

FINAL REPORT

For

Schedule Vista

Prepared by The Gold Team:

Mark Flores

Isaac Delgado

Marcelo Garza

Spencer Ellisor

Andrew Lopez

**Organization: Texas A&M Corpus Christi Computer Science Department-
Capstone**

5/1/2017

Project Overview and Objectives

The Nursing Department at Texas A&M Corpus Christi relies on proper allocation and preparation for classrooms, labs, and lab equipment. Instructors of the Nursing Department must have the proper room with the proper equipment set-up in order to proceed with their intended lesson. If essential scheduling and equipment set-up are not satisfied then we are inefficiently using our educational resources and hindering our developing nurses. This resource management lies under the responsibility of Ms. Eloisa Beltran the Lab Coordinator. Currently, the scheduling process consists of a single meeting between instructors who negotiate on dates, times, and rooms for the their curriculum that semester. The scheduling during this day is done through the means of a physical calendar and sticky notes. This calendar with its sticky notes is eventually given to Ms. Eloisa Beltran who will then transfer this information to Astra. However, after this scheduling day Ms. Eloisa Beltran will have to account for occasional schedule request and cancelations from instructors throughout the rest of the semester and must allocate resources accordingly so that no conflicts occur. Additionally, she must make sure that the proper equipment is set up for the specific lesson. However, instructors often do not provide sufficient information as to exactly what equipment is needed for their lab and thus providing the proper equipment is handled through *experience* or the “Lab bible”. Ms. Eloisa Beltran often passes lab-prep duties to her student workers who do not have as much lab-prep *experience* as she does and thus *experience* cannot be a reliable tool. Subsequently these student workers proceed to use the “Skill Lab Bible” (an enormous binder), which abstractly defines the requirements for hundreds of different types of labs. The scheduling system for the nursing department is clearly cumbersome with many moving parts. The computer science department is proposing (and developed) a scheduling system that will develop a system to help alleviate some of the scheduling disorder and in turn benefit the Nursing Program. This project will allow the Nursing Department instructors to schedule their future labs and courses in an organized manner, which will tremendously help Eloisa in properly allocating time and resources for nursing students.

Requirements

Plain-English Description

Our scheduling system will essentially be middleman software for the Lab Coordinator of the Nursing Program at Texas A&M Corpus Christi as Lab scheduling is made. The prime software used for recording lab scheduling is ASTRA. It is mandatory for the lab coordinator to enter the individual scheduled labs into ASTRA to make them official. However, the process of confirming official date and times require additional information and procedures that ATSR alone does not supply. In example the lab coordinator must supply proper equipment for each scheduled lab. Eloisa (lab coordinator) will be the main user of the system while the instructors of the nursing department will utilize the system for scheduling their desired labs. Eloisa will manage the site for her convenience and use this system as a supplemental tool for ASTRA.

Functional vs. Non-Functional Structured Requirements

Product Scope

The Nursing Department at Texas A&M Corpus Christi relies on proper allocation and preparation for classrooms, labs, and lab equipment. Instructors of the Nursing Department must have the proper room with the proper equipment set-up in order to proceed with their intended lesson. Instructors often do not provide sufficient information as to exactly what equipment is needed for their lab and thus providing the proper equipment is handled through *experience* or the “Lab bible”. Moreover ASTRA, the school’s official scheduling system, does not provide attributes to include this extra, yet important, information. Additionally, the scheduling process consists of a single meeting between instructors who negotiate on dates, times, and rooms for the their curriculum that semester. The scheduling during this day is done through the means of a physical calendar and sticky notes. This calendar with its sticky notes is eventually given to the Lab Coordinator who will then transfer this information to ASTRA. However, after this scheduling day the Lab Coordinator will have to account for occasional schedule request and cancelations from instructors throughout the rest of the semester and must allocate resources accordingly so that no conflicts occur. If essential scheduling and equipment set-up are not satisfied then we are inefficiently using our educational resources and hindering our developing nurses. The scheduling system for the nursing department is clearly cumbersome with many moving parts. Thus this scheduling system helps alleviate some of the scheduling disorder and in turn benefit the Nursing Program.

Product Perspective

Our scheduling system will essentially be middleman software for the Lab Coordinator of the Nursing Program at Texas A&M Corpus Christi as Lab scheduling is made. The prime software used for recording lab scheduling is ASTRA. It is mandatory for the lab coordinator to enter the individual scheduled labs into ASTRA to make them official. However, the process of confirming official date and times require additional information and procedures that ASTRA alone does not supply. In example the lab coordinator must supply proper equipment for each scheduled lab. The Lab Instructors who are requesting to schedule a lab date should provide this information but this is often not provided. Additionally slight schedule changes must be accounted for as well as checking conflicting schedule times. Thus our scheduling system will provide an interface that will handle this process. This will allow the lab coordinator to easily finalize any changes and obtain the required information from the instructors.

Product Functions

The following are the major functions of the Scheduling System:

- *Present scheduled labs graphically on a digital calendar*
- *Request labs to the admin user*
- *The system will Check for conflicts*
- *Admin will be able to adjust request forms*

User Classes and Characteristics

The Scheduling System will have two different types of user classes, *ADMIN* and *GUEST*. There will only be one Admin user at a time, in this case the admin will be the lab coordinator. All other users will be Guest users. As base functionality both users will be able to view the calendar with all its contents and pending lab schedule requests. The Guest will be able to submit a request for a lab and thus will follow through request form interface. The Admin user will be able to see the request submitted by Guest users and decide whether to reject or accept the request. If the request is accepted by the Admin, the scheduled lab be added to the globally view calendar for all users to see. The admin will also have access to a table view of all schedule information, which will include schedule labs and pending requested labs.

Operating Environment

The scheduling system will be hosted off a raspberry pi at first to get initial data. The Raspberry pi is small computer system running <operating system>. Our goal will be to move our scheduling system from this raspberry pi to Texas A&M Corpus Christi's servers. Once we have successfully transferred our system over, the system will be run off of a Red Hat Linux distribution. Regarding users of this system, their operating environment will not be depended as our system will simply hosted by on campus and their will be a URL provided to all users to access the scheduling system's functionality.

Designs and Implementation Constraints

The major implementation constraints will be allowing all users to use our system from their own computers. This scheduling system will be designed to be easily accessible similar to *blackboard* or any other academic online resources. In doing so we must be able to host the system and all its data on a separate host because the computers on campus are periodically erased to a default state. Thus, as a temporary solution we have hosted the system on raspberry pi that is locate outside of school. However we will get space on Texas A&M Corpus Christi's servers to have an official hosted location that will always be available to on campus computers.

User Documentation

All of our intended user who will be using the system will have access to a user manual on our actual product. There will be a user manual link available on both the user menu and the admin menu. This link will direct the user to another page with links to specific pdf's of different functionality of our system. These pdf documents will have screen shots and bullet listing of step-by-step instructions for their specific task.

Assumptions and Dependencies

An assumption about the system and environment that is crucial is that there will not be much scheduling traffic. Currently in the Nursing Program, the scheduling process consists of a single meeting between instructors who negotiate on dates, times, and rooms for the their

curriculum that semester. After this scheduling day there will be occasional changes to the schedule but we do not expect a lot of changes and high traffic to occur. Thus our system will not have to be robust enough to handle a heavier load of traffic. Moreover this system will only account for the rooms on the second floor of Island Hall. However there will be a feature to add rooms if necessary. An image of a map of all accounted rooms follows this section. Additionally the only dependency for our system is that it is hosted on outside the school or on the school's servers. This is vital to keep all scheduling information.

