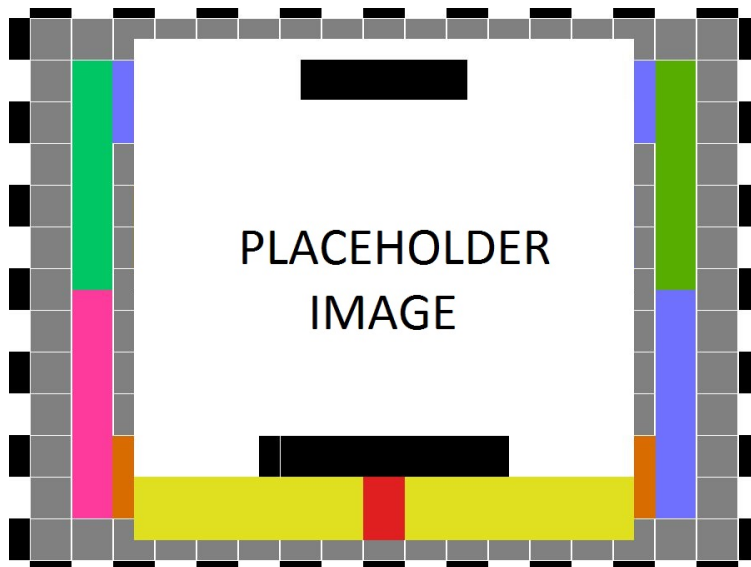


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
FALL 2019**



**TEAM MINTS
UTA ADVISING**

**ISHOR RIJAL
MAHESHWOR RAUT
TUFAN ACHARYA
NAWARAJ BHURTEL
SAMEER CHAULAGAIN**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	9.12.2019	IR	document creation
0.2	9.20.2019	MR, NB, SC, TA	complete draft
0.3	9.24.2019	SC, TA, IR, NB	release candidate 1

CONTENTS

- 1 Introduction 5**
- 2 System Overview 5**
- 3 COMMUNICATION LAYER SUBSYSTEMS 6**
 - 3.1 Layer Hardware 6
 - 3.2 Layer Operating System 6
 - 3.3 Layer Software Dependencies 7
 - 3.4 Login 7
 - 3.5 Authentication 8
 - 3.6 Filter 8
 - 3.7 Response 9
- 4 CONTROL LAYER SUBSYSTEMS 10**
 - 4.1 Layer Hardware 10
 - 4.2 Layer Operating System 10
 - 4.3 Layer Software Dependencies 10
 - 4.4 VALIDATION 10
 - 4.5 ROLES 11
- 5 DATA LAYER SUBSYSTEMS 12**
 - 5.1 Layer Operating System 12
 - 5.2 Layer Software Dependencies 12
 - 5.3 Storage 12
 - 5.4 Data Control 13
- 6 VIEW LAYER SUBSYSTEMS 14**
 - 6.1 Layer Hardware 14
 - 6.2 Layer Operating System 14
 - 6.3 Layer Software Dependencies 14
 - 6.4 User Interactions 14
 - 6.5 Graphical User Interface 15
- 7 Appendix A 16**

LIST OF FIGURES

1	System Overview for UTA Advising	6
2	Example subsystem description diagram	7
3	Example subsystem description diagram	10
4	Example subsystem description diagram	12
5	Example subsystem description diagram	14

LIST OF TABLES

1 INTRODUCTION

The advising applications that we are targeting as of now is going to be a web application. The application will have three end points each for the advisor login, student login and another for the admin/developer to track the issues and the changes that we are going to implement.

The first and the most important page is for the students who are targeting to schedule their advising with one of the advisors. The advisors could either faculty advisors or the staff. The students who are targeting to schedule an appointment can simply input their basic details like UTA ID, majors and the classes they have taken along with the semester for which they are looking to get advised. The students will also be able to choose particular advisors like faculty/staff if they are available and can put their name in the respective advisors wait list.

The second end point would be for the advisors. Advisors could see the list of the students waiting to get advised. Also the advisors will be able to see the UTA ID of the respective students and will also have the form already filled by the students waiting to get advised. The form includes the basic details necessary for the advisors like major information, classes taken, classes they are planning to take in the respective semester. This way they will be able to advise the students more effectively. We will have the databases that stores all the data of the students that have been advised and store their login credentials in cases they came back for advising in near future.

