

University of Texas at Arlington

MavMatrix

2016 Fall Honors Capstone Projects

Honors College

12-1-2016

THE SMART HOSPITAL MANAGEMENT SYSTEM

Neelim Haider

Follow this and additional works at: https://mavmatrix.uta.edu/honors_fall2016

Recommended Citation

Haider, Neelim, "THE SMART HOSPITAL MANAGEMENT SYSTEM" (2016). *2016 Fall Honors Capstone Projects*. 4.

https://mavmatrix.uta.edu/honors_fall2016/4

This Honors Thesis is brought to you for free and open access by the Honors College at MavMatrix. It has been accepted for inclusion in 2016 Fall Honors Capstone Projects by an authorized administrator of MavMatrix. For more information, please contact leah.mccurdy@uta.edu, erica.rousseau@uta.edu, vanessa.garrett@uta.edu.

Copyright © by Neelim Haider 2016

All Rights Reserved

THE SMART HOSPITAL MANAGEMENT SYSTEM

by

NEELIM HAIDER

Presented to the Faculty of the Honors College of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

HONORS BACHELOR OF COMPUTER ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2016

ACKNOWLEDGMENTS

I want to thank my Mom and Dad for helping me obtain this degree. I admit, there were times when I felt like it was extremely hard and this degree was not worth pursuing, but their pushing me forward through this plan and their constant help enabled me to obtain not only a degree in Computer Engineering, but an Honors Degree in Computer Engineering.

I especially want to thank my sister and brother, who were always there to cheer me up when it got hard, and gave me advice not only for my academic life, but more importantly, for the rest of my life.

I also want to thank the professors who were kind to me, were willing to listen to my problems, and gave suggestions as well as take feedback from me in order to improve me and their own selves.

I want to thank my friends, both in my Senior Design class and from all of my classes. They helped me a lot, and there were times that I simply could not have gotten through without their help.

I finally want to thank Allah, God the one and only, for helping me get through these years. There were times that I did not think I could have made through without relying on him and seeking patience with him. He was the one who really helped me in the end.

November 28, 2016

ABSTRACT

THE SMART HOSPITAL MANAGEMENT SYSTEM

Neelim Haider, B.S. Computer Engineering

The University of Texas at Arlington, 2016

Faculty Mentor: Christopher McMurrrough

The UTA Smart Hospital is a training facility for nursing students. Our goal is to create a web platform that allows the faculty to manage their hospital information in one place; previously, this information was scattered among different websites and files. We first derived requirements for the website, and using these requirements, our team then created a prototype of a website. It was found that our website had the capability of implementing most of the features needed by the hospital, yet we could only implement features that were considered to be a higher priority. In conclusion, the website was able to accomplish the most important feature, which was being able to provide interfaces that allowed the hospital to manage information in one place. Our project has shown that it is a worthy investment to create a website that manages the Smart Hospital's information.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iii
ABSTRACT.....	iv
LIST OF ILLUSTRATIONS.....	vii
Chapter	
1. INTRODUCTION	1
1.1 Statement of the Problem.....	1
1.1.1 Main Goals.....	1
2. LITERATURE REVIEW	3
2.1 A Study of Existing Resources Used in This Project	3
2.1.1 timeclockwizard.com.....	3
2.1.2 MySignUp.com.....	5
2.1.3 Quartzzy.com.....	9
2.2 Other Computer Applications and Web Interfaces Built for Hospitals ...	9
2.2.1 Hospital Patient Database Management System Concept by S. A. Asabe, et. al.....	11
2.2.2 The Intelligent Agents in Hospital Appointment Scheduling System by A. Hylton III and S. Sankaranarayanan	11
2.2.3 A Dynamic Platform for Workflow Management using Web Services; a Hospital Scenario, by N. Prinyapol, et. al	12
3. METHODOLOGY	14
3.1 Requirements Gathering	14