User Interfaces

Schedule Vista's first page will give the option to login as an admin user or simply begin as a guest. If the user logs in as an admin then the user will be directed to the admin menu with all its specific viable options will. Additional options include "View Appointments" which will take the user to a searchable listing of schedules already in system and a User manual that will lead the user to documentation on how to utilize Schedule Vista. Here are flow charts illustrating their logical constructs:

All guest users will click "Begin" and then will be taken to the Main Calendar view. The Main Calendar view will be very interactive, showing a full monthly calendar. The user can click on a specific date and will be greeted with request form that can be filled out. Once submitted the request will be added to the calendar as a small tab. Additionally you can click on of these tabs to get the schedule details (under an admin account the ability to delete one of these schedules will be available). The following images show screen shots of the GUI respectively:

The admin user will be able to log in once they click "Admin Login". This will take them to the log in page where they can enter their credentials (username & password). Once they successfully log in the admin will be taken to the admin menu.

Along with being able to see the full calendar the admin will be able to view the appointment listing in a searchable format with a search bar on top and bottom of the page. You will also be able to manage the rooms and see currently used rooms as well. In viewing the appointments the admin will be able to add or delete appointments accordingly with a respective menu for doing so. The same goes for managing rooms, as the admin will have the functionality to add or remove rooms from being scheduled with.

Software Interfaces

The scheduling system is constructed with HTML, PHP, JavaScript, and MySQL database. The HTML/PHP/JavaScript create the entirety of the user base functionality and Graphical User Interface. As users use this site the PHP will be enacted to initiate MySQL commands to manipulate the database accordingly. There will be request notifications send from the guest user to the admin user. This will be carried out through the database, as the admin user will simply see any additions to request queue. This system utilizes this request system to allow more organizational power to the admin by accepting or denying the requested forms.

Communications Interfaces

The requirements for using this system will include a web browser and email service. This scheduling system will be hosted on server and viewable using a URL on a web browser.

All users will simply need a to use any web browser on a computer on campus. Additionally email may be used as once the admin rejects and request from a user it is appropriate for the admin to communicate why it was rejects. This can be solved through directing the admin to an email service. This will be added late.

Nonfunctional Requirements

Performance Requirements

The scheduling pattern for the Nursing Program consists of a one-day meeting where majority of schedule labs are done. After this day moderate traffic is expected for scheduling throughout the semester. Thus the scheduling system must be able to accommodate this single high traffic day and then adhere to the mild changes later on.

Security Requirements

Our system security will be taken care of by being on Texas A&M Corpus Christi's servers. By being on these servers this will only allow system users that are currently using a campus computer. This will limit the amount of foreign users that could potentially enter meaningless request. However, even if unwanted users utilize the system they will not be able to compromise the data of the calendar, as they will have to be logged in as an admin in order to do so.

Software Quality Attributes

Software qualities that are high priority for this scheduling system are availability, usability, and update correctness of data. These are important, as the goal of this scheduling system is to improve the efficiency and ease of scheduling labs. This includes improving the communication/relationship between the lab coordinator, lab instructors, and the actual schedule (in this case our scheduling system). Our software should be readily available for all user similar to blackboard or the campus website. The software must be usable in order to accomplish an improvement in efficiency and ease of use. Finally the software must be up to date with data. In order to use a useful scheduling system and improve the relationship between our stakeholders they must be able to see the changes made by each other as soon as possible. Software Qualities with a lower priority include flexibility, maintainability, and reusability. Our scheduling system is designed to last until a better solution is found. Till then this system must be able to sustain and accommodate any new and problems that arise. This is taken care of by the admin user capabilities that include adjusting system settings to changing rooms/times.

Analysis

The major implementation constraints will be allowing all users to use our system from their own computers. This scheduling system will be designed to be easily accessible similar to *blackboard* or any other academic online resources. In doing so we must be able to host the system and all its data on a separate host because the computers on campus are periodically

erased to a default state. Thus, as a temporary solution we have hosted the system on raspberry pi that is located outside of school. However we will get space on Texas A&M Corpus Christi's servers to have an official hosted location that will always be available to on campus computers. All of our intended user who will be using the system will have access to a user manual on our actual product. There will be a user manual link available on both the user menu and the admin menu. This link will direct the user to another page with links to specific pdf's of different functionality of our system. These pdf documents will have screen shots and bullet listing of step-by-step instructions for their specific task. An assumption about the system and environment that is crucial is that there will not be much scheduling traffic. Currently, in the Nursing Program, the scheduling process consists of a single meeting between instructors who negotiate on dates, times, and rooms for their curriculum that semester. After this scheduling day there will be occasional changes to the schedule but we do not expect a lot of changes and high traffic to occur. Thus our system will not have to be robust enough to handle a heavier load of traffic. Moreover this system will only account for the rooms on the second floor of Island Hall. However there will be a feature to add rooms if necessary. An image of a map of all accounted rooms follows this section. Additionally the only dependency for our system is that it is hosted on outside the school or on the school's servers. This is vital to keep all scheduling information.



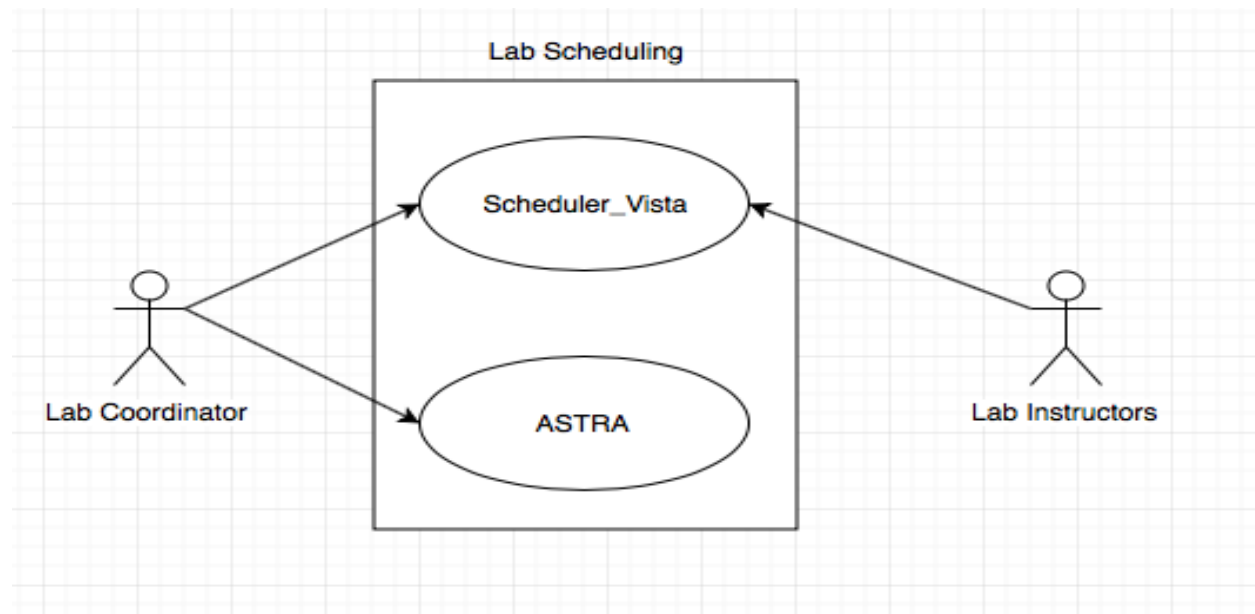
(Figure 1 - Example Physical Calendar the Nursing Program utilizes)