We will be making the advising process completely online by signing up students online and getting rid of the paper based sign-up sheet. The students will also be able to choose the advisor who advised them in the past or pick a new advisor every time they make appointment. The advisor will also have a very easy interface where they can see who is coming next and if they are new or someone who was already advised by them in the past. The advisors will also be able to re-schedule the appointments and change schedule which can be seen by the students before making appointment.

Our system will be very simple to use by any new or existing user because we will be using GUIs and figures to represent options and features. Everything happening in the back-end is secure and safe which takes care of user's privacy.

2 SYSTEM OVERVIEW

The advising system is structured as traditional dynamic web application which would be primarily be decisive based on the communication layer, control layer, data layer and view layer. These layers would function highly individually as well as interact with each other to perform the required functionality for the system. The communication layer would perform the role of bridge to connect between the view layer and control layer, whereas data layer would be helping to handle the data storage and access. Even though data layer might be functional in data access, control layer would be highly administrative in term of roles and data access process.

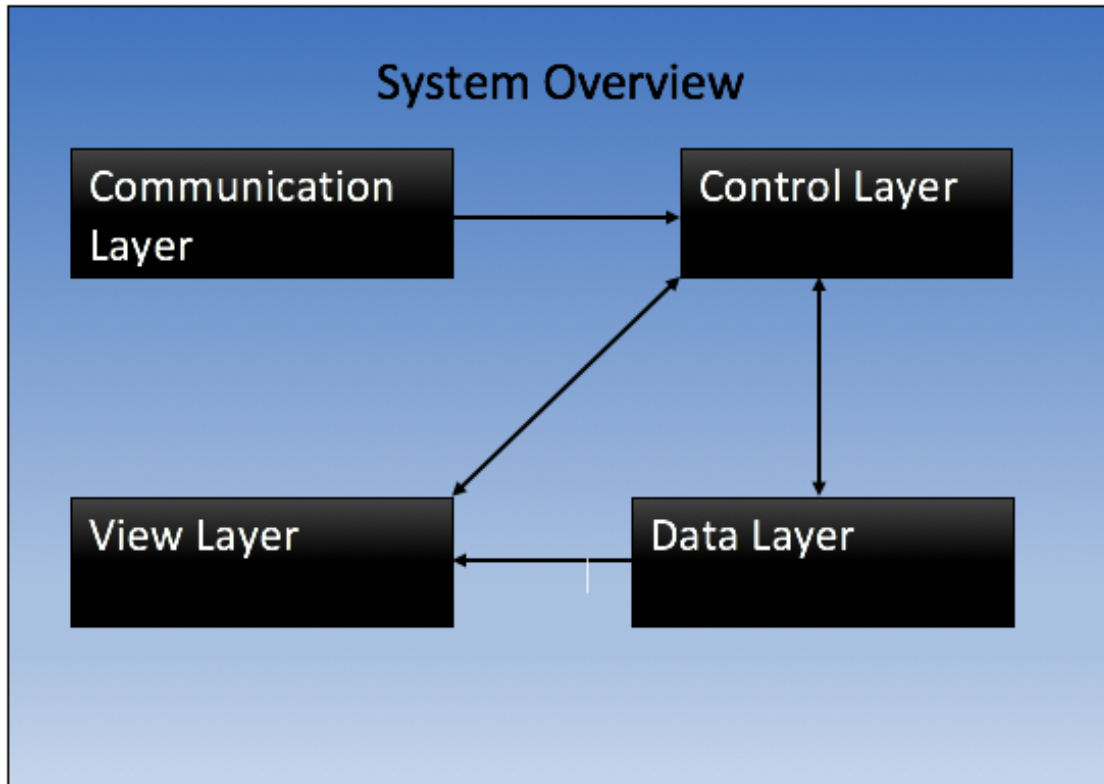


Figure 1: System Overview for UTA Advising

3 COMMUNICATION LAYER SUBSYSTEMS

In this layer to communicate and retrieve and process the request of actions, the communication layer will request the control layer to verify the request action and once the control layer verified the request with the data layer it will authenticate the request and provide the access for that particular request. The communication layer is the main component in this section since it acts as the medium for communicating between several layers in the system. To do any task the information will forward through the communication layer and pass along several subsystems present in the system. For example, if the admin tries to log in to the system that information will pass through the communication layer to the control layer and once the control layer verifies the credential with the data layer and get the green signal it will allow the person to get in the system. So the communication layer is the key component in this project.

