TEAM5OIES

Team Project Report

Version 3.0

Signature Block

COSC 4351	Name	Signature	Date
SE Team Leader	Obinna Ugwuzor		
SE SQA	Edison Guevara		
SE SQA	Shah Zaib		
SE Team Leader	Obinna Ugwuzor		3-9-15
SE SQA	Edison Guevara	Edison Guevara	3-9-15
SE SQA	Shah Zaib		3-9-15
SE Team Leader			
SE SQA			
SE SQA			

Table of Contents

1. Report Format and Content

1.1 Chapter I: Introduction to *TEAM5OIES*

1.1.1 Introduction

The purpose of this project is to create a database for the surgeons to place EVAR information about their patients. The database will automatically anonymize the patient's information to protect their confidential information. Surgeons and other authorized users will be able to upload data, download image analysis, and view specific data. Once a user downloads an image, the user can perform automatic quantitative image analysis on the 3D reconstruction by automatically extracting relevant anatomical data, with 7 measurements.

1.1.2 Research Methodology

There are many software products that are slightly similar to our product, many of which are databases for other medical purposes such as *Endrov*, an Image and data viewer and editor, *InVesalius*, a 3d medical imaging reconstruction software, and many others. This product will have more capabilities as compared to the other software such as anonymizing patient's data, performing automatic quantitative image analysis, and uploading and downloading data.

1.1.3 Report Organization

Chapter II: Requirements will be discussing the main problems that could arise during the project. It focuses on the pros and cons of learning new languages versus utilizing languages that the programmers already know. In addition, it will discuss software maintenance that will allow expanding of the system as opposed to creating an entirely new system after several years.

Chapter III: Educating the User discusses the phases of software development as well as several possible problems that could arise during implementation of the project. These include software complexity, conformity, and changeability. The large size of the data files is one topic we will discuss.

? iv, v, -10

2. Chapter II: Requirements

2.1 Problem Statement

Historically, a large portion of software products have been deemed failures, whether it is due to not meeting deadlines, not finishing within the budget, or the product simply not working. A result of this was the creation of the term software engineering, in hopes that it would hold software products to the same standards as other engineering products such as bridges, planes, etc. Economically, the issue of cost must be addressed. One main aspect of cost in developing a

software product is time taken to complete. In this case, a particular technology/coding method is what can make a difference in the time taken to develop. On one hand, a team of developers might be familiar with a language that takes a certain amount of time. A new language might be much faster to develop, making the cost less to the client. However, the team has to take into consideration the time it will take to train on the new language, implement it with a few more issues at first than with the old language, and various other issues. While the new language is easier to code in, it might end up taking more time in the long run and not decrease cost overall. It could also create more issues with maintenance of the code after installation, which prolongs costs over the lifetime of the software. There is more time spent with maintenance than the actual development of code, so in the long run, it might be best to go with the technology that helps create less need for maintenance.

TEAM5OIES plans to address these issues by ensuring to have documents and deliverables finished with plenty of time to look over and change anything needed. One way to avoid historical issues is to make sure everyone has the knowledge needed and to make sure there is a strong leader over everyone. We will ensure that we have both of these. Economically, we will focus on using languages that we are familiar with in order to avoid lengthy learning curves. This will prevent the cost from being high. By avoiding languages we are not familiar with, this should allow us to develop code that will not require heavy post-delivery maintenance.

2.2 Expanding the Borders

Maintenance, as compared to delivery, typically accounts for a much greater percentage of the software production process. Since many of the software deal with variables that are constantly changing, such as tax percentages, laws, and labor costs, maintenance must be done post-delivery to keep the software up to date. This can involve heavy measures if the code isn't developed in a way that it can be changed easily after installation. Because of this, implementing measures to avoid lengthy maintenance is worthwhile cost-wise. One way TEAM5OIES can ensure this will be done is to utilize constants on variables that could change at one point in time. This will decrease the number of actual changes to be made as well as reduce the chance an error can be made during its implementation.

2.3 Expanding vs. Replacing the System

Both perfective and adaptive maintenance are essential in the software life cycle. Perfective maintenance is the maintenance done when the client thinks a change can be made to increase effectiveness of the product. This can be anything from a change in code to improve upon response time to a change in website appearance to better appeal to customers. Adaptive maintenance is done due to a change in the environment in which a software product runs. These can be changes because of change in laws, hardware changes, or various other environmental issues.

One important aspect of both types of maintenance is the development of Object Oriented code upon implementation. By using OO from the beginning, maintenance can be done by reusing code if it is implemented correctly. This decreases the time needed to complete the maintenance and thus the cost of the product. Creating code that has the ability to be expanded upon instead of code that is created without room to have any changes made or grow is a great help to a client who plans on keeping the software long term.

2.4 Concluding Remarks

The historical, economic, and maintenance aspects of software engineering are very important to consider at the beginning of development. TEAM5OIES will focus on deadlines, time of production, and creation of code that will be able to be maintained quickly and effectively in order to create a successful software product.

3. Chapter III: Educating the User The Software Process

The unified process consists of four distinct phases, inception, elaboration, construction, and transition. Inside each of these phases are various amounts of workflows including requirements, analysis, design, implementation and testing. To ensure that we meet our goals on time and with a working product, TEAM5OIES is following this process software closely by starting with understanding the product domain, working with the client to ensure our proposed solution meets their requirements, and using the object oriented paradigm to complete the construction and testing of our solution. Since we are targeting CMM 3, following the unified process allows us to meet these requirements.

3.1 Problems with Software Complexity

One of the major problems with this product will be the complexity of the scans. Not only are these large files, but they are extremely complex data structures. It's potentially difficult to do simple things such as getting an image or date from them and even more difficult to explain this complexity to a non-programmer due to the existence of programs that already do this.

3.2 Problems with Software Conformity

As referenced in the section above, these scan files we will be working with are extremely large and complex files. They also include quite a bit of extraneous information for our purposes, however because this system must conform to their existing workflow, we must keep this complexity.
Ideally, the two additional steps of analyzing the measurements and running the flow analysis software would be integrated into the system and run automatically. However, this is an extremely complex operation, and one that is far outside of our current scope of the project.

3.3 Problems with Software Changeability

It has already been established that this program is expected to last for many years (as evidenced by the fact that it's for a long-term study). It should be expected that during this duration there will be requests to change the software to better fit into the existing workflow or to conform to a new one. To assist with these inevitable demands, TEAM5OIES will be ensuring that all code is well documented and that it follows all established conventions.