Design

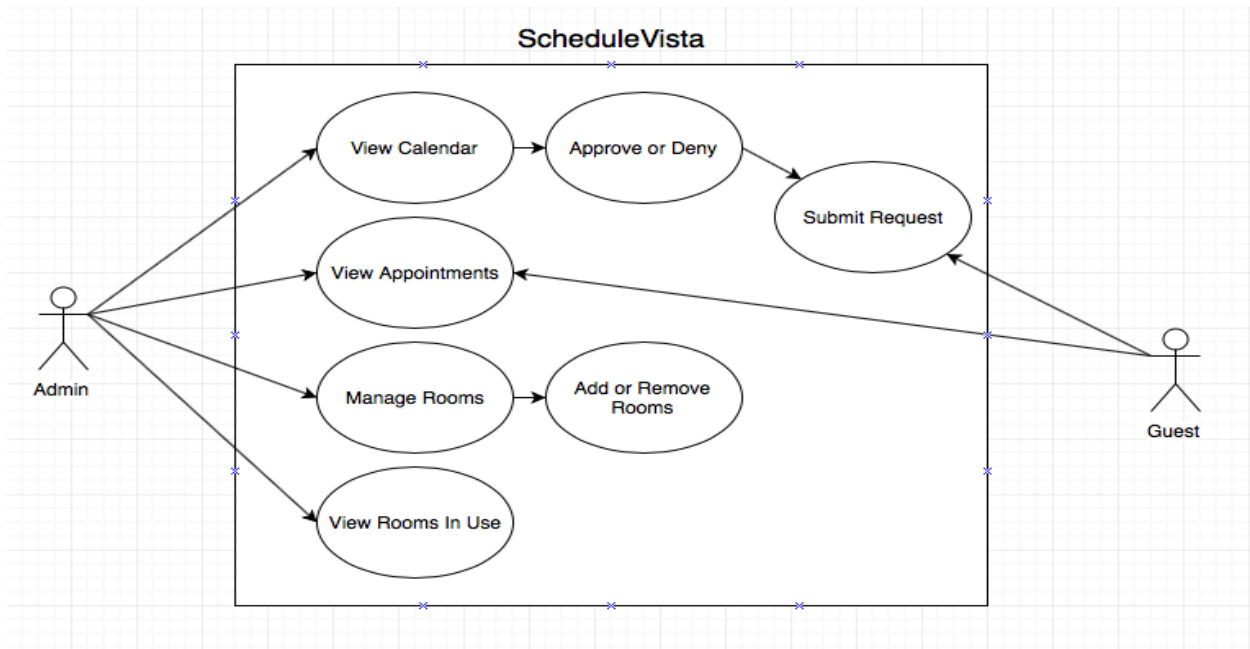
Our initial design for our system would label users with either an admin or guest user status. Every user will begin as a guest user and only if you have admin credentials (username

and password) will you be able to upgrade your user status to admin. We had designed our system like this, as there will only be one admin user majority of the time. The admin user will be the lab coordinator specifically Eloisa. Most of the users will be guest users and thus removing any kind of credentials for guest users seemed logical and user friendly. All users will be able to view a calendar to visually see appointments that already scheduled or not scheduled. This calendar view will be the main interface as we intended it to be as close to the physical calendar as possible. The closer we can make it to their usual way of implementing these schedules the more user friendly (and useful) our system will be. There will be a universal form that will allow users to schedule a request and this will allow us to gather information to find conflicts a lot quicker and thus improve their scheduling. On the admin side we will allow the maintenance of the information gathering and calendar info manipulation by creating functions readily available to the admin. Following are design diagrams of the major functions of our system.

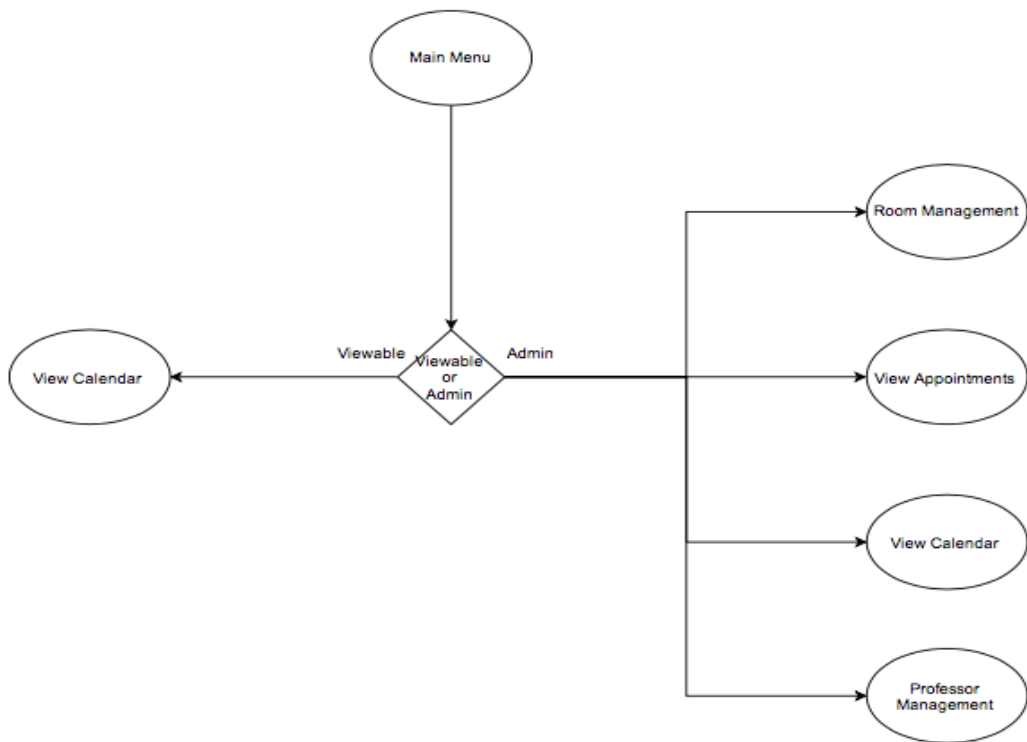
Design Diagrams (Class, Use Case, Sequence...etc.)