3.1 LAYER HARDWARE

Since this project is completely based on the software platforms, we do not have any direct dependencies with any kind of hardware.

3.2 LAYER OPERATING SYSTEM

Since this web based application there will be no dependencies with operating system So any device with compatible internet connection will be sufficient to run this web application.

3.3 LAYER SOFTWARE DEPENDENCIES

We are trying to execute our project based on php framework. It will operate on the Model View Controller. We are still discussing the platform for message notification. We are using phpMyAdmin a free software tool written in PHP, to handle the administration of MySQL over the Web to store user's data.

3.4 LOGIN

This subsystem is based on php framework. Basically when user tries to login in the system the request will forward to the control layer and control layer will match the credentials in the database also known as authentication process, if the information gets matched then in the response system will grant the access to that particular actions. And if the information doesn't match in the database it will filter the request and deny the permission for log in.

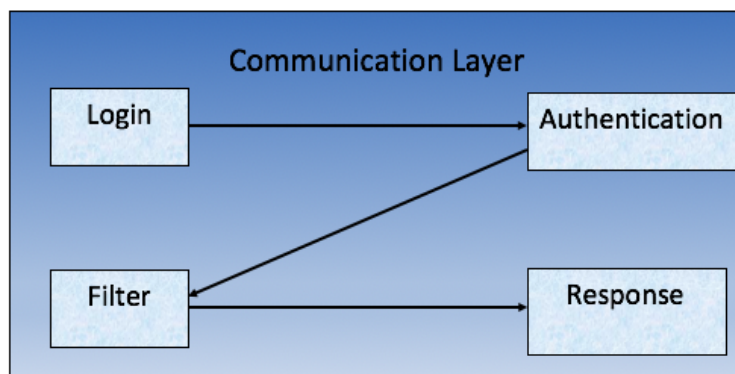


Figure 2: Example subsystem description diagram

3.4.1 SUBSYSTEM HARDWARE

Since this project is completely based on the software platforms, we do not have any direct dependencies with any kind of hardware.

3.4.2 SUBSYSTEM OPERATING SYSTEM

Since this web based application there will be no dependencies with operating system. So any device with compatible internet connection will be sufficient to run this web application.

3.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

We are trying to execute our project based on php framework. It will operate on the Model View Controller.

3.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

We are using php as a programming language for this task.

3.4.5 SUBSYSTEM DATA STRUCTURES

There is no particular data structure used for login subsystem.

3.4.6 SUBSYSTEM DATA PROCESSING

We are using phpMyAdmin a free software tool to handle the administration of MySQL over the web to handle users data.

3.5 AUTHENTICATION

This is also based on the php framework which checks if the format of the data entered is valid. We have our own standard set for authentication so this system will not interact with database for validity. The authentication will also make sure that fields and requests are correct according to the system. There are certain standard for the authentication process.

3.5.1 SUBSYSTEM HARDWARE

Since this project is completely based on the software platforms, we do not have any direct dependencies with any kind of hardware.

3.5.2 SUBSYSTEM OPERATING SYSTEM

Since this web based application there will be no dependencies with operating system So any device with compatible internet connection will be sufficient to run this web application.

3.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

We are using JQuery, a javascript library which allow us to do a lot of things without having to write javascript our self.

3.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

We are using php as a programming language for this project.

3.5.5 SUBSYSTEM DATA STRUCTURES

We are using some types of data structures like queue, list, arrays and so on. We are also using verification algorithms to verify username and password entered by the user.

3.5.6 SUBSYSTEM DATA PROCESSING

We are using phpMyAdmin a free software tool to handle the administration of MySQL over the web to handle users data.

3.6 FILTER

This subsystem mainly focuses on limiting the amount of data is being sent by sending only required data by filtering.

3.6.1 SUBSYSTEM HARDWARE

Since this project is completely based on the software platforms, we do not have any direct dependencies with any kind of hardware.

3.6.2 SUBSYSTEM OPERATING SYSTEM

Since this web based application there will be no dependencies with operating system So any device with compatible internet connection will be sufficient to run this web application.

3.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

We are using JQuery, a javascript library which allow us to do a lot of things without having to write javascript ourself. We are also using symfony as framework for the project

3.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

We are using php as a programming language for this system.