3.4 Problems with Software Invisibility

In general, it is extraordinarily difficult to, with any accuracy, visually describe a software program. Of course there are UML diagrams and the like which are useful to visualize certain sections of the program, but there's no good way to visualize the entire thing at once like a 3-d model of a building could do for an architect. To combat this as effectively as possible, TEAM5OIES will be utilizing the UML constructs and diagrams as often as possible. Our plan is to create a full set of UML diagrams documenting every portion of the system so that, much like a set of blue prints for a building, all but the tiniest of details will be documented.

3.5 Concluding Remarks

While this project will certainly be a challenge, and there are many things we must consider, TEAM5OIES is confident we can succeed. Throughout the design and implementation processes we will be vigilant for issues that could affect the design through any of the above mentioned issues.

4. Chapter IV: Choosing the Right Team ("Teams" - textbook)

4.1 Problem Statement

Most software products, including ours, are too large to be made by a single person in a timely manner. Therefore, a team must be assembled to complete the project, and we must decide how this team is organized in order to ensure everything can be completed by the deadline.

4.2 Assembling the Team

4.2.1 The Democratic Approach

A democratic team is a team of up to 10 egoless programmer. Egoless programmers are programmers that have had their values restructured so that they no longer see their code as an extension of themselves, and view finding faults in their code as a normal and expected event rather than a negative thing. In this we the democratic team is able to develop an ethos, or group identity.

4.2.2 The Classical Approach (Chief Programmer)

The classical chief programmer approach allows the team to cut down on lines of communication by having all members of the team communicate with the chief programmer rather than with each other. This reduces time wasted in meetings. Additionally in this approach some of the programmers have special positions. A backup programmer is responsible for taking over for the chief programmer if he is unable to fulfill is duties and a secretary programmer takes care of all parts of the project that aren't directly programming.

4.2.3 The Modern Approach (Chief Programmer & Business Manager)

The classical chief programmer approach runs into a major issue, an expert programmer and an expert manager are unlikely to be the same person. The modern approach to the chief programmer team solves this by separating out the managerial duties of the chief programmer to a separate team manager, and rebranding the chief programmer as the team leader. Another advantage of this approach is that it is highly scalable.

4.2.4 Additional Team Approaches

Other team approaches include Synchronize and Stabalize, an approach in which several parallel teams complete different parts and then synchronize them at the end of the day. Another is Agile team, this involves pair programming as a common technique. The last is the open-

source team, this may be any number of programmers who communicate asynchronously, they have no team leaders or mangers.

4.3 Recommendation for TEAM50IES

For TEAM50IES I recommend using the modern approach. The team members have been selected in such a way that the modern approach is the ideal team organization for our team.

4.4 Concluding Remarks

At this point our team organization is set up in the way we need it to be in order to complete the software product. We should not need to modify it further.

5. Chapter V: Choosing the Right Model ("Software Life-Cycle Models" - textbook)

5.1 Problem Statement

In an ideal world software is developed perfectly in order requirements -> analysis -> design -> implementation. However in the real world programmers make mistakes and clients' requirements change. To solve this problem we must address the model we use for developing software.

5.2 B. Comparing Models

5.2.1 Waterfall Model

The waterfall model involves going through the steps of requirements -> analysis -> design -> implementation sequentially and backtracking when a fault is found to correct it at the step it occurred. This is a very disciplined and document driven approach. However, due to the model's rigid nature it is likely that the final product may not meet the client's needs.

5.2.2 Incremental Model

The incremental model involves going through the steps of requirements -> analysis -> design -> implementation simultaneously changing how much of each workflow is done at each increment. A major advantage of this model is that it very closely resembles real-world software development.

5.2.3 Extreme Programming

The Extreme programming model is a mode related to Agile processes in which features are presented with time and cost estimates to the client on each iteration (usually 2-3 weeks). The process also utilizes test-driven development and heavily involves the client in each step of the process.

5.2.4 Synchronize and Stabilize

The synchronize-and-stabilize model is a variation of the incremental model that is used by Microsoft. The requirements phase is conducted by interviewing numerous potential clients and extracting a list of features of highest priority. Then the work is divided into builds. Each day parallel teams work on different parts and then at the end of the day they synchronize the incomplete code by testing and debugging it.

5.2.5 Spiral Model

The spiral model is model very focused on minimizing risk. It does this by generating prototypes for the product and any new features, it also involves risk analysis at each step. If all significant risk cannot be mitigated at a given step then the project is terminated.

5.2.6 Open Source Model

The open source model is a far less structured model for development that has worked extremely well in a handful of circumstances, however it more typically fails. It most typically involves an individual making a first version or a product and providing it freely with source code allowing others to implement changes to it.

5.3 Recommendation for TEAM5OIES

For TEAM5OIES I recommend using an Iterative model for developing our product. This closely models the real world situation and uses the Unified Process.

5.4 Concluding Remarks

Overall we should continue our current path in our life-cycle model following each workflow of the Unified Process and adjusting as we move up increments in our product.

6. Chapter VI: Analysis and Development Methods ("The Tools of Trade" - textbook)

6.1 Problem Statement

In software development we have the choice of several different tools to implement in order to make it a more efficient and cooperative process. The tools available to software engineers can be categorized into different spectrums, which are analytical and software tools. Due to lack of first-hand experience with medical data and some of the ambiguities associated with the needs of the client, which method we choose will be critical to accomplishing the product in a timely fashion.

6.2 Conventional Engineering Methods

6.2.1 Stepwise Refinement

Stepwise refinement is defined as the means to delay decisions regarding the details of a project until as late as possible in order to concentrate on the most important issues. Stepwise Refinement is an especially powerful tool for the object oriented paradigm, since it is iterative and incremental.

6.2.2 Cost Benefit Analysis

Cost Benefit Analysis is the process of determining whether a possible course of action would be profitable when the estimated future benefits are compared with projected future costs. Assumptions are made in order to quantify otherwise intangible benefits, so that a dollar value may be assigned to it.

