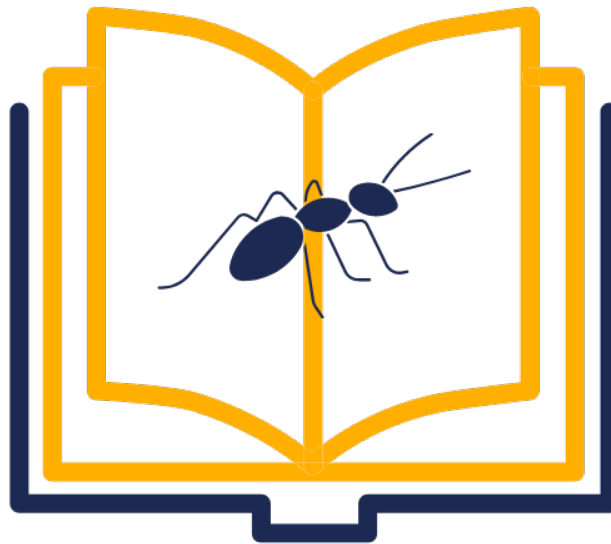


**DEPARTMENT OF COMPUTER SCIENCE  
NORTH CAROLINA A&T STATE UNIVERSITY**

**DETAILED DESIGN SPECIFICATION**

**COMP 496: SENIOR DESIGN II**

**FALL 2020**



**GROUP 3  
ANNOTEXT**

**TIARA BELL  
MYLES NELOMS  
PATRICK CRUMP, JR.**

## REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	10.29.2020	TB	document creation
0.2	11.04.2020	TB, MN, PC	complete draft
0.2	11.05.2020	MN	turned in assignment

## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>System Overview</b>	<b>5</b>
2.1	Description . . . . .	5
<b>3</b>	<b>Diagram</b>	<b>6</b>
<b>4</b>	<b>X Layer Subsystems</b>	<b>7</b>
4.1	Layer Hardware . . . . .	7
4.2	Layer Operating System . . . . .	7
4.3	Layer Software Dependencies . . . . .	7
4.4	Server / Router . . . . .	7
<b>5</b>	<b>Y Layer Subsystems</b>	<b>8</b>
5.1	Layer Hardware . . . . .	8
5.2	Layer Operating System . . . . .	8
5.3	Layer Software Dependencies . . . . .	8
5.4	View and Annotext Web App . . . . .	8
<b>6</b>	<b>Z Layer Subsystems</b>	<b>10</b>
6.1	Layer Hardware . . . . .	10
6.2	Layer Operating System . . . . .	10
6.3	Layer Software Dependencies . . . . .	10
6.4	Controller . . . . .	10
6.5	Model . . . . .	11
6.6	Database . . . . .	11
<b>7</b>	<b>Appendix A</b>	<b>12</b>

## LIST OF FIGURES

1	X Layer Subsystems . . . . .	8
2	Y Layer Subsytems . . . . .	9
3	Z Layer Subsystems . . . . .	10

## LIST OF TABLES

# 1 INTRODUCTION

Annotext is a web based application that allows users to access PDF files through our system and annotate them. Annotations capabilities will consist of highlights, footnotes, and summaries. Annotext uses a multi-threaded server connection process to allow for many users to constantly connect to the website and use the features that we offer. Each group and/or individual will be given a specific port on our domain to work from while they are annotating the PDF files they choose. When a user connects they will be given a View to work from. In this View they will be able to annotate the document and see other annotations from other users. Our product has four major inputs: account holder individual annotations, up votes for specific annotations from account holders and down votes for specific annotations from account holders. Our product will have two major outputs: textbook files and collective annotations per PDF sorted by relevance.

Our end user will be able to log into our website and scroll through digital textbooks separated by topic, after clicking on a PDF an account holder will have the option to enter an annotation for the page they are viewing. Annotations are logged with account username, section referenced (paragraph and page number), file referenced, and vote value. When viewing a page the user can scroll through previous annotations through a side tab. These annotations will be ordered by relevance based on other users using the up and down voting mechanic.