3.6.5 SUBSYSTEM DATA STRUCTURES

We are using some types of data structures like queue, list, arrays and so on.

3.6.6 SUBSYSTEM DATA PROCESSING

We are using phpMyAdmin a free software tool to handle the administration of MySQL over the web to handle users data.

3.7 RESPONSE

This subsystem deals with what to respond with. Based on the filtered input the response will be sent out. This is the end point of the Communication layer. After sending the response Communication layer interacts with control layer for validation.

3.7.1 SUBSYSTEM HARDWARE

Since this project is completely based on the software platforms, we do not have any direct dependencies with any kind of hardware.

3.7.2 SUBSYSTEM OPERATING SYSTEM

Since this web based application there will be no dependencies with operating system So any device with compatible internet connection will be sufficient to run this web application.

3.7.3 SUBSYSTEM SOFTWARE DEPENDENCIES

We are using JQuery, a javascript library which allow us to do a lot of things without having to write javascript ourself

3.7.4 SUBSYSTEM PROGRAMMING LANGUAGES

We are using php as a programming language for this system.

3.7.5 SUBSYSTEM DATA STRUCTURES

We are using some types of data structures like queue, list, arrays and so on. We are also using some sorting algorithms.

3.7.6 SUBSYSTEM DATA PROCESSING

We are using phpMyAdmin a free software tool to handle the administration of MySQL over the web to handle users data.

4 CONTROL LAYER SUBSYSTEMS

The advising application that we are working on is a web based application and does not involve any hardware equipment to operate other than the phone/computer required for the end users and the computer for the developers for designing and developing. The application that we are building is written in PHP and we have used the php MySQL server for the database connectivity. The MySQL server hosts all the data in table. The table contains the important data like login information, details of the student waiting for advising and some advisors information too for login in future. We are using XAMPP for the server part to communicate with the client. XAMPP server is a solution stack package consisting mainly of the Apache HTTP Server. After completing the project on the local machine we are planning to host our application either in College Server or in Amazon Web Services.

4.1 LAYER HARDWARE

Since this project is completely based on the software platforms, we do not have any direct dependencies with any kind of hardware.

4.2 LAYER OPERATING SYSTEM

Since the application is a web based application, it works on any Operating Systems. (Linux, MacOS, Windows etc.)

4.3 LAYER SOFTWARE DEPENDENCIES

We are using php as a programming languages and have used MySQL , XAMPP as a dependices in our web based software application.

4.4 VALIDATION

Validation is a web services that validates all the login information entered by the users. After the communication layer validates the requirements for the user name and password, the validation in the control layer validated who is entering the systems advisors or the students. Then the system use the database to check who is entering the system. After the login, the data is stored in the database and we have built this validation web-services used to retrieve the information from database and store the data to the database.

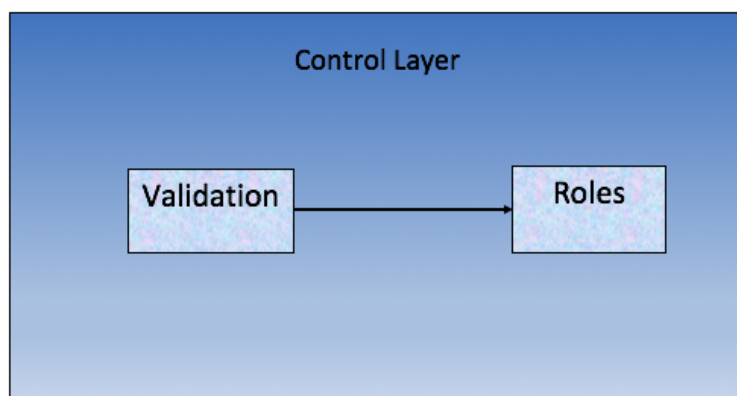


Figure 3: Example subsystem description diagram

4.4.1 SUBSYSTEM HARDWARE

This a software based application so no hardware is involved.

4.4.2 SUBSYSTEM OPERATING SYSTEM

Any Operating System from where you can use the web browsers.

4.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

We have used php as a programming language and used XAMPP as a web servers and used mysql for database and used validation for the data manipulation.

4.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

We have used PHP and Mysql for the database.

4.4.5 SUBSYSTEM DATA STRUCTURES

We have a class and we have defined HTTP methods for the data to get,post and delete the data to/from database.

