernate and the repository pattern to a simple data layer performing CRUD operations. This approach provided us with a solid, yet limited, base for creating the prototype with some of the key components to be demonstrated.

In the future this project can go one of many different ways to be completed. First, the project can be refactored to go with the original plan of using NHibernate, the repository pattern and domain driven development and proceed from there into a fully functional system that would be easy to add onto or modify because of the simplicity and decoupling of the components once these patterns are in use. Another direction that could be taken would be to proceed in a pragmatic fashion forgetting future maintenance and use the a simple, coupled n -tier project and use the existing prototype as a starting point that anyone could add onto and have a head start in implementing the final system. A disadvantage of that route, that we've found, when dealing with the simple n -tier approach with a project as big as this is the complexity of the data access when dealing with the many entities that the database contains. For each entity, you are required to have a get, update, create, and delete function which when you account for dependencies turns into a nasty web of connections that is hard to debug and implement and when you have many entities this requires a large amount of coding. The latter approach also compels developers to break many of the SOLID principles of good object oriented design as described by Martin. (Martin)

The team has found it beneficial to take a route that would benefit future brownfield developers of the project and practice good object oriented design and industry best practices. Not only do these techniques improve the maintainability of the project in its entirety, but we, as a team, are also allowed the opportunity of getting a head start on many of our peers in these methods.

Bibliography

Fowler, Martin. Patterns of Enterprise Application Architecture. New York: Addison-Wesley Professional, 2003.

Google. About Google Instant. 6 December 2010 <<http://www.google.com/instant/>>.

Martin, Robert. Principles of OOD. 6 December 2010
<<http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>>.

Maulo, Fabiou. NHibernate Forge. 6 December 2010 <<http://nhforge.org/>>.

Nilsson, Jimmy. Applying Domain-Driven Design and Patterns: With Examples in C# and .NET. Boston: Addison-Wesley Professional, 2006.

Smith, Josh. WPF Apps With The Model-View-ViewModel Design Pattern. 1 February 2009. 6 December 2010 <<http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>>.

Software, Domain-Driven Design: Tackling Complexity in the Heart of. Evans, Eric. Boston: Addison-Wesley Professional, 2003.

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>>.

Approval Signatures

Dr. Duah

Date

Prof. Richard Fry

Date