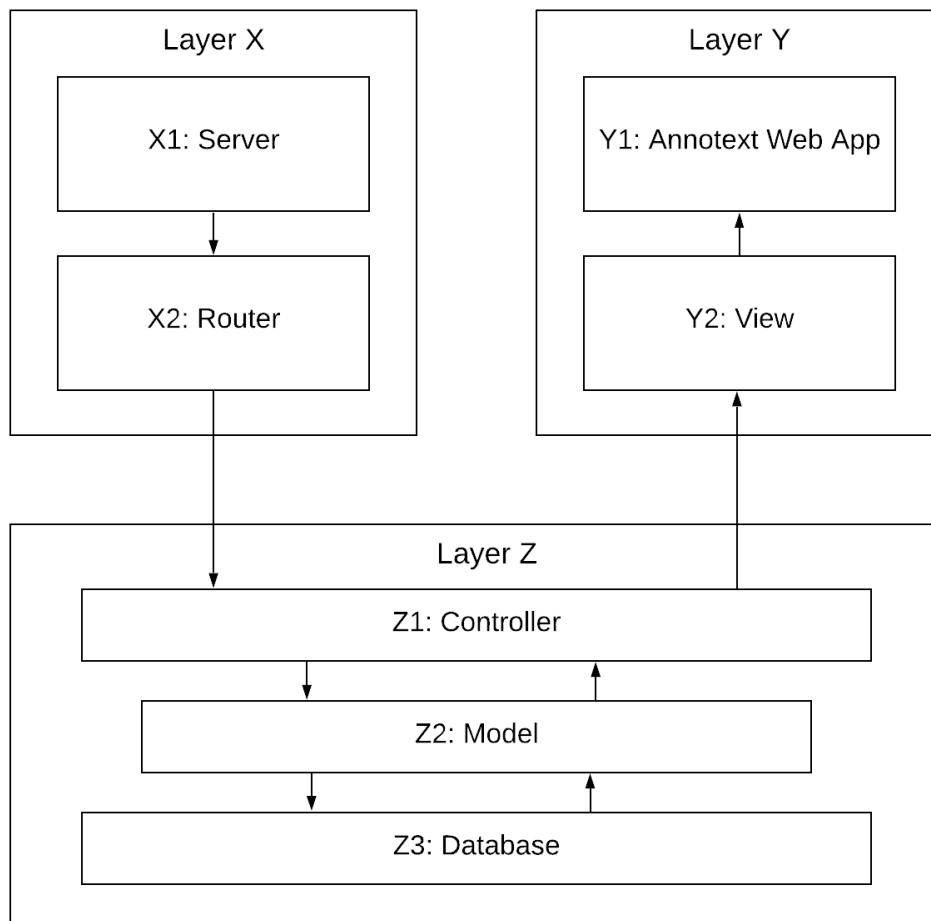
Our core requirements include: account creation, making and posting annotations, viewing PDFs, toggling on and off annotations from view, and sorting relevant annotations by rating.

## 2 SYSTEM OVERVIEW

### 2.1 DESCRIPTION

Our Server receives http requests from the Annotext Web Application ran by our client. Clients can navigate throughout web-pages indexed on our homepage. If on the library page the client will have the opportunity to submit an annotation using a GUI like text form using an ASP.NET partial view. The information submitted in this form will be handled by a controller to be encapsulated into an object called an annotation model and input into the database using LINQ query language and Entity Framework to access "code first" tables in our database. The next path of data flow available in our system will be the display of PDF documents on the library page. These are stored in our local server memory and will be accessed through a basic HTML PDF viewer and will be displayed as a partial view above our annotation text form. Lastly we will display all annotations that refer to a PDF that a user is viewing. This will require basic Entity Framework to implement. The annotations will be pulled from our database into a partial view using LINQ query language. With Entity Framework we can update this catalog in real time.

### 3 DIAGRAM



## **4 X LAYER SUBSYSTEMS**

This layer is about setting up a connection with the server and sending the request to the appropriate party. Once the HTTP request is sent from the client's system, a connection is made with our server to accept the request. The request is then transported to its intended location via the Router. The information required to send the data request to its destination is embedded in the HTTP request.

### **4.1 LAYER HARDWARE**

A physical server running at annotext.ncat.edu. User must access NCAT's VPN in order to access server.

### **4.2 LAYER OPERATING SYSTEM**

No specific OS required.

### **4.3 LAYER SOFTWARE DEPENDENCIES**

Server depends on SQL Server to respond to queries from HTML requests and Visual Studios and MVC using Entity Framework in order to service requests from the Web app and route them to the correct controller.

### **4.4 SERVER / ROUTER**

The server is the key hardware component used to host our application. Once the client makes an HTTP request, it is sent from the client's system and a connection is made with our server to accept the request. The request is then transported via the Router, a software component on the server. The HTTP request tells the Router not only where the data needs to go (using the IP address) but who specifically to send the information to on the server (using the port number). Once contact is made with the Server the Router routes the information to not only the appropriate Controller, but the appropriate Action within that Controller.

#### **4.4.1 SUBSYSTEM HARDWARE**

Our only hardware is a web server housed at NCAT.

#### **4.4.2 SUBSYSTEM OPERATING SYSTEM**

No required OS, but we are using Windows.

#### **4.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES**

The router's embedded software within the server is the only component needed for this sub-layer.