3.2 Participating in Scrum Meetings and other Group Meetings.....	15
3.2.1 What Is Scrum?.....	15
3.2.2 The Current Project's Scrum Meetings	16
3.3 Methodology Followed for Actually Creating the Website	17
3.3.1 Set-up of Tools.....	17
3.3.2 Creation of Database.....	18
3.3.3 Creation of Website Domain and Web Pages.....	19
4. RESULTS.....	20
4.1 Website Components	20
4.1.1 Registration Page	20
4.1.2 Login Page	21
4.1.3 Dashboard	22
4.1.4 Inventory Page	23
4.1.5 Management Page.....	23
4.1.6 Calendar Page	25
5. FUTURE WORK.....	26
6. CONCLUSION.....	27
REFERENCES	28
BIOGRAPHICAL INFORMATION.....	29

LIST OF ILLUSTRATIONS

Figure		Page
2.1	timeclockwizard.com Dashboard and Side Menu	4
2.2	Website Dashboard and Top-Menu	4
2.3	MySignUp.com Sign-Up Sheet	6
2.4	Monthly Calendar View of Calendar Page	7
2.5	Weekly View of Calendar Page.....	8
2.6	Daily View of Calendar Page	9
2.7	Quartzy.com Inventory Item Interface.....	10
2.8	Website Inventory Page	10
4.1	Website Registration Page	20
4.2	Website Login Page	21
4.3	Website Dashboard	22
4.4	Website Inventory Page	23
4.5	Website Management Page.....	23
4.6	A Sample Generated Report	24
4.7	Calendar Page	25

CHAPTER 1

INTRODUCTION

1.1 Statement of the Problem

The UTA Smart Hospital is a training facility designed to train students to become professional nurses. A great deal of data must be stored in order to keep track of faculty, students, rooms, inventories, time logs, and other pieces of information. However, the Smart Hospital currently has no method of keeping this data in one place. Rather, they must rely on different files and interfaces, such as different websites and programs, in order to keep track of this information. Some of the resources that the Smart Hospital uses are: MySignUp.com to manage signup information for lab sessions, Quartzy.com for inventory information, and Microsoft Access, in order to manage their student information. The goal of this project is to create a website that allows the Smart Hospital to keep track of their different pieces of information, from inventory to events. My responsibilities were to create the files necessary for the database. I also ensured that the credentials that the user entered in the registration page were inputted into the database. I then helped create the functionality that allowed the user to supply credentials in the login page so they can access the main website.

1.1.1 Main Goals

The first and foremost goal of this project was to ensure that the website provided an interface that allows the Smart Hospital faculty and students to login and view information relevant to them. The webpage utilizes different tabs that allow the user to

view different information; for instance, one tab is used to display information about scheduled events, and another tab may be used to display information about the inventory of the hospital. The second goal was to ensure that the data is stored correctly and accurately. For example, if a building that stores a particular medicine is changed, the user would be able to see an up-to-date version of the list of medicines in this building. Upon completion of these goals, the website will then be able to store any additional information like all medicines that the Smart Hospital uses. Another key goal was to make this website accessible to registered users by creating a login page that only allows users to login with their registered email address and their created password. By completing these goals, at least a framework of the website can be created, one that can allow new features to be added.

CHAPTER 2

LITERATURE REVIEW

2.1 A Study of Existing Resources Used in This Project

As mentioned earlier, the Smart Hospital has used existing resources in order to carry out their tasks. There are other existing websites and software applications that have had similar goals. Design aspects were drawn from these websites. This section will present examples of these websites that were drawn upon, both from an internal system perspective, and from an external, aesthetic view.

2.1.1 timeclockwizard.com

The Smart Hospital currently uses www.timeclockwizard.com to allow students to gain lab hours that they must accumulate in order to reach a certain number of required lab hours. While the specific feature of clocking in and clocking out has just started to be implemented on the website, two other important features were based heavily on the timeclockwizard.com interface. One of these features was the use of a dashboard. The dashboard would show key upcoming events and alerts.