6.2.3 Software Metrics

Software Metrics, or software measurements, can serve as an early warning system for potential problems in the software development process. As such, software metrics can be applied to any phase or process in the development. The five essential metrics to any software process are as follows:

1. Size(ex. Lines of code, lines per module, or lines per team member)
2. Cost
3. Duration
4. Effort(in person-months)
5. Quality(number of faults detected)

6.2.4 CASE

Simplest form of CASE is the software tool, a product that assists in one aspect of the production of software. The scope of CASE is dependent on the usage during the software process, i.e. a tool used for the first one phase is only a tool, three phases is a workbench, and the entirety, or close to it, is considered an environment.

6.2.5 Software Versions

It is important to keep at least two versions of any software product; however, considering the scope of the assignment it may be better to have more versions. This

would allow us to perform adaptive maintenance to the product to ensure there is always a working version of the software.

6.2.6 Configuration Management

Configuration management is the process of controlling variations of the software product, but it can also help handle problems caused by development and maintenance by teams. Configuration management ensures that there is no synchronization issues when members are making changes for a known fault.

6.2.7 Problem Statement

There are many products available at our disposal, so it may be difficult choosing the tools that best fits our needs. Using stepwise refinement we may only need tools for specific portions in development or different phases in the overall process.

6.2.8 Other Applicable Engineering Tools

A build tool is an example of something that can be used to select the correct version of each compiled-code artifact to be linked to form a specific version of the product.

6.3 Recommendation for TEAM5OIES

I suggest we continue using SVN as our configuration management tool, Microsoft Word/Excel for our planning and documentation, and Visual Studio 2010 as our primary IDE when programming. An active and continual usage of stepwise refinement should be implemented for both coding and time management purposes.

6.4 Concluding Remarks

Overall, TEAM5OIES is in good condition regarding software tools but we could improve on our use of the analytical tools available to us. Over time a need for new software may be necessary; however, so long as we remain methodical and organized it is well within our capacity to handle any problem in the future of this product.

7. Chapter VII: Testing ("Testing" - textbook)

7.1 Problem Statement

It is not sufficient to test the product at the end of a workflow. There must be continual testing throughout the entirety of the product in order to ensure we have a work product come the deadline. The scope of the project far exceeds anything we could have anticipated, so devising a test plan for all phases of production gives our team a better chance of having a fully functional product by the end of the semester.

7.2 Quality Assurance

The quality of software is the extent of which the product satisfies the specifications; however, as developers it is also our responsibility to ensure high-

quality software at all times. This is only made possible by testing and refining the product. There are two forms of testing commonly known in development and they are non-execution based testing and execution based testing. Test cases for specific input/output responses or product functionality with a known environment are examples of execution based testing. Testing without test cases are referred to as non-execution based testing, and usually consist of reviewing the code or mathematically determining the feasibility of the code. In order to mathematically test the code the correctness proof is usually used, which is sometimes referred to as correctness proving. This is done with flowcharts referring to code segments in order to verify outputs.

7.3 Recommendation for *TEAM5OIES*

My suggestion is that we have weekly walkthrough with any lengthy new sections of code, and also have a designated partners in order to run inspections on code. Of course, we should review our code as well before submitting it; however, we are less likely to catch our own mistakes. Test cases derived from our use cases should also be designed and verified as we complete various modules which compromise the final product.

7.4 Concluding Remarks

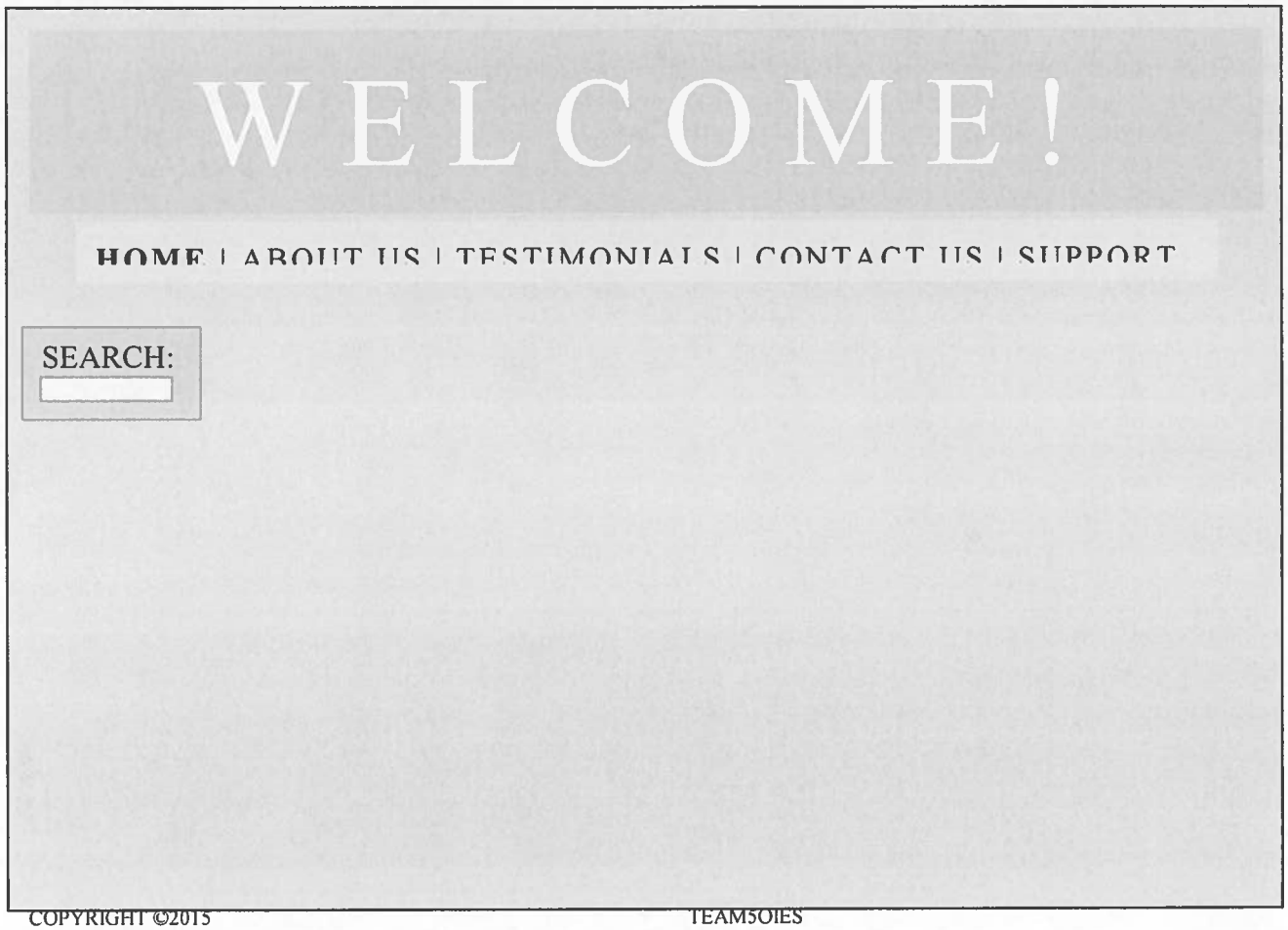
By frequently checking and testing our code we are less likely to fall behind schedule, develop a better understanding our understanding, and reduce the potential for catastrophic faults