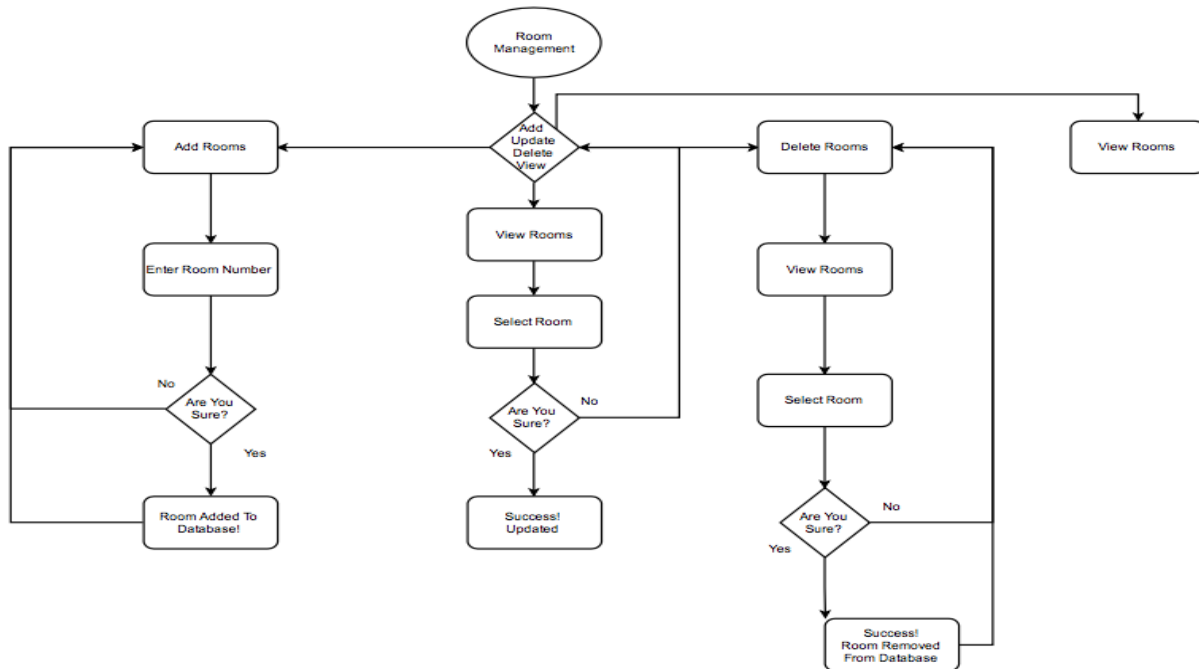
(Figure 2 - Relational System Use Case Diagram)



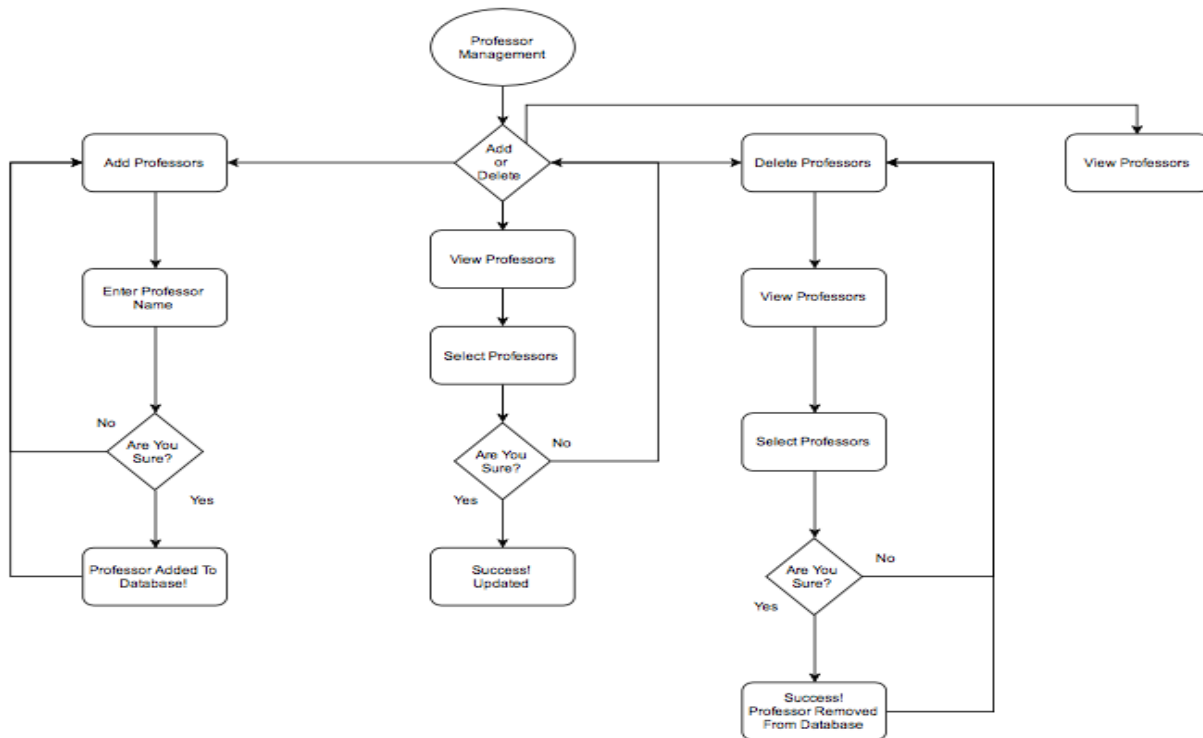
(Figure 3 - Functionality Use Case)



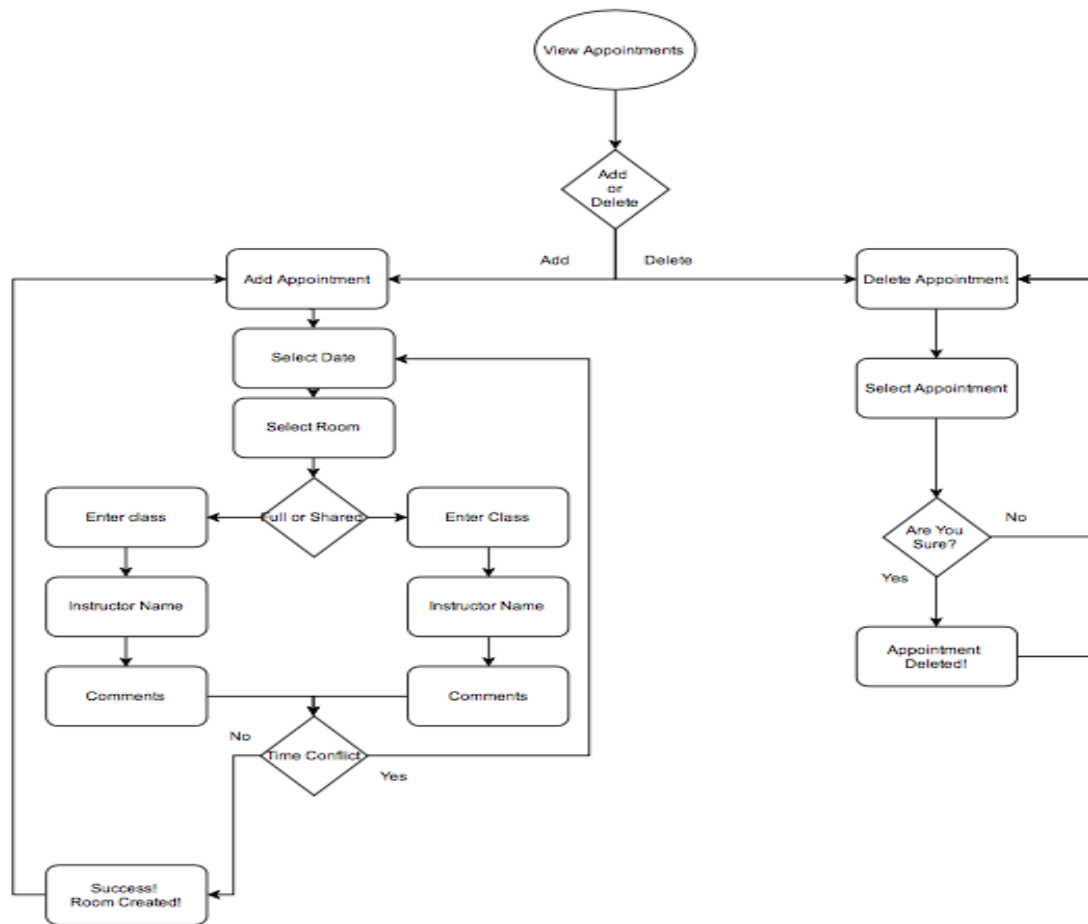
(Figure 4 - Welcome Screen)