4.4.6 SUBSYSTEM DATA PROCESSING

This web servies validates the data and takes the user to new User Interface.

4.5 ROLES

This section covers the roles subsystem that is in our controller system. We mostly have three roles of people, students, advisors and the administrator panel. After validating we can direct the users to their individual user page.After the validation the users can be either the students, advisors or the administrator. Students can put their email as a user name and the validator APIs verify it with the database while the advisors and administrator can have the admin access and see all the information entered by the students. After the validation and verification of the roles the individual UI pages of the users are shown and the data collection for the further process in done.

4.5.1 SUBSYSTEM HARDWARE

This a software based application so no hardware is involved.

4.5.2 SUBSYSTEM OPERATING SYSTEM

Any Operating System from where you can use the web browsers.

4.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

We have used php as a programming language and used XAMPP as a web servers and used mysql for database and used validation for the data manipulation.

4.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

We have used php as a programming languages.

4.5.5 SUBSYSTEM DATA STRUCTURES

We have a class and we have defined HTTP methods for the data to get,post and delete the data to/from database.

4.5.6 SUBSYSTEM DATA PROCESSING

After the validator APIs verify the login credentials with the database the validation and verification of the roles the individual UI pages of the users are shown and the data collection for the further process in done. Users role can be Advisors/Students.

5 DATA LAYER SUBSYSTEMS

Data Layer is one of the most important part of the entire system. It contains all the databases and the persistence layer to access the data from the databases. We have Data Control and the Storage in the data layer subsystems.

5.1 LAYER OPERATING SYSTEM

Any operating system having web browser will run the application.

5.2 LAYER SOFTWARE DEPENDENCIES

There is no software dependencies.

5.3 STORAGE

Storage is a databases that we are using. For this particular web application we are using the MYSQL databases for the storage. We have various tables, figures, encrypted password in the storage. Therefore the proper implementation and the security aspects are given priority.

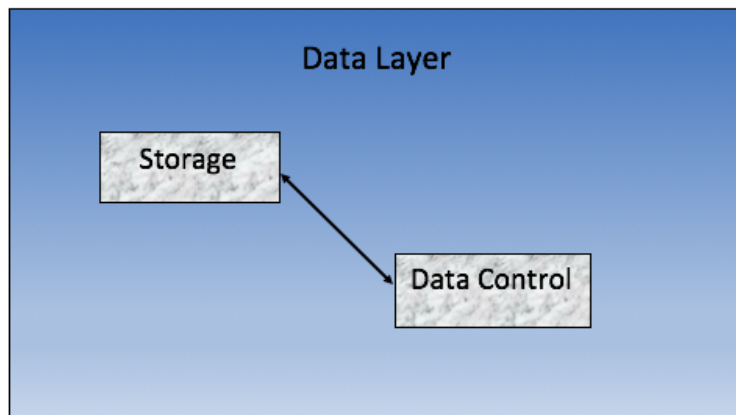


Figure 4: Example subsystem description diagram

5.3.1 SUBSYSTEM OPERATING SYSTEM

Any operating system having web browser will run the application.

5.3.2 SUBSYSTEM SOFTWARE DEPENDENCIES

We are using php as a programming languages and have used MySql , XAMPP as a dependencies in our web based software application.

5.3.3 SUBSYSTEM PROGRAMMING LANGUAGES

We have used PHP and Mysql for the database.

5.3.4 SUBSYSTEM DATA STRUCTURES

We have a class and we have defined HTTP methods for the data to get,post and delete the data to/from database.

5.3.5 SUBSYSTEM DATA PROCESSING

This web service will send and retrieve data from the database using PHP.

5.4 DATA CONTROL

Data Control is in the system where we manipulate the data. This layer works as a data persistence layer. In this layer we will have a bunch of APIs to work on the GET, POST, CREATE and DELETE the data from the databases. This layer will have a data connection API/library like JDBC or Hibernate.

5.4.1 SUBSYSTEM OPERATING SYSTEM

Any operating system having a web browser will run the application.

5.4.2 SUBSYSTEM SOFTWARE DEPENDENCIES

This layer will have a data connection API/library like JDBC or Hibernate.

5.4.3 SUBSYSTEM PROGRAMMING LANGUAGES

We have used PHP and MySQL.