#### **4.4.4 SUBSYSTEM PROGRAMMING LANGUAGES**

A description of any programming languages used by the subsystem.

#### **4.4.5 SUBSYSTEM DATA STRUCTURES**

None.

#### **4.4.6 SUBSYSTEM DATA PROCESSING**

Data is processed through a series of internet protocols. TCP, UDP and the IP Protocol stack are used to route the information from the client to our server and back again.

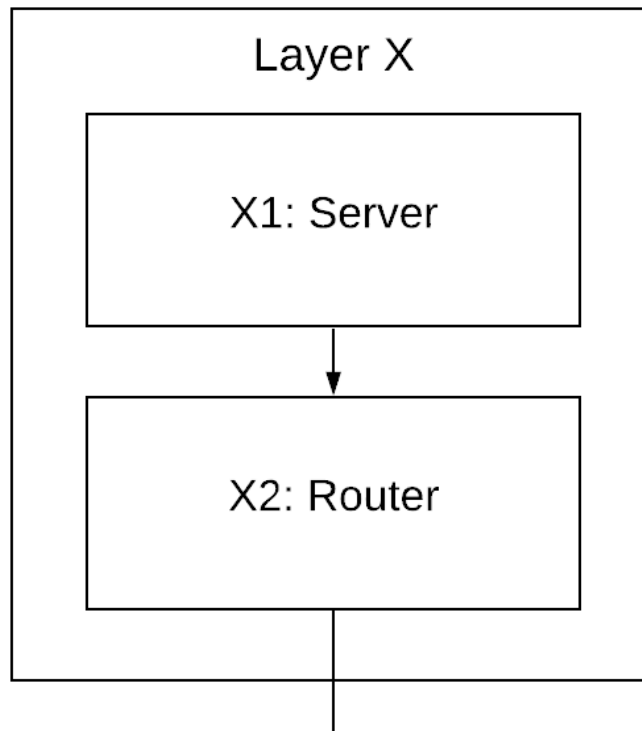


Figure 1: X Layer Subsystems

## 5 Y LAYER SUBSYSTEMS

This layer is about viewing the data the client requested. Once the request has been processed and returned, the View class then formats the information in a format conducive for user interaction and use. The View is displayed on our web application, Annotext and the data cycle will begin again when the user makes another request.

### 5.1 LAYER HARDWARE

This Layer is stored on our physical server on the campus of NCAT.

### 5.2 LAYER OPERATING SYSTEM

No specific OS required, but we are using Windows.

### 5.3 LAYER SOFTWARE DEPENDENCIES

This layer requires the frameworks created by ASPNET.

### 5.4 VIEW AND ANNOTEXT WEB APP

The View, a class within the MVC model, is used to display content on our web application, Annotext, and the data cycle will begin again when the user makes another request. The Annotext Web App sublayer is about viewing the data the client requested. Annotext is a web application and once the request has been processed and returned from the View class, it then formats the information in a format conducive for user interaction and use.



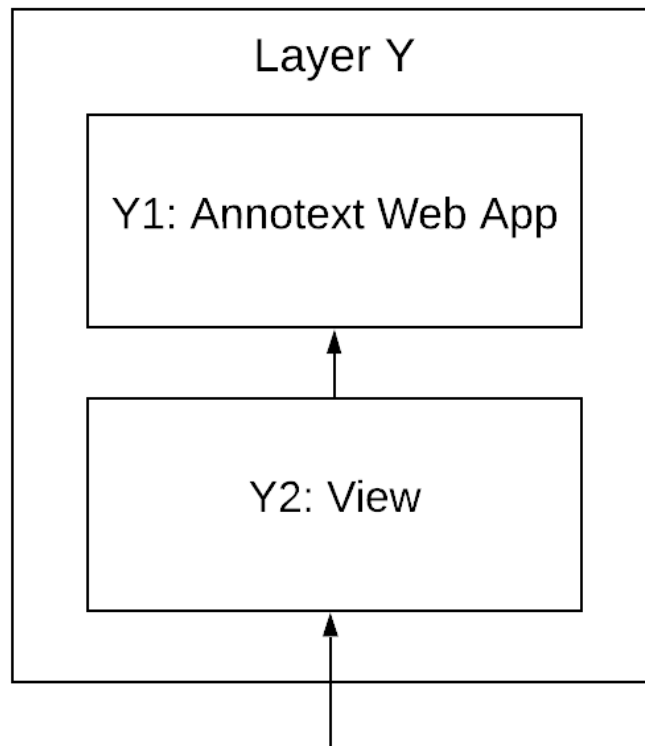


Figure 2: Y Layer Subsystems

#### **5.4.1 SUBSYSTEM OPERATING SYSTEM**

Requires a Windows OS.