(Figure 5 - Room Management Flow Chart)



(Figure 6 - Professor Management Flow Chart)



(Figure 7 - View Appointments)

Implementation/Details (languages used, environment, etc)

The scheduling system is constructed with HTML, PHP, JavaScript, and MySQL database. The HTML/PHP/JavaScript create the entirety of the user base functionality and Graphical User Interface. As users use this site the PHP will be enacted to initiate MySQL commands to manipulate the database accordingly. There will be request notifications send from the guest user to the admin user. This will be carried out through the database, as the admin user will simply see any additions to request queue. This system utilizes this request system to allow more organizational power to the admin by accepting or denying the requested forms. This scheduling system will be hosted on server and viewable using a URL on a web browser. All users will simply need a to use any web browser on a computer on campus. (URL: <http://schedulevista.hopto.org/>)

The following are some images of our implemented GUI for our system:

Capstone Project Prototype



Begin

View Appointments

Admin Login

Home Page

Brought to you by: Gold Team, Inc.

(Figure 8 - Welcome Screen)

Welcome to Schedule Vista

<	>	today	March 2017				month	week	day
Sun	Mon	Tue	Wed	Thu	Fri	Sat			
26	27	28	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	4	5	6	7	8			

(Figure 9 - Full Calendar View)

←

→

week

day

Sun	Sat
	4
	11
	18
	25
	1
	8

×

Title

Fundamentals Lab

Professor

Josh lopez

Shared

Yes

Start date

Tue Mar 14 2017 03:00:00 GMT+0000

End date

Tue Mar 14 2017 05:00:00 GMT+0000

room

276 ICU

Class Size

15

Resources

ICU dummies and 2 extra beds

Date Added

2017-03-20 00:12:35

☐ Delete event

Figure 10 - Request Schedule Form when you double click a day on the Calendar

Welcome to Schedule Vista

[<](#)
[>](#)
[today](#)
[March 2017](#)
[month](#)
[week](#)
[day](#)



Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
		3a Fundamenta		4a Skills Lab 3		
19	20	21	22	23	24	25
5a Co-hort Meet			1a Experimenta			
26	27	28	29	30	31	1
2	3	4	5	6	7	8

(Figure 11 - Calendar after schedule request forms have been submitted)

[Back](#)

Capstone Project Prototype Room Management

[Add Room](#)
[Update Room](#)
[Delete Room](#)

Powered by  and 

Building	Room
Island Hall	286 Bed 1
Island Hall	286 Bed 2
Island Hall	286 Bed 3
Island Hall	286 Bed 4
Island Hall	286 Bed 5
Island Hall	286 Bed 6
Island Hall	286 C.A - Breakout Room
Island Hall	286 B.A - Breakout Room
Island Hall	282 Breakout Room
Island Hall	285 Breakout Room
Island Hall	276 ICU
Island Hall	277 Breakout Room
Island Hall	264 Computer Lab - 46 seats
Island Hall	265 Computer Lab - 51 seats
Island Hall	288 Skills Lab

(Figure 12 - Manage Room Main Menu)

Capstone Project Prototype Professor Management

Back

Add Professor

Update Professor

Delete Professor



First Name	Last Name
Isaac	DelGado
Jesus	Hello
Josh	Iopez
Marcelo	Garza
Mark	Flores
Spencer	Ellisor

(Figure 13 - Manage Professor Main Menu)

Testing Activities

In testing our system we were able to get some testing data from Eloisa Beltran. This testing data was calendars from previous semesters as seen on figure #. This test data included the professor names along with their desired time and date. We utilized this data to test our conflict functions as well inputting as realistic information as we could. However, during our early development phases we utilized dummy data of our own as seen in the screen shots of our GUI. This allowed us to create intense data testing by creating unlikely scenarios. If our system could hold up through a storm of chaotic scheduling them it will be able to stand the scheduling of the nursing program labs.

Testing Description

In the development of this web application, we tested the system by creating fake appointments named “Fundamentals”. In the process of this, we utilized our own data in the form fields to ensure that the information was being stored in the database. The test proved successful, if the appointment was stored on the calendar. This can be seen on Figure. 11, as to how scheduled appointment would appear on the screen. Once the appointment was clicked, if we were able to see the information for the appointment that was scheduled, it proved that the test was successful as well. An example of this can be seen on Figure. 10.

Testing Example Procedure

- 1) Click a blank date on the calendar, or fill out the form below the calendar.
 - 2) If in step 1 the blank space was clicked, fill out the appearing form with test data.
 - 3) If appointment is rendered on screen, test was successful.
 - 4) Double click scheduled appointment.
 - 5) If Appointment is filled with correct information, test was successful.
- NOTE if testing recurring events, must use bottom form and select recurring checkbox.

Meeting Minutes

Week 4

2/8/17 Meeting Notes – Marcelo and Josh

Reviewed various options, designs, and methods for implementing MySQL to be used in this project. We also reviewed SQL Syntax and Flow Charts that we designed in accordance to what the client described.

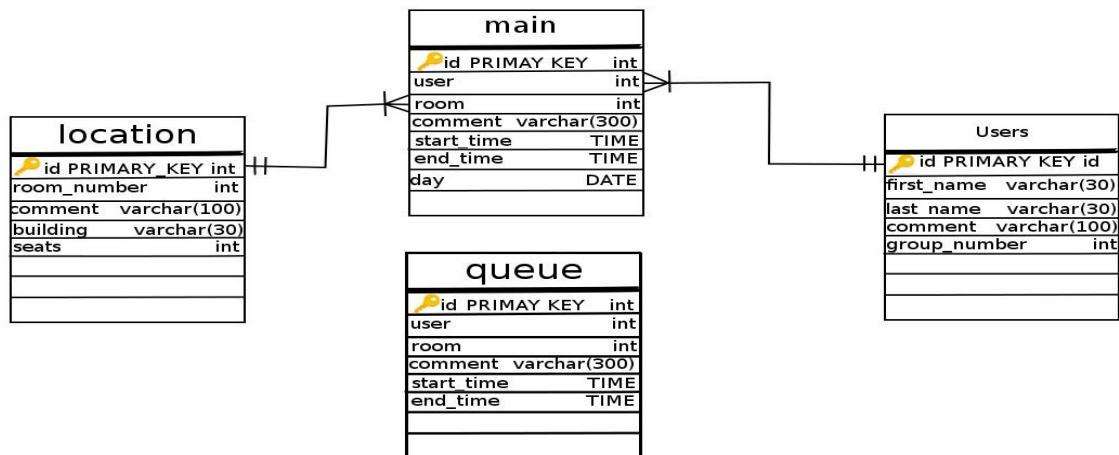
2/9/17 – Mark and Josh

Reviewed MySQL tables and compared the relationship logic between the tables to the logic designed in the Flow Chart. Made a few adjustments to both the relationship tables, and the flow chart logic, so that both entities agreed with each other. Discussed possible location of database for clients use, and discussed security measures taken to protect information in database from unauthorized use.