5.4.4 SUBSYSTEM DATA STRUCTURES

This subsystem is responsible to access the data from our User Interface Page. Whenever the users enter anything it is stored in our database in the form of an object. The stored data is then accessed by the advisors and the algorithm to calculate the best estimated time.

5.4.5 SUBSYSTEM DATA PROCESSING

This web service will send and retrieve data from the database using PHP.

6 VIEW LAYER SUBSYSTEMS

View layer is dominantly front end of the the application. It is responsible for managing all the interaction between the user and system. Its subsystems are user interactions and graphical user interface. Even though user interaction and graphical user interface might be categorized in the same system, we have categorized it make interactions and interfaces different components.

6.1 LAYER HARDWARE

Since this project is completely based on the software platforms, we do not have any direct dependencies with any kind of hardware.

6.2 LAYER OPERATING SYSTEM

This layer does not have any dependencies on the operating system, however the growing use of Linux environment among the open source development platform makes the use of framework like React easier to operate on the Linux system than the Windows. However, no dependencies on operating system interrupts the subsystem.

6.3 LAYER SOFTWARE DEPENDENCIES

This layer is completely UI based, to make UI design more interactive we are using some supportive platform like React which is CLI based framework for organizing the various aspects of design. Apart from react, yarn is also used to assist the some of the functions in the react.

6.4 USER INTERACTIONS

This is a class level sub system which is specifically responsible for only providing the data from the user to the controllers. Though this subsystem might be placed as the part of GUI since we aim this class just to be used for interactions, it gets a separate place in the system design.

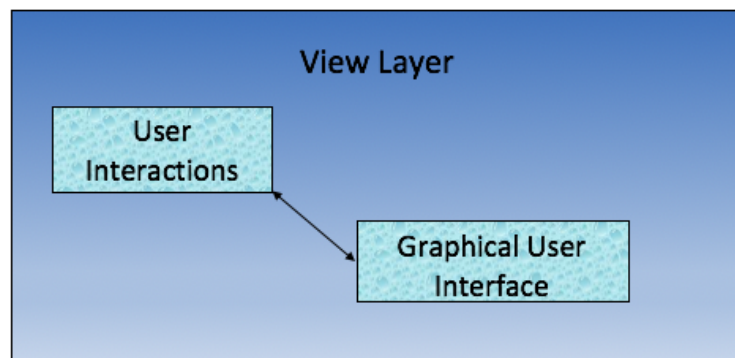


Figure 5: Example subsystem description diagram

6.4.1 SUBSYSTEM HARDWARE

Hardware is not used in this subsystem.

6.4.2 SUBSYSTEM OPERATING SYSTEM

This subsystem is made compatible for every browsers, so no operating systems dependencies includes.

6.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Apart from the frameworks used and general view layer dependencies, this specific subsystem do not have software dependencies.

6.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

php 5.3.6 is being used as programming language, not technically a programming language, however its the element that's being used. And some HTML,CSS, and JavaScript is also used.

6.4.5 SUBSYSTEM DATA STRUCTURES

Any specific standard data structures do not apply for the subsystem.

6.4.6 SUBSYSTEM DATA PROCESSING

This layer does nothing but takes data from user and submits it to the controller hence, no any specific data processing techniques are required here.

6.5 GRAPHICAL USER INTERFACE

Except for taking the input from the user, every other things that are required for the website to be dynamic on the front end is developed by this subsystem. It is responsible for displaying information, publishing information, facilitating the admin and the user of the application all other functionalities on the presentation level.

6.5.1 SUBSYSTEM HARDWARE

This subsystem is not dependent on any hardware either.

6.5.2 SUBSYSTEM OPERATING SYSTEM

Any operating system should be easily compatible with this layer as well.

6.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This layer depends on the various libraries for the designs and effective data manipulation. Some frameworks like validation for validating the user data, recaptcha to confirm the human user, carbon for date and time handling, snappy for making the pdfs of the student classes completed forms are being used.

6.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

Javascript, Php , CSS are the programming languages being used.

6.5.5 SUBSYSTEM DATA STRUCTURES

Data from the get line from user would be managed and manipulated in the controller layer using the queue.

6.5.6 SUBSYSTEM DATA PROCESSING

This layer submits data to control layer and gets control layer manipulates anything required, hence no any subsystem data processing algorithm is used up untill now.

7 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES