Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2015

Team Name
Product Name

Allison Johnsgard
Edward Fankhauser
Narayan Rimal
Victor Garcia
Neelim Haider

Revision History

Revision	Date	Author(s)	Description
0.1	10.01.2015	GH	document creation
0.2	10.05.2015	AT, GH	complete draft
0.3	10.12.2015	AT, GH	release candidate 1
1.0	10.20.2015	AT, GH, CB	official release
1.1	10.31.2015	AL	added design review requests

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

The Smart Hospital Management Tools website is meant to combine multiple pages in order to allow the Smart Hospital to manage their information in one place. The website is meant to fuflfill key requirements, such as allowing simulations to be scheduling, managing inventory logs, create events and displaoy them on a calendar, and allow the user to log in and log out.

2 System Overview

This section describes our system by presenting it in different layers, illustrating the layer that captures data on the top, and interacting with the lower-level layer on the bottom, the data processing layer. The Data Capture Layer components both obtain information from the Data Processing Layerr, such as current medication stored in the database, and information inputted by the user from the Data Capture Layer into the Data Processing Layer.

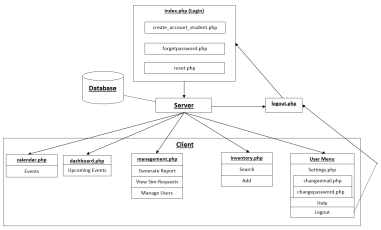

Figure 1: A simple architectural layer diagram

2.1 Client

This layer takes input from the user and stores it in the database. It also takes data from the database as well as static content for the webpages from the server in order to display the graphical user interface , along with their data, if they have any. Take the inventory page, for example. The client presses on the inventory tab. The server obtains data from the database, such as what the inventory items are, and the static content for the webpages from the server, and these are linked together to have data loaded on the actual graphical user interface to the client.

2.2 Server

This layer contains static information for the webpages, such as the CSS, Bootstrap, HTML, and php code needed to display the web page, and contains the interface that is needed to link the webpages with the data, if they needed any, as described in the client layer.

2.3 Data

This layer stores all the information that needs to be used by users of the website, such as the actual inventory items to be stored in the lab.

3 Subsystem Definitions & Data Flow

This section breaks down the layer abstraction to another level of detail.

4 X Layer Subsystems

In this section, the client layer is described in some detail in terms of its specific subsystems.

4.1 Calendar

The calendar keeps track of simulation events for each month. This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

4.1.1 Assumptions

The predinfined php structures were actually connected to the css calendar.

4.1.2 Responsibilities

The calendar allowed users to move around events on the calendar. Also, the oher responsibility was to display the events.

4.1.3 Subsystem Interfaces

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	navigation	mouse click	web address

4.2 Dashboard

The dasboard allows students to sign up for simulation events.

4.2.1 Assumptions

No assumptions were made.

4.2.2 Responsibilities

The responsibility of the dashboard was to take user input. Also, to store the information into the database.

4.2.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	textfield	variable data	true/false statement

4.3 Management

The management allowed for administrators to manage their users.

4.3.1 Assumptions

The user has to be administrator to access the page.

4.3.2 Responsibilities

The responsibility of management was to manage users by deactivating or editing them. The editing allows for the administrator to change whether the user as student, gta, simulation tech, admin, and as well as some others. Another responsibility was to accept or reject simulation requests. Lastly, management generates a month report of the hours a user has worked.

4.3.3 Subsystem Interfaces

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	buttons	mouse click	action
#2	POST	variable data	N/A
#3	naviagtion	mouse click	web address

4.4 Inventory

The inventory keeps tracks of inventory items.

4.4.1 Assumptions

No assumptions were made.

4.4.2 Responsibilities

The responsibility of the inventory was to display the inventory items from the database. Also, allows the user to edit, delete, and add inventory items.

4.4.3 Subsystem Interfaces

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	buttons	mouse click	action
#2	textfield	variable data	true/false statement
#3	navigation	mouse click	N/A

5 Y Layer Subsystems

In this section, the Server Layer is described in some detail in terms of its specific subsystems.

5.1 PHP Scripts and Interfaces

This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

Sript Subsystem is a portion of every .php file on the website. This data flows from HTML pages into the PHP scripts which use the information in SQL statements to retrieve or send data to the database. Then, PHP generates code for HTML output.

5.1.1 Assumptions

No Assumptions were made.

5.1.2 Responsibilities

The PHP script is the backbone of the website. It is hosted on a internet client with access to running PHP scripts. PHP provides a means to communicate with the database and send information to client systems. Each GUI system has a backend PHP interface to display features - such as tables, user information, and create data - which pulls or pushes data to the database.