8. Chapter VIII: Developing *TEAM5OIES* Page Master and Home Page ("Requirements Workflow" - textbook)

8.1 Problem Statement

A common misconception when in the Requirements Workflow is that the developers of the software must determine what the client wants but not particularly needs for their company. However along with that the client does not exactly know or understand what they need to help their company grow. Developers must be able to recognize the way that the current software product collects and store data and be able to improve upon that to present the client with a new, better product. This is not as straightforward as it sounds. Developing a new improved software product is already difficult for developers and software engineers to visualize and come up with a full functioning product, it is even more difficult to get the client to understand exactly what they want their product to look like when finished to meet their needs.

8.2 *TEAM5OIES* PAGE MASTER AND HOME PAGE



TEAM5OIES

HOME | **ABOUT US** | TESTIMONIALS | CONTACT US | SUPPORT

SEARCH:

WHO WE ARE:

.....

WHAT WE DO:

.....

TEAM5OIES

TEAM5OIES

HOME | ABOUT US | **TESTIMONIALS** | CONTACT US | SUPPORT

SEARCH:

JOHN says:
.....

SALLY says:
.....

TYLER says:
.....

TEAM5OIES

HOME | ABOUT US | TESTIMONIALS | **CONTACT US** | SUPPORT

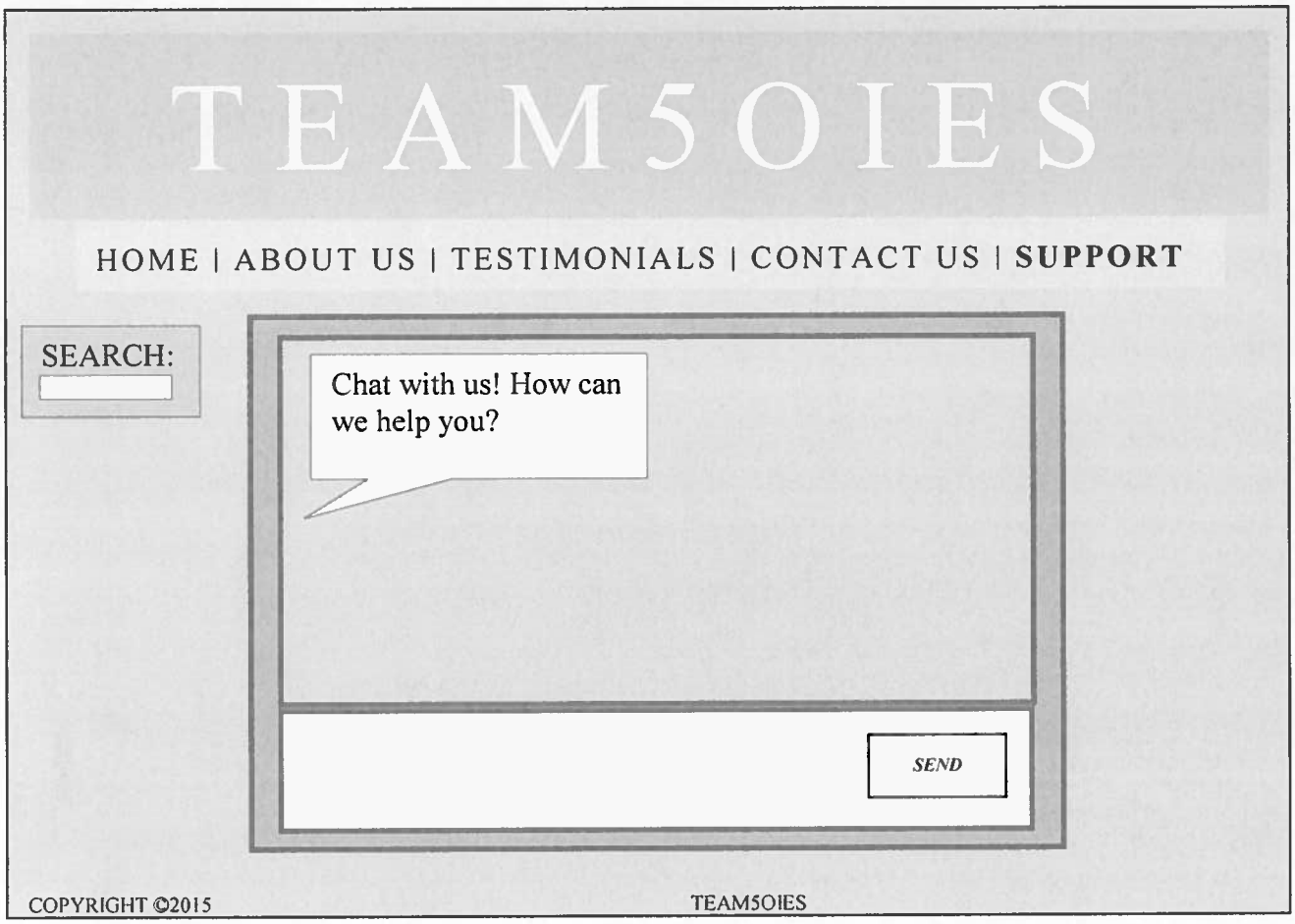
SEARCH:

We are located at:

Directions

Our phone number is ###.###.#### ext. ####

Or contact us via email at ****@cs.uh.edu



8.3 Concluding Remarks

The biggest problem we may encounter may be when the client decides to add something else to their needs and changing our requirements. As long as our design for the TEAM5OIES is flexible the changes should not be too hard to add and implement.