2/10/17 – Group Meeting

Reviewed and edited flow chart to better suite database needs. Group also discussed various used case scenarios for the design of the graphical user interface that will be used when developing software. Worked on editing database crow chart to further relationships between each table used. Along with discussing client interface problems and possible calendar issues, we also utilized a prototype demo for the database to get a viewable understanding of how tables will interact with each other upon user interaction. Group decided on further project organization, such as possible scenarios of how the software will be utilized and taking into account possible conflicts and errors that could occur while scheduling.

Database Crow Foot Diagram



Flow Chart

Week 5

Met with Eloisa to discuss initial design and concerns about how project will be developed. Josh, modified database to further replicate requirements of the client. Flow charts were further edited, and were divided onto multiple pages instead of being displayed on a single page. Group also discussed alternative languages to program the system in, incase client does not agree with Java as the primary language for the system. Group also further discussed where database will be stored, and how it will be connected to the system. Current testing for the database is done on html and php, group is beginning how to replicate such testing in a java based system, or how to run a java based system within an html page. Group also constructed project management document using Microsoft project, in which we indicated the time line for development of the system. In addition to this, group also created a rough draft for the SRS document which serves to outline how the system will function.

Week 6

Group met after class on Monday, to finalize decision to switch project over to a web based application coded in PHP. Flow charts for design of the project were then further updated at the request of the client.

Tuesday, Feb 21, group met again to further develop application. At this meeting, it was decided that the project would be relocated to a specific raspberry pi which would be shared by all users in the group so that the files of the software would not get lost or mixed up when transferring code. All changes to the project were to be made on the raspberry pi hosting the application, so that they would be centralized and up to date for everyone in the group.

Thursday, Feb 23, members of the group met with Dr. Sheta to ask for advice on how to better the current Microsoft Project file. After receiving advice on how to better the document, the document was updated to further reflect the progress the group was making towards developing the software.

Friday, Feb 24, the current issue regarding time conflicts within the system when scheduling appointments, was partially fixed. Whenever a user would make an appointment with the system, it would not check for time conflicts, however this is in the process of being fixed and near complete. Also, group met with client to discuss progress made with the project, and gather input on how certain situations would be handled when using the system. Client made suggestions on approving the GUI that was demonstrated, and group further discussed how to handle time conflicts as well as the calendar issue.

Sunday, Feb 26, group met to work on styling for the project and work on developing the calendar that will be used in the project. Basic calendar with no functionality was developed for the application, and styling was added to some parts of the application. The styling and calendar will be shown in homework 6 PowerPoint.

Week 7

- 3:05 Demonstrated progress of application with client
- 3:10 Client expressed satisfaction with state of the application, and asked about making aesthetic changes to the system
- 3:15 Group asked client if information in database was authentic with her notes
- 3:18 Group took note of missing pieces of information to add to the database. I.E. Room Numbers
- 3:20 Group informed client of upcoming day in which no more functionality may be added to the system, and that she will have to sign the SRS document.
- 3:20 Client agreed that there was no more functionality that she could think of adding at the moment, but will keep that in mind.
- 3:25 Group discussed with client how the system prototype will be tested, and discussed how the group would handle the situation if any problems arise during testing.
- 3:30 meeting ended.

Major Code Components

The application was employed through the three tier solution mode, the database tier, the logic tier, and the presentation. The core of the system was the database. The database was implemented through the MySQL relational database software. The application utilized the database for recording teacher names, room names, and appointments in various tables that had relationships with the a main table which contained the times for the appointments being schedules. Also in the main table were the identification numbers for the instructors, the primary key identifying the appointment that was scheduled, and the identification numbers identifying the various rows from other tables. The database also consisted of the users table, which contained the instructors, location which contained the rooms and their names, room_block which defines a date range in which a room can't be scheduled through the application. The logic layer was programmed in php, which handled the data inside of the database via insertion, updates, and deletion. The database's most significant feature was the ability to detect time conflicts. MySQL doesn't have an if statement inside of it's syntax that could be easily used for rejecting insertions if they conflicted with an appointment. So in order to detect time conflicts with the appointments that currently existed in the main table, php would issue a query which searched the table to find a row that had a record in which the times currently attempting to be inserted was in between existing times, or where the begin time was less than an existing begin time, and the end time was greater than an existing end time. All of these conditions were placed in the query in the sequence of or statements and with the between keyword. If the query detected conflicts/had it's parameters satisfied by what was given in the query, MySQL would return rows in which satisfied the conditions that were placed in the insertion query. PHP has a function which dumps the number of rows returned and stored it into a variable, whenever MySQL returned rows for the conflict testing features, the application would display a message to the user indicating that a conflict was detected and the insertion query will not execute.

Another significant feature of the application was the ability for the client to schedule an appointment in multiple rooms. To accomplish this, a bridge table was implemented which contained the appointment identification number, which was the primary key from the main table, and the room identification number which was the primary key from the location table. An SQL query with an inner join was executed in order to find the rooms in which the appointment was associated with, the query would then return the times for each appointment and the rooms in which they were using. The addition of an identification number for a room at the end of the query would check which appointments were using the room, and at what time those rooms were being used. The PHP end of the application would check each room sequentially through a loop which compared the begin time and end time of an appointment, if at least one room returned a conflict, the insertion query is not executed, if the loop doesn't detect any conflicts, the insertion query is issued and an appointment is created. Because the bridge table was being used to associate an appointment with multiple rooms, the insertion query needed to be executed more than once, one for each room that will be added to the bridge table.

The top section of the calendar is a form, which allows the users to input the appropriate fields when making an appointment. The bottom section of the calendar is another form, but is used to only hold the appropriate fields retrieved from the database for an appointment that is scheduled.

When the form is filled out, it is posted to another php file called Add Event, this file is where the time conflicts are checked, and where the fields are inserted into the database. There is also another form in the calendar file for the user to fill out. This form has the option for recurring events, if the user wishes to schedule a recurring event. At the bottom of the form, the fields are retrieved from the database, and are binded with javascript, which allows them to display when a scheduled event is selected. For some of the fields in the calendar, there is some php code embedded within the html, which access the database and queries data for the user to select. This is done for dynamic data, so that the user can choose from the database when filling out the form. On the second form in the calendar, which displays the details of a scheduled event, this also takes place in order to query the data that was scheduled into the database. This method is only used for dynamic data not static data.