5.1.3 Subsystem Interfaces

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	MySQL	SQL statement Connection Data	SQL Rows
#2	POST() and GET()	N/A	PlainText Variable
#3	echo	String variable	HTML Code

5.2 Static File Storage

The website's storage system. Internet host stores code, script and image files to be used on the website.

5.2.1 Assumptions

No assumptions where made for file storage.

5.2.2 Responsibilities

The responsibility of the file storage is to store scripts, style, html, and security files.

5.2.3 Subsystem Interfaces

No system interfaces for file storage.

6 Z Layer Subsystems

In this section, the Database Layer is described in some detail n terms of its specific subsystems.

6.1 Database Tables

The Database Tables Subsystem is what provides the database with all the needed information for the website. This data is connected through HTML and PHP code to gather information from the tables or even put data back into the tables properly. This allows for proper storage of information to the database so that the website portion may run smoothly and without any error or incorrect information when sought by a user.

6.1.1 Assumptions

The main assumption concerning the Database Tables was that it was an important part of the project to run efficiently.

6.1.2 Responsibilities

The Database Tables were a significant portion of the project. It stores everything needed for the actual website to display some sort fo information without hard coding data into fields. The Database Tables needed to be correct and also be connected to one each other to extract information from multiple tables and make sure that the information needed was the correct information that is being sent. This ensured any communication with the website server and database went successfully and provided the correct info needed.

6.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	CREATE DATABASE	Name of Database	Database is created with given input name
#2	CREATE TABLE	Name of Table	Table is created in current Database
#3	INSERT TO and VALUES	Table's Column's Names and Values	Information filled tables