Chapter IX: OO Analysis Models ("OO Analysis Workflow" - textbook)

(Develop *TEAM5OIES* UML OO Analysis Models)
A. Problem Statement
B. The UML Diagrams
C. Concluding Remarks

Chapter X: OO Design Models ("OO Design Workflow" - textbook)

(Develop *TEAM5OIES* UML OO Detailed Design models)
A. Problem Statement
B. The UML Diagrams
C. Concluding Remarks

Chapter XI: Deployment Diagram ("More on UML" - textbook)

(Develop *TEAM5OIES* UML Package and Deployment models)
A. Problem Statement
B. The UML Diagrams
C. Concluding Remarks

DOCUMENT CONTROL

CHANGE HISTORY

Table 1: TLs entries (assigned work and due dates) before releasing to the team (all SQAs)

Revision	Name	**Due Date**	**Description**
1.A	TM Michelle George	02/09/2015	Complete Chapter 1 and 2
1.B	TM Joe Lu	02/09/2015	Complete Chapter 1 and 2
1.C	DBA John Loveall	02/09/2015	Complete Chapter 3
1.X	Shah Zaib	02/09/2015	Review Document
1.Y	SQA Name	02/09/2015	Review Document

Table 2: Entries when work completed (SVN Commit Comment matches Description)

Revision	Name	**Completed Date**	**Description**
1.A	TM Michelle George	02/05/2015	I Completed Chapter 2
1.B	TM & Joe Lu	02/06/2015	I Completed A and B chapter 1
1.C	DBA John Loveall	02/07/2015	I Completed Chapter 3
1.X	Shah Zaib	02/09/2015	I reviewed Document
1.Y	SQA Edison Guevara	02/09/2015	I reviewed Document

Table 3: TL entry for RED DELIVERABLES (SVN Commit Comment matches Description)

Revision	Name	**Due Date**	Description
2.0	TL Obinna Ugwuzor	02/11/2015	I changed Version to 2.0

DOCUMENT STORAGE

This file is stored in SVN at https://svn.cs.uh.edu/svn/cosc4351/team5/TEAM PROJECT DELIVERABLES/**Team Project Report.doc**.

DOCUMENT CONTROL

CHANGE HISTORY

Table 1: TLs entries (assigned work and due dates) before releasing to the team (all SQAs)

Revision	Name	Due Date	Description
1.A	TM Johnathan Hornik	03/03/2015	Complete Chapter 4 and 5
1.B	TM Hector Reyna	03/03/2015	Complete Chapter 6 and 7
1.C	DBA Jessica Balanag	03/03/2015	Complete Chapter 8
1.X	SQA Shah Zaib	03/04/2015	Review Document
1.Y	SQA Edison Guevara	03/04/2015	Review Document

Table 2: Entries when work completed (SVN Commit Comment matches Description)

Revision	Name	Completed Date	Description
1.A	TM Johnathan Hornik	03/04/2015	I Completed Chapter 4 and 5
1.B	TM Hector Reyna	03/03/2015	I Completed Chapter 6 and 7
1.C	DBA Jessica Balanag	02/26/2015	I Completd Chapter 8
1.D	TM Hector Reyna	03/09/2015	Entered the Table of Contents/Updated Each Section with headings/subheadings
1.X	SQA Shah Zaib	03/04/2015	I reviewed Document
1.Y	SQA Edison Guevara	03/04/2015	I reviewed Document

Table 3: TL entry for PURPLE DELIVERABLES (SVN Commit Comment matches Description)

Revision	Name	Due Date	Description
2.0	TL Obinna Ugwuzor	03/09/2015	I changed Version to 3.0

DOCUMENT STORAGE

This file is stored in SVN at https://svn.cs.uh.edu/svn/cosc4351/team5/TEAM PROJECT DELIVERABLES/**Team Project Report.doc**.

TEAM5OIES

Web Site Design

Version 2.0

Signature Block

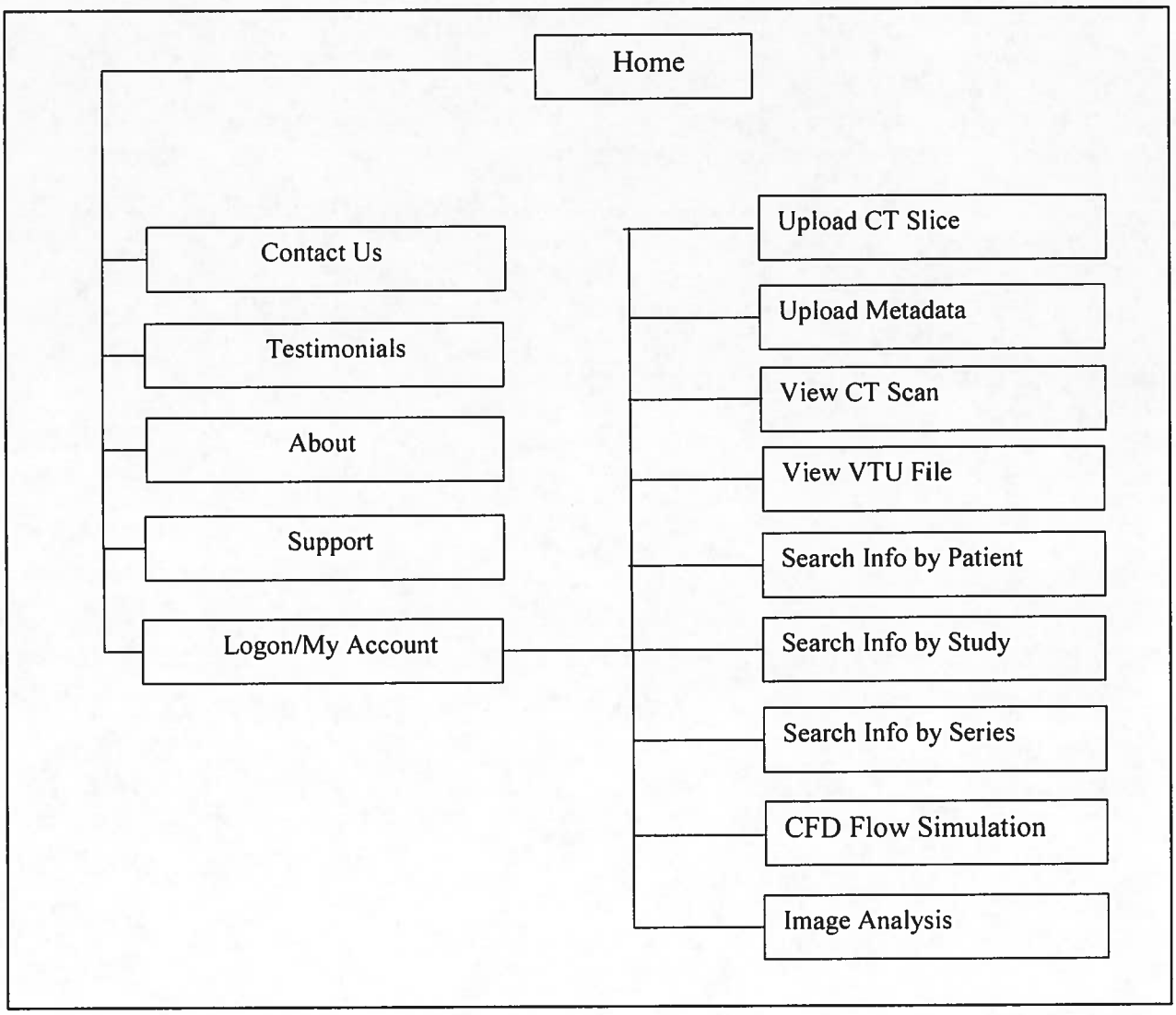
COSC 4351	Name	Signature	Date
SE Team Leader	OBINNA UGWUBOR		3-9-15
SQA	EDISON GUEVARA		3-9-15
SQA	Shah Zaib		3-9-15