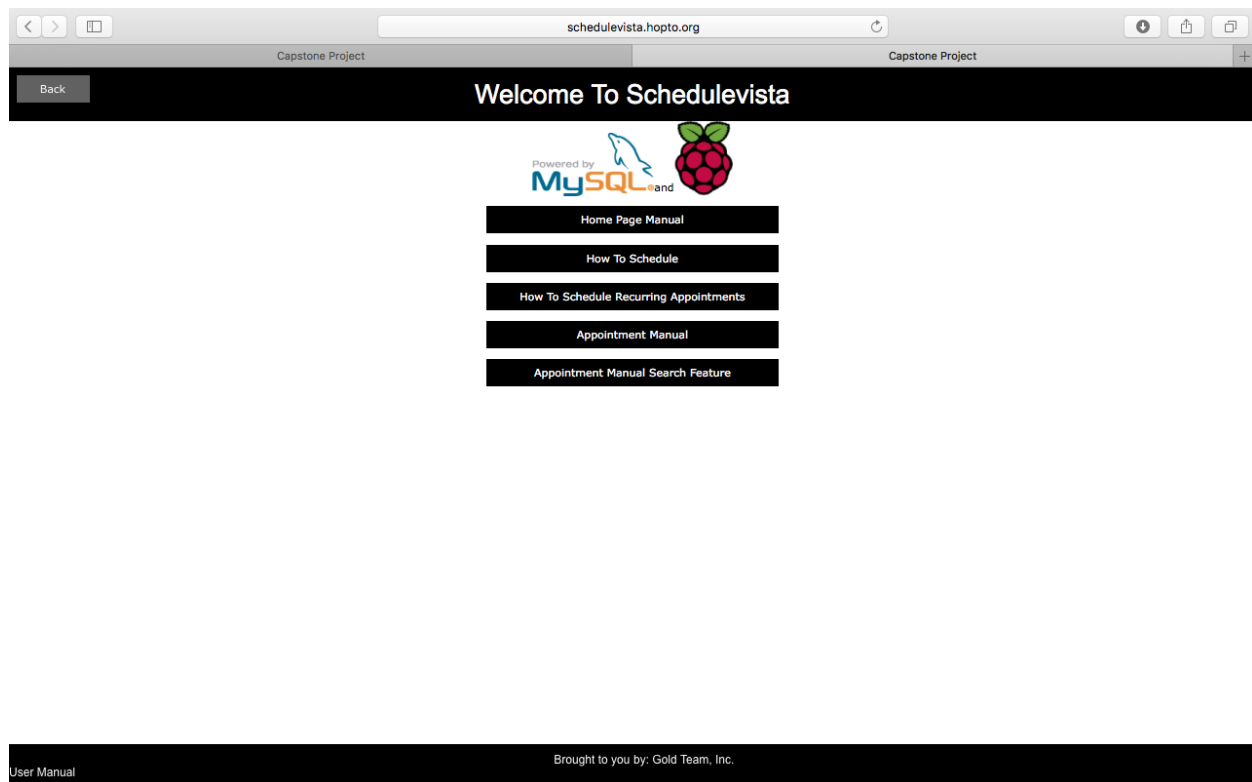
The css used in this project is used for the User Interface and is a modified CSS which was open source from W3schools, and by definition CSS is a way of formatting HTML elements with classes and parameters that predefined, using CSS to specify the parameters thou we are able to change the look and feel of the on screen elements. An example of this would be the "Begin" Button on the Homepage, by setting the HTML class to equal "button black" classes set in the TSS style sheet we are able to get the button to have a black background on the button input tag, code below:(note there can be additional styling done in the style equal tag area in HTML but its not viable to crowd that area of the code.)

```
<button class="button black" type="submit" style="width: 200px">Begin</button>
```

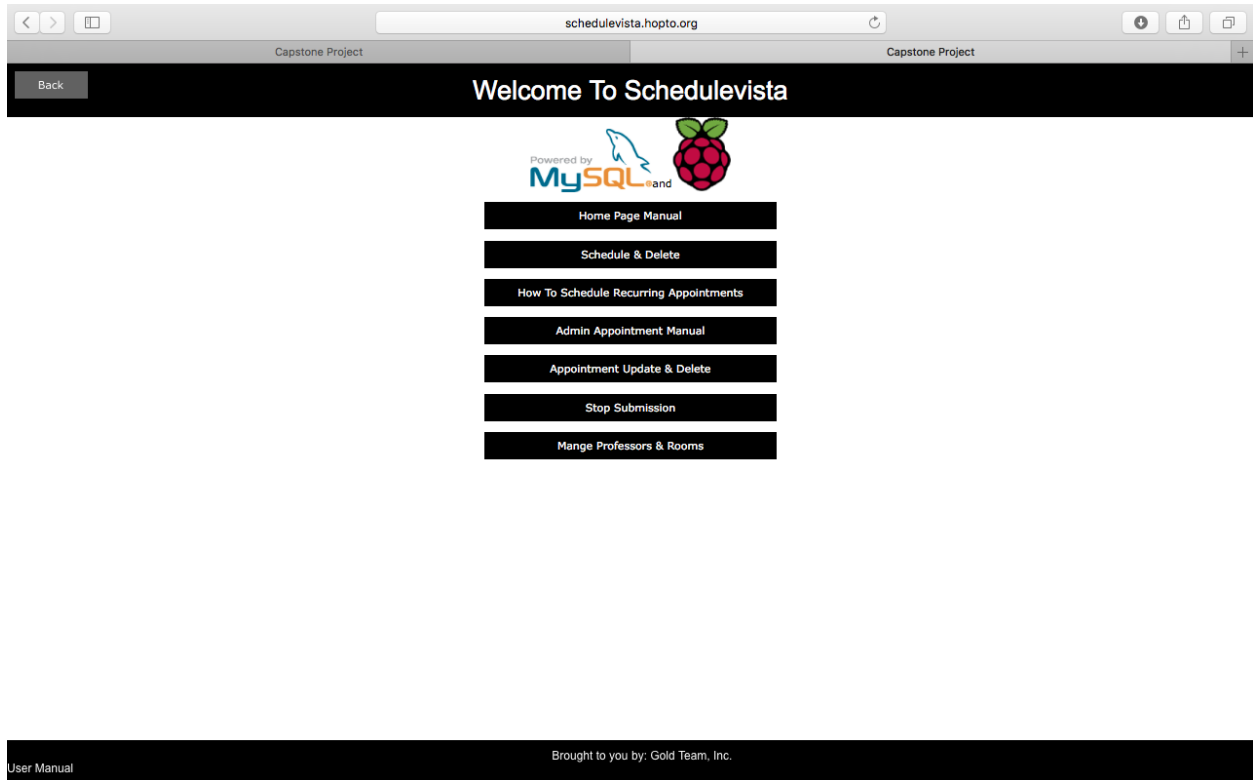
The forms use the 2 CSS style sheets to be formatted in certain way, which is how they look on screen, and also the forms are using the HTML <form> tag which can then be styled with the CSS sheets.

User Manual

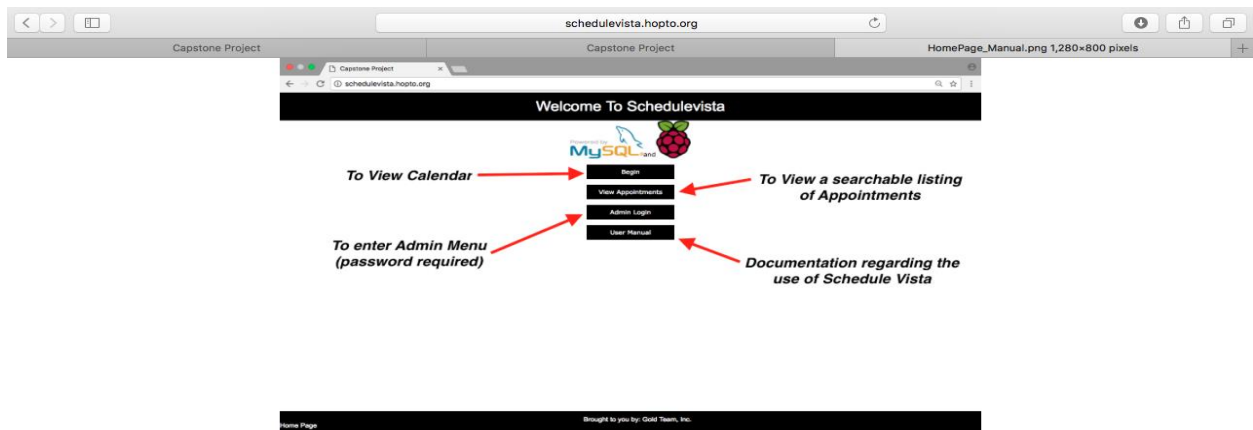
The user manual was designed to be very image based. There will be a link available on the user and admin menus that will open up user manual associated with that particular user. The user manual for guest have “Home Page Manual”, “How to Schedule”, “How to Schedule Recurring Appointments”, “Appointment Manual”, “Appointment Manual Search Feature”. The admin manual “Home Page Manual”, “Schedule & Delete”, “How To Schedule Recurring Appointments”, “Admin Appointment Manual”, “Appointment Update & Delete”, “Stop Submission”, and “Manage Professors & Rooms”. These user manuals will include mainly screenshots with red arrows, circles, and squares to be as user friendly as possible. The manuals will be as reference oriented as possible. The following are screen shots of the user manual menus and as well as an example manual document.



(Figure 14 – Guest User_Manual)



(Figure 15 – Admin User_Manual)



(Figure 16 – Example Manual Document)

Known Issues Recommended Future Work

The scheduling system definitely needs to be hosted on campus. As of writing this report the scheduling system is being hosted on a raspberry pi. The system may be hosted on campus after this report was written but this is the main concern. Additionally future concerns and room for improvement are joining a system similar to ASTRA and Schedule Vista in order to make one cohesive system. This will allow for a more embedded and concise structure for Eloisa however our system is definitely the first step to that direction.

Appendices

- (Figure 1 - Example Physical Calendar the Nursing Program utilizes)
- (Figure 2 - Relational System Use Case Diagram)
- (Figure 3 - Functionality Use Case)
- (Figure 4 - Welcome Screen)
- (Figure 5 - Room Management Flow Chart)
- (Figure 6 - Professor Management Flow Chart)
- (Figure 7 - View Appointments)
- (Figure 8 - Welcome Screen)
- (Figure 9 - Full Calendar View)
- (Figure 10 - Request Schedule Form when you double click a day on the Calendar)
- (Figure 11 - Calendar after schedule request forms have been submitted)
- (Figure 12 - Manage Room Main Menu)
- (Figure 13 - Manage Professor Main Menu)
- (Figure 14 – Guest User_Manual)
- (Figure 15 – Admin User_Manual)
- (*Figure 16 – Example Manual Document*)

Client Biography

Ms. Eloisa Beltran, BSN, RN

Elosia M. Beltran, BSN, RN received her Bachelor's Degree from Texas A&M University Corpus Christi. She is currently responsible for the Nursing Learning Resource Lab in which she coordinates and oversees students practice lab skills and skill check offs for junior and senior nursing students.

She started her career in Adult and Pediatric Medical Surgical nursing. However, most of her experience for the last fourteen years has been in surgery. She worked nine years in Perioperative Nursing in both the inpatient and outpatient settings, and five years as a Director of Surgical Services in an outpatient setting.^[1]

Contact Information:

Email: eloisa.beltran@tamucc.edu

Phone: 361-825-2353

Member Biographies

Mark Flores

I am a Computer Science student at TAMUCC with a concentration in Information Systems. I currently hold a Bachelors Degree in Health Sciences, and will graduate with my Computer Science degree May 2017. I currently know how to program in ASP.NET,C#, C,C++,COBOL, HTML , Java, and MS SQL, MySql, and PHP. I know how to be a team leader, and a team player and I work well in groups and individually.

Contact Information:

Email: mflores524@yahoo.com

Phone: 361-946-8602

Spencer Ellisor

I am a computer science major at TAMUCC in the information system track. I've taken a wide variety of classes here in the computer science field. I know how to code in COBOL, C++, C, Java, HTML, CSS, PHP and My SQL. I've got experience making databases and I am currently getting experience in web programming. I have become acquainted with PHP over the course of my Capstone project and have experience working in a variety of groups.

Contact Information:

Email: srellisor@hotmail.com

Phone: 512-900-0606

Isaac Delgado

I am a computer science major at TAMUCC with a concentration on systems programming. I will have a minor in Mathematics and Philosophy alongside my major. I plan to pursue a masters in computer science. I have a lot of experience with C/C++ and Java Programming with new experience with HTML/CSS/JavaScript/ASP.net languages and frameworks. I have worked with many groups throughout my undergraduate and developed team oriented work style.

Contact Information:

Email: isaacdelgado1994@gmail.com

Phone: 210-965-0094

Andrew Lopez

Ever since I was in high school, I have had an interest in the sciences. But late into high school I started to make the decision on who I wanted to be and what I wanted to do. Ultimately it came down to what would earn me the greatest salary, Let's just say my priorities have changed since then since I study what I do because I enjoy it more than I desire the potential paycheck I would reap from it. Initially I wanted to be an electrical engineer, mostly because I was routinely taking things apart and assembling them back together again or rebuilding them into completely different machines, I was convinced that this might be a potential path when I successfully converted an old camping Cathode Ray Tube television into an oscilloscope. Low quality but it was able to display the waves from the alternating currents or audio signals that you would feed it. One semester in high school, I took the electrical trades class, since I went to a very low income school, the electrical engineering program was confined to residential wiring and assembly, not so much circuit design and testing like I had expected. I have since then changed my focus to computer science since the school recently obtained an instructor that was teaching the Cisco discovery program (small time CCENT certification). Though I do plan to go into this route, I still have a working knowledge of electrical circuits and components like transistors, capacitors and resistors. The reason I gave up on engineering and only do circuit design as a hobby was the intense amount of mathematics involved in the field, I found it overwhelming and thought that computer logic would have been a less intense. The capstone course has helped me discover some new skills that I didn't know I have. For the capstone project, I devised much of the database portion of the application and integrated it into the php web application. I am now expanding my outreach in jobs for full stack developer. Because of the culminated amount of skills and knowledge that I have obtained through my college career, I am now seeking jobs in computer network and administration, database development and administration, cyber security, and software development despite my weaknesses in C# and some of the more complicated C++ programming features, I might start learning MPI programming through C++ very soon.

Contact Information:

Email: carcrazy1102@gmail.com

Phone: 361-944-2614

Marcelo Garza

I am Marcelo L. Garza, and this is a short description of my professional field and work. I am a student graduating here in Spring 2017 from TAMUCC with a Bachelor Degree in Computer Science. In being a CS professional familiar with C++, JAVA, C#, and web programming languages, I have experience focused in the game design and development area. Throughout my college career I have lead several groups as head of a programming team, and I feel that in order to complete a programming project the consideration for end users should be taken very seriously, first and foremost in the design and functionality areas.

Contact Information:

Email: mgarza66@islander.tamucc.edu

Phone: 956-792-1403