#### **5.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES**

Visual Studio Community, and ASP.NET MVC CORE 5 using Entity Framework.

#### **5.4.3 SUBSYSTEM PROGRAMMING LANGUAGES**

These processes are coded with html5 with a C wrapper.

## 6 Z LAYER SUBSYSTEMS

This layer is about retrieving the data and transferring it back to the client in a user friendly format. The controller handles the HTTP request of the user by using methods called "ActionResults" in order to save the edited models to the database. These methods return the views that the user sees on their local browser. The specific Controller chosen based upon the HTTP request. Once the Controller is chosen it then utilizes the information passed to it by the view, including appropriate method, to make any changes to the database or even redirect a user to another page.

### 6.1 LAYER HARDWARE

This Layer is stored on our physical server on the campus of NCAT.

### 6.2 LAYER OPERATING SYSTEM

The server is running a windows OS.

### 6.3 LAYER SOFTWARE DEPENDENCIES

This layer depends on SQL Server, and Entity Framework for entries and queries to the database. The layer also depends on Visual Studios and ASPNET MVC CORE 5 in order to create model objects and controller actions/methods.

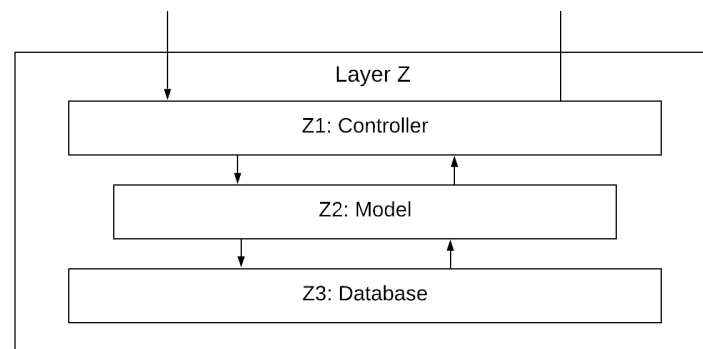


Figure 3: Z Layer Subsystems

### 6.4 CONTROLLER

The controller handles the http requests delivering the data to the correct model to be saved as an object in our database.

#### 6.4.1 SUBSYSTEM HARDWARE

Stored in code on our physical server.

#### 6.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES

ASPNET library, and Visual Studios Community.

#### 6.4.3 SUBSYSTEM PROGRAMMING LANGUAGES

This subsystem will be coded in CSharp.

#### 6.4.4 SUBSYSTEM DATA STRUCTURES

- The PDF controller consist of a public string to hold the selected PDF, and an Action method to redirect to the Annotation creation page. This string will be passed to the annotationsController using a Dynamic Variable Array called "TempData"

- The Annotation controller will house several action methods for PDF creation, voting mechanics, Annotation listings. Most importantly this controller will set a Dynamic Variable Dictionary called the ViewBag using this dictionary we can pass information from the controller to the view in order to display PDFs and filter the appropriate annotations for display.
- 

#### **6.4.5 SUBSYSTEM DATA PROCESSING**

We will be using Entity Framework to create and manage our database using their "code first" method and LINQ queries.

### **6.5 MODEL**

#### **6.5.1 SUBSYSTEM HARDWARE**

Stored in code on our physical server.

#### **6.5.2 SUBSYSTEM SOFTWARE DEPENDENCIES**

ASP.NET library(MVC5), and Visual Studios Community

#### **6.5.3 SUBSYSTEM PROGRAMMING LANGUAGES**

This subsystem will be coded in C.

#### **6.5.4 SUBSYSTEM DATA STRUCTURES**

A model is an object that takes in parameters and allows us to create several instances of any given mold. We have two Models, the Annotation model and the PDF model.

- Annotation model holds getters and setters for filename, author, voteValue, PageNumber, and Paragraph. it also houses a foreign key connecting each to a PDF with a one to many relationship.
- PDF model only holds getters and setters for the id and the filename(path).

### **6.6 DATABASE**

#### **6.6.1 SUBSYSTEM HARDWARE**

Stored in code on our physical server.

#### **6.6.2 SUBSYSTEM SOFTWARE DEPENDENCIES**

ASP.NET library(MVC5), Visual Studios Community, and Entity Framework.

#### **6.6.3 SUBSYSTEM PROGRAMMING LANGUAGES**

This subsystem will be coded partially through SQL Query language in SQL server, and partially through Entity Framework using C and LINQ Query language

#### **6.6.4 SUBSYSTEM DATA STRUCTURES**

The Database will include several tables holding annotation models,PDF file paths, and account objects.

## **7 APPENDIX A**

Include any additional documents as an appendix as necessary.

## REFERENCES