Content Outline for Master Page

Contact Us – Will be a link to contact members of the EVAR system, to be determined by the client.
Support – Anyone who visits the site will be able to chat with a representative of the system.
About – Page will contain general information about the system and what it can do, for visitors who are not already registered on the website and interested in being part of it.
Testimonials – Current users can leave messages so that new users will know how they feel about the product. Visitors can search them.
Logon/My Account- This is where an established user will go to log in to the website. New users may also go here to register for the website.
The following will be viewed according to the access rights they have:

Upload CT Slice – This page allows the user to upload an EVAR CT slice into the database.
Upload Metadata – This page allows the user to upload metadata so that it will enter the database.
View CT Scan – This will allow the user to view slices of the CT scan. They will be able to use tools to either zoom in or zoom out on the image.
View VTU File – This page will allow the user to view the pressure, velocity, and wall shear stress files produced by the CFD Flow Simulation.
Search Info by Patient - User will be able to search for information on all of the patients they have access to, by patient. This will show various fields that the user would like to see, and more information to be given by the client on what they would like to see. Will also contain an option to produce an excel report from the data searched for.
Search Info by Study - User will be able to search for information on all of the patients they have access to, by study. This will show various fields that the user would like to see, more information to be given by client on what they would like to see. Will also contain an option to produce an excel report from the data searched for.
Search Info by Series - User will be able to search for information on all of the patients they have access to, by series. This will show various fields that the user would like to see, more information to be given by client on what they would like to see. Will also contain an option to produce an excel report from the data searched for.
CFD Flow Simulation – This page will allow the user to select information from the database to use in a CFD flow simulation. The results will be stored in the database. This will be .vtu files for pressure, velocity, and wall shear stress.
Image Analysis – This page will allow for the analysis of EVAR CT data. There will be automatic quantitative image analysis on the 3D reconstruction (using proprietary software) which will extract seven measurements. These will be stored in the database.

Site Diagram



Highest Priority

Support: Anyone who visits the site will be able to chat with a representative of the system.

Logon/My Account- This is where an established user will go to log in to the website. The users view will vary according to their access rights. New users may also register.

Image Analysis – This page will allow for the analysis of EVAR CT data. There will be automatic quantitative image analysis on the 3D reconstruction (using proprietary software) which will extract seven measurements. These will be stored in the database.

CFD Flow Simulation – This page will allow the user to select information from the database to use in a CFD flow simulation. The results will be stored in the database. This will be .vtu files for pressure, velocity, and wall shear stress.

Search Info by Series - User will be able to search for information on all of the patients they have access to, by series. Will show various fields that the user would like to see. More information to be given by client on what they would like to see. Will also contain an option to produce an excel report from the data searched for.

Search Info by Study - User will be able to search for information on all of the patients they have access to, by study. Will show various fields that the user would like to see.

Search Info by Patient - User will be able to search for information on all of the patients they have access to, by patient. Will show various fields that the user would like to see.

View VTU File – This page will allow the user to view the pressure, velocity, and wall shear stress files produced by the CFD Flow Simulation.

View CT Scan – This will allow the user to view slices of the CT scan. They will be able to use tools to either zoom in or zoom out on the image.

Upload Metadata – This page allows the user to upload metadata so that it will enter the

Upload CT Slice – This page allows the user to upload an EVAR CT slice into the database.

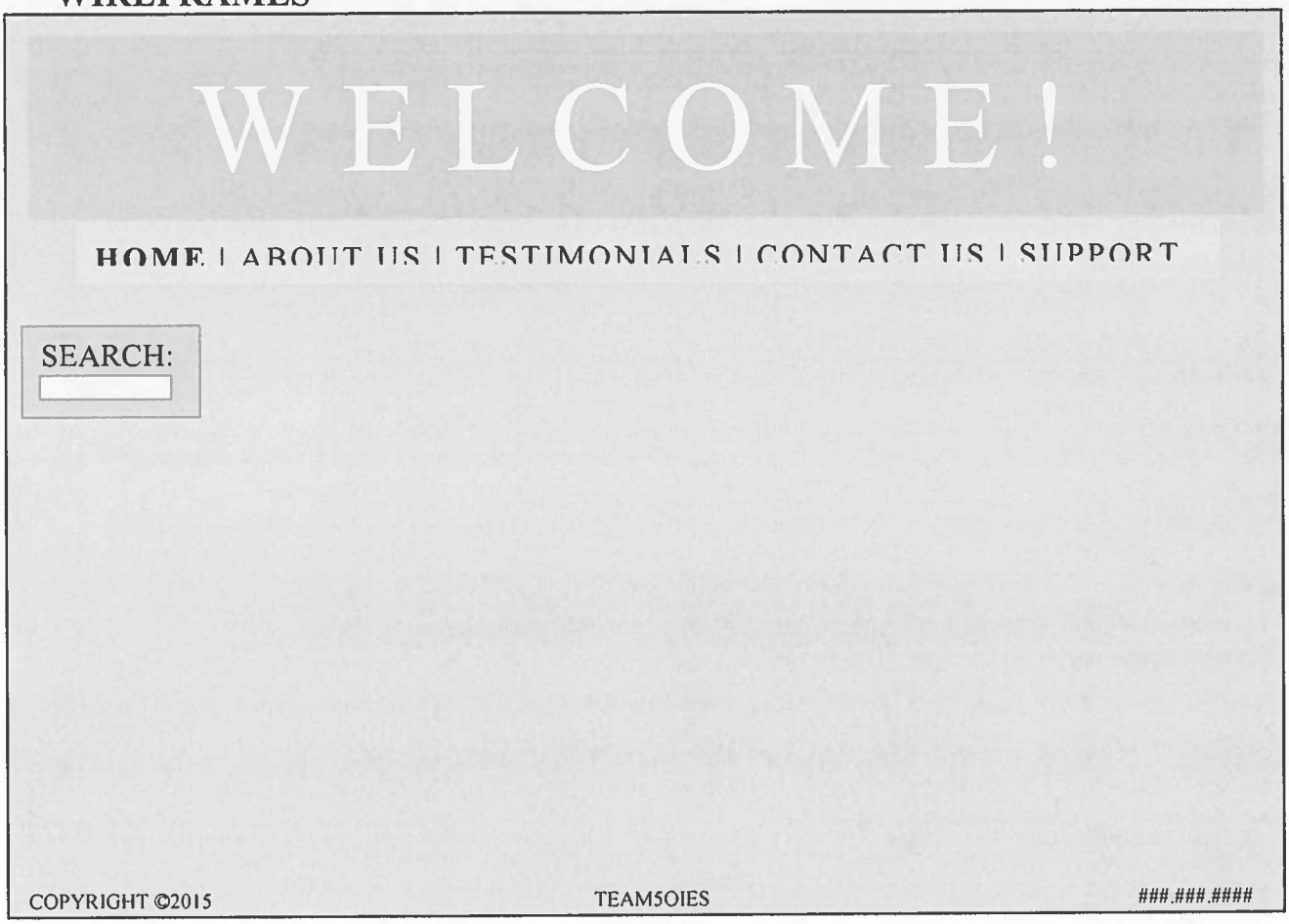
Lowest Priority

About – Page will contain general information about the system and what it can do, for visitors who are not already registered on the website and interested in being part of it.

Contact Us – Will be a link to contact members of the EVAR system, to be determined by the client.

Testimonials – Current users can leave messages so that new users will know how they feel about the product. Visitors can search them.

WIREFRAMES


WELCOME!
HOME | ABOUT US | TESTIMONIALS | CONTACT US | SUPPORT
SEARCH:
COPYRIGHT ©2015
TEAM5OIES
###.###.####

TEAM5OIES

HOME | **ABOUT US** | TESTIMONIALS | CONTACT US | SUPPORT

SEARCH:

WHO WE ARE:
.....

WHAT WE DO:
.....

TEAM5OIES

HOME | ABOUT US | **TESTIMONIALS** | CONTACT US | SUPPORT

SEARCH:

JOHN says:
.....

SALLY says:
.....

TYLER says:
.....

TEAM5OIES

HOME | ABOUT US | TESTIMONIALS | **CONTACT US** | SUPPORT

SEARCH:

We are located at:

Directions

Our phone number is ###.###.#### ext. ####

Or contact us via email at ****@cs.uh.edu


TEAM5OIES
HOME | ABOUT US | TESTIMONIALS | CONTACT US | SUPPORT
SEARCH:
Chat with us! How can we help you?
SEND
COPYRIGHT ©2015
TEAM5OIES
###.###.####

WEBSITE SCREENSHOT

DOCUMENT CONTROL

CHANGE HISTORY

Table 1: TLs entries (assigned work and due dates) before releasing to the team (all SQAs)

Revision	Name	Due Date	Description
1.A	TM Michelle George	03/03/2015	Complete the Content Outline, Site Diagram
1.B	TM Hector Reyna	03/03/2015	Complete Page Description Diagrams
1.C	DBA Jessica Balanag	03/03/2015	Complete Wireframes
1.X	SQA Hector Guevara	03/04/2015	Review Document
1.Y	SQA Shah Zaib	03/04/2015	Review Document

Table 2: Entries when work completed (SVN Commit Comment matches Description)

Revision	Name	**Completed Date**	Description
1.A	TM Michelle George	02/28/2015	I completed the Content Outline, Site Diagram
1.B	TM Hector Reyna	03/02/2015	I completed Page Description Diagrams
1.C	DBA Jessica Balanag	03/03/2015	I completed Wireframes
1.D	TM Michelle George	03/07/2015	I updated Content Outline, Site Diagram, Description Diagrams
1.E	TM Michelle George	03/08/2015	Added website screenshot
1.X	SQA Edison Guevara	03/04/2015	I reviewed Document
1.Y	SQA Shah Zaib	03/04/2015	I reviewed Document

Table 3: TL entry for RED DELIVERABLES (SVN Commit Comment matches Description)

Revision	Name	Due Date	Description
2.0	TL Obinna Ugwuzor	03/09/2015	I changed Version to 2.0

DOCUMENT STORAGE

This file is stored in SVN at https://svn.cs.u.edu/svn/cosc4351/team5/TEAM PROJECT DELIVERABLES/**Web Site Design.doc**.

80

Input Forms and Output Reports

Did not use SE Purple Requirements

Please iterate for BLUE!

TM finished on 02/26 ??

Version 3.0

Signature Block

COSC 4351	Name	Signature	Date
SE Team Leader	Obinna Ugwuzor		
SE SQA	Edison Guevara		
SE SQA	Shah Zaib		
SE Team Leader			
SE SQA			
SE SQA			
SE Team Leader			
SE SQA			
SE SQA			

TEAM5OIES

SE Team Project with Line Numbers
TEXTUAL ANALYSIS for Requirements Workflow
UML USE CASE DIAGRAM

Version 3.0

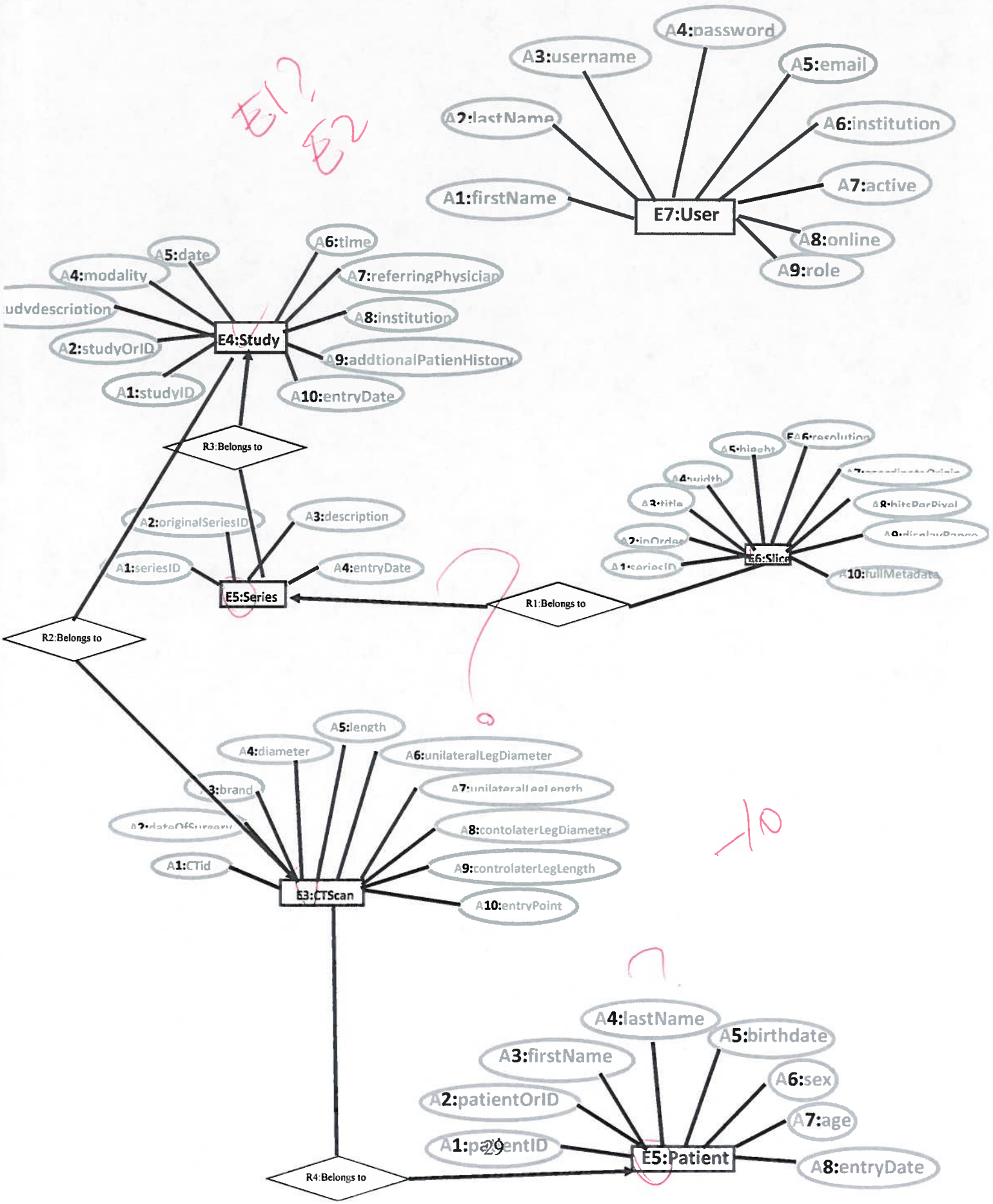
TEAM5OIES

DB Team Project with Line Numbers for ERD Modeling

Version 2.0

Approvals Signature Block

	Name	Signature	Date
DBA	Kenny Loveall	Kenny Loveall	9 February 2015
DBA	Jessica Balanag		02/11/15
TL	Obinna Ugwuzor		?
SQA	Shah Zaib		?
SQA	Edison Guevara	Edison Guevara	11 Feb 2015


A4:password
A3:username
A5:email
A2:lastName
A6:institution
A7:active
A1:firstName
E7:User
A8:online
A9:role
A5:date
A6:time
A4:modality
A7:referringPhysician
A8:institution
E4:Study
A2:studyOrID
A9:addtionalPatienHistory
A1:studyID
A10:entryDate
R3:Belongs to
A2:originalSeriesID
A3:description
A1:seriesID
A4:entryDate
E5:Series
R1:Belongs to
R2:Belongs to
A5:length
A4:diameter
A6:unilateralLegDiameter
A8:contolaterLegDiameter
A1:CTid
A9:controlaterLegLength
E3:CTScan
A10:entryPoint
A4:lastName
A5:birthdate
A3:firstName
A6:sex
A2:patientOrID
A7:age
E5:Patient
A8:entryDate
R4:Belongs to

Document Control

CHANGE HISTORY

Table 1: TLs entries (assigned work and due dates) before releasing to the team (all DBAs & SQAs)

Revision	Name	Due Date	Description
1.A	DBAs Jessica Balanag	02/09/2015	Complete ERD What Modeling
1.B	DBAs John Loveall	02/09/2015	Complete ERD What Modeling
1.X	SQA Edison Guevara	02/09/2015	Review Document
1.Y	SQA Shah Zaib	02/09/2015	Review Document

Table 2: DBAs & SQAs entries when they completed their work

Revision	Name	**Completed Date**	Description
1.A	Jessica Balanag	02/03/2015	I completed XXX **I certify that the TEAM has used "COMPILABLE" ERD LANGUAGE where EACH E, R, and A has a NUMBER and LABEL and they are marked on THIS DOCUMENT.**
1.B	John Loveall	02/03/2015	I completed YYY **I certify that the TEAM has used "COMPILABLE" ERD LANGUAGE where EACH E, R, and A has a NUMBER and LABEL and they are marked on THIS DOCUMENT.**
1.X	SQA Edison Guevara	02/08/2015	I reviewed Document
1.Y	SQA Shah zaib	02/09/2015	I reviewed Document

355

Table 3: TLs entry for RED DELIVERABLES (SVN Commit Comment matches Description)

Revision	Name	Due Date	Description
2.0	TL Obinna Ugwuzor	02/11/2015	I changed Version to 2.0

DOCUMENT STORAGE

This file is stored in **SVN** at https://svn.cs.uh.edu/svn/cosc4351/team5/**DB TEAM PROJECT DELIVERABLES /DB Team Project with Line Numbers for ERD Modeling.doc.**