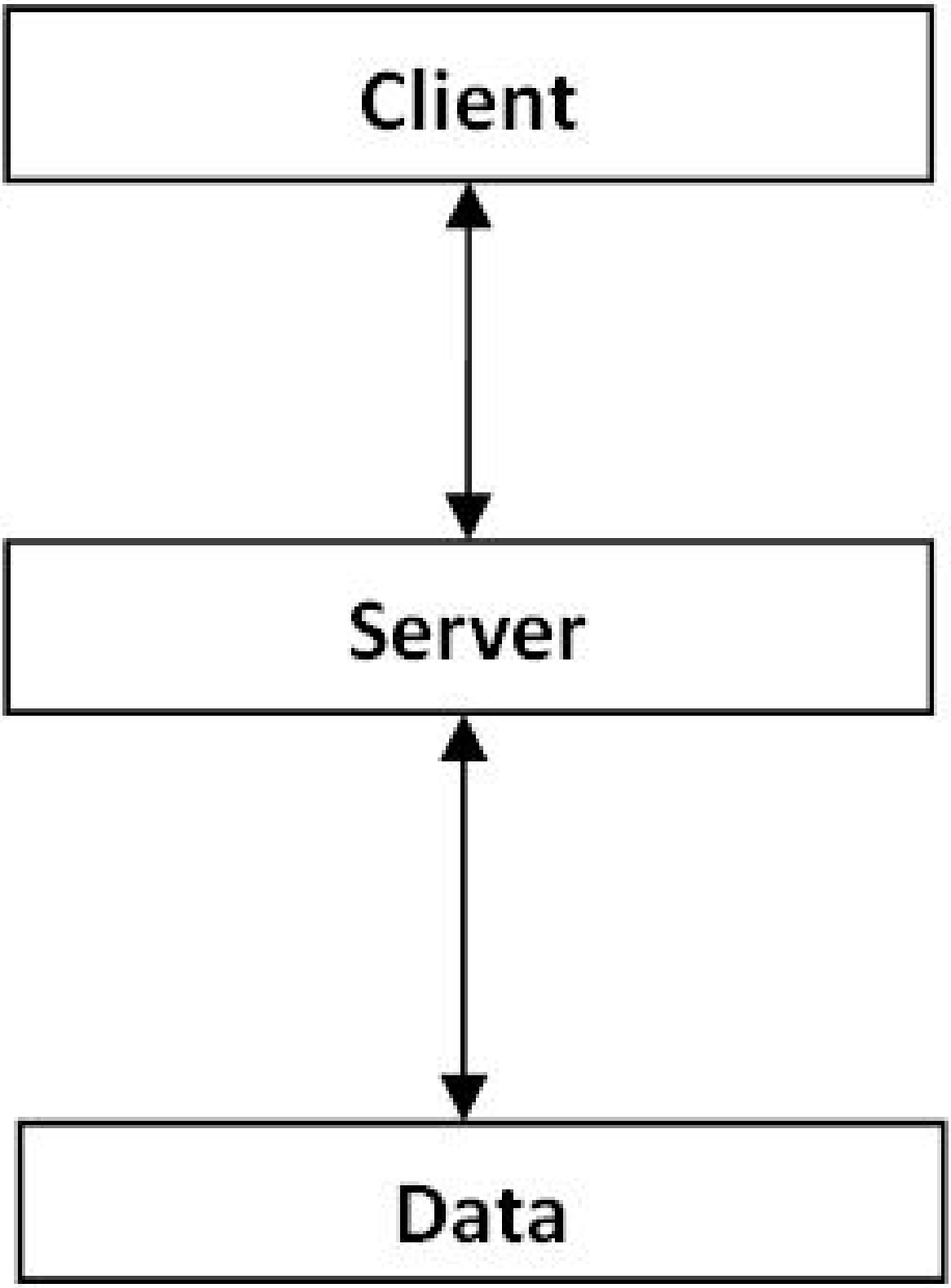

Figure 2: A simple data flow diagram

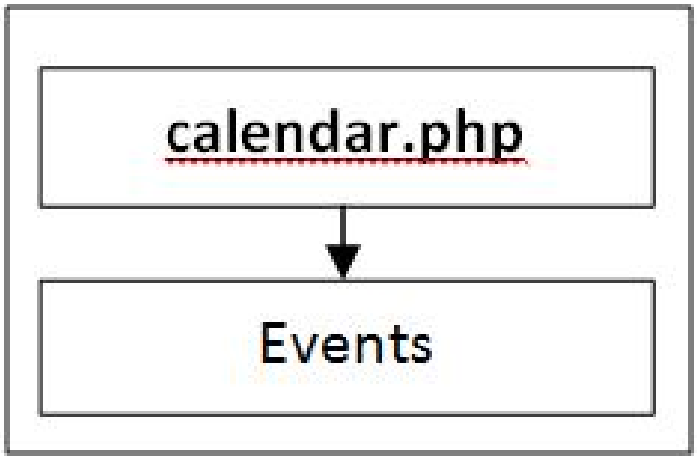

Figure 3: Calendar Subsystem Diagram

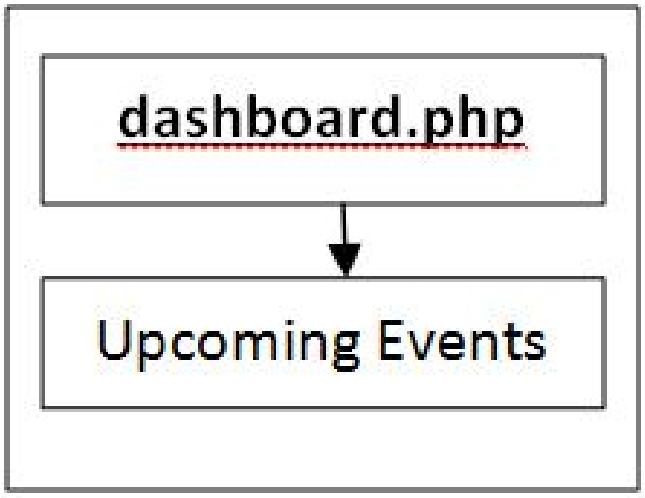

Figure 4: Dashboard Subsystem Diagram

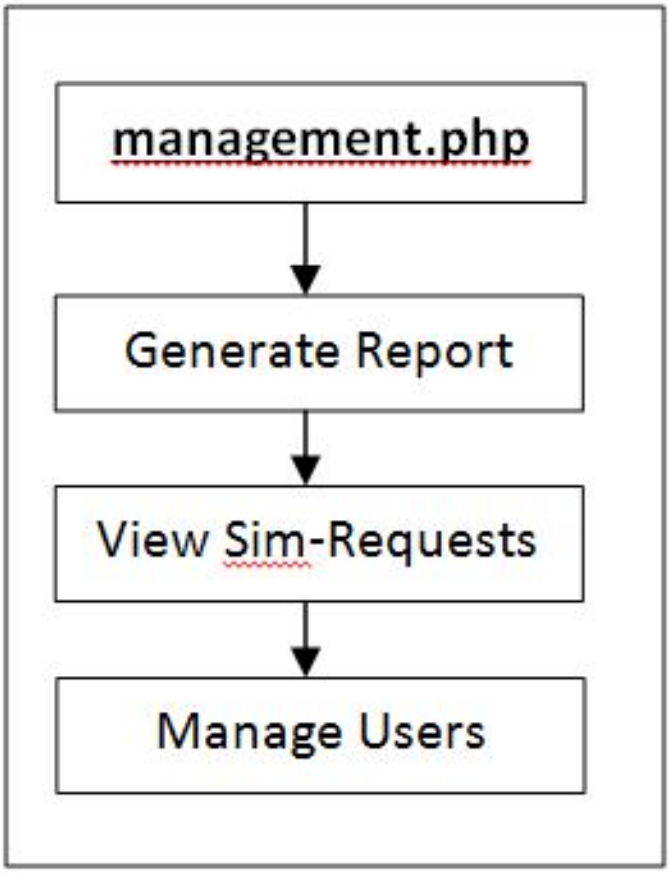

Figure 5: Management Subsystem Diagram

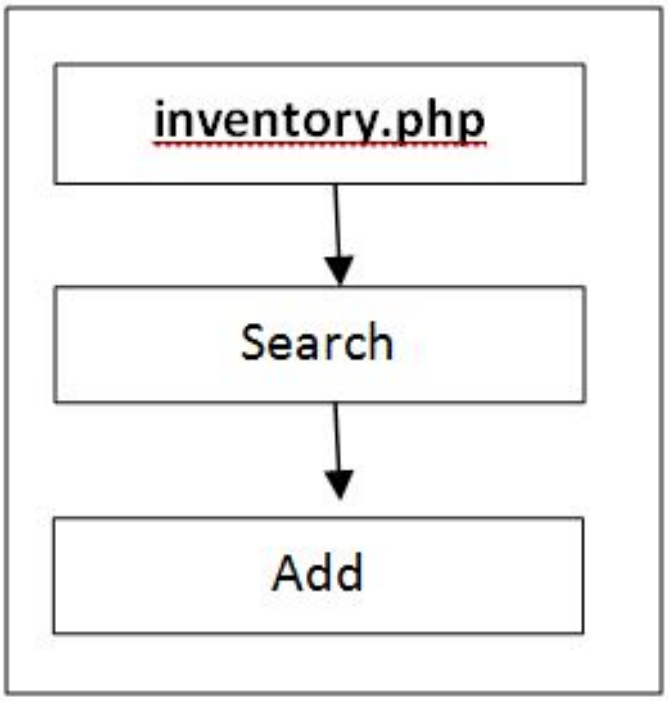

Figure 6: Inventory Subsystem Diagram

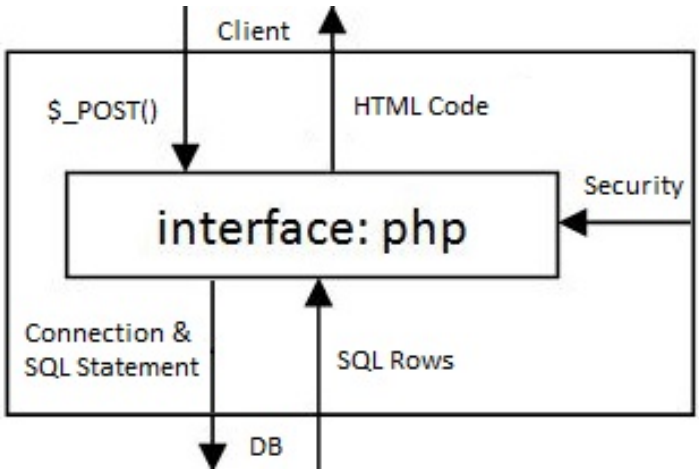

Figure 7: PHP Script echos table formatting

Figure 8: The File Browser Layout

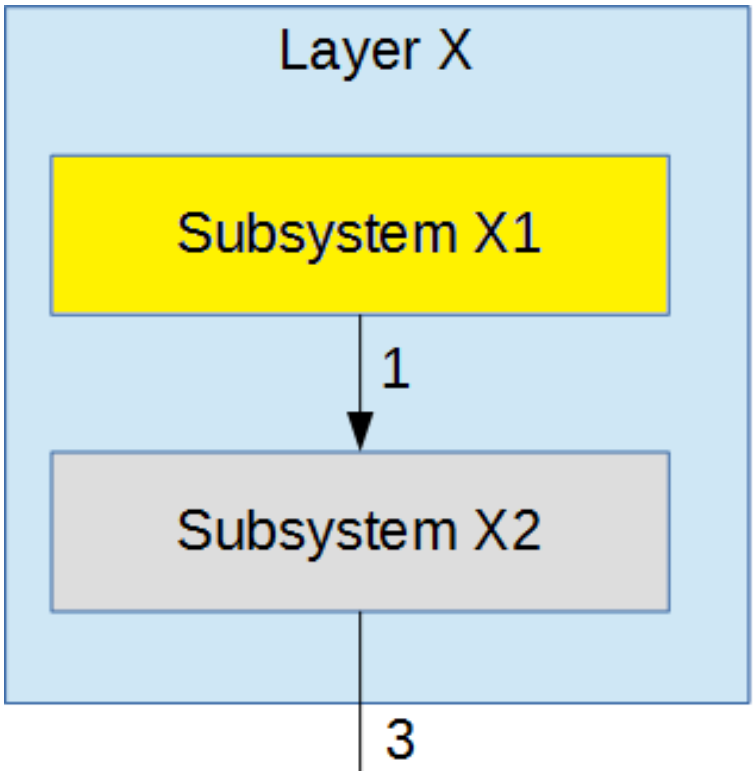

Figure 9: Example subsystem description diagram