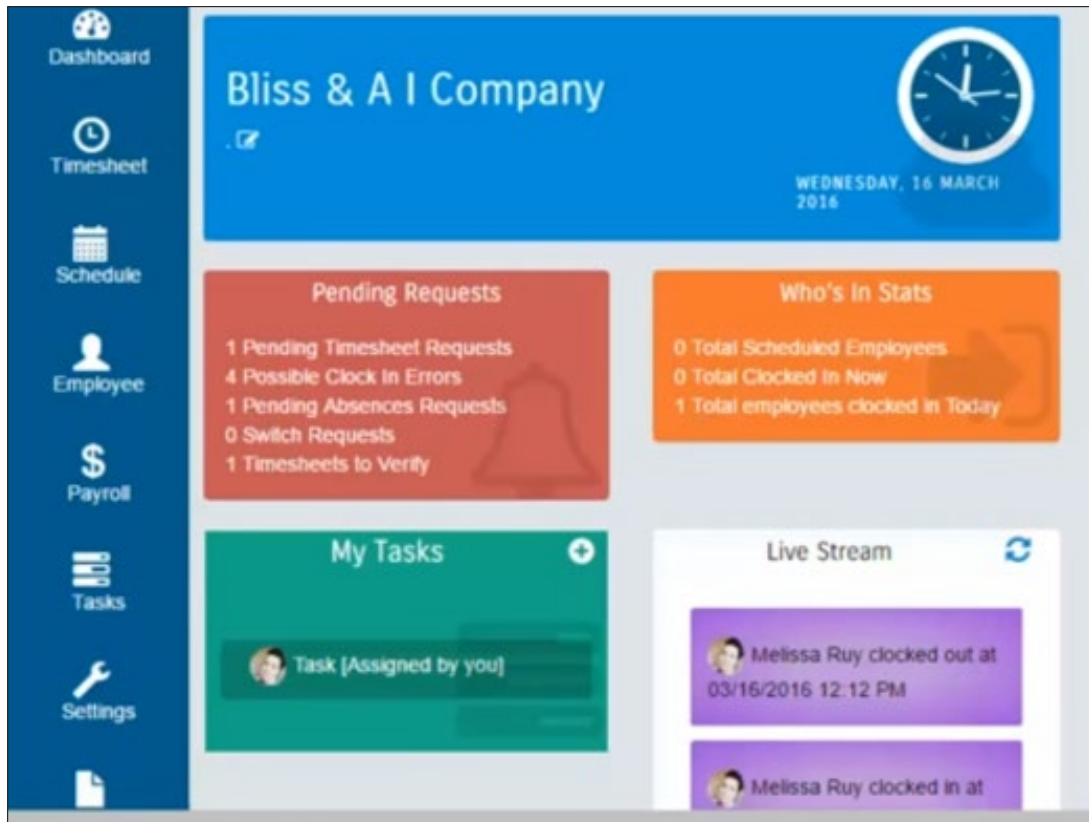


Figure 2.1: timeclockwizard.com Dashboard and Side Menu



Figure 2.2: Website Dashboard and Top-Menu

As shown in Figure 2.1, the dashboard used in the main page of timeclockwizard.com contains important tasks that need to be done by a person and requests to the manager made by an employee or someone else. The dashboard created for this project, as shown in Figure 2.2, was a little different, as it was intended to display important information relevant to the students and employees working in the Smart

Hospital, such as upcoming events like a lab session. This way, users can see information that is pertinent to them at the time. Another design aspect that was derived from the timeclockwizard website was the side menu, which contains links, shown as icons, to other web pages that are important to the user. This concept was used in this project's dashboard in the form of showing all the links to the other pages as tabs, the only difference being that a top menu was used rather than a side menu.

2.1.2 MySignUp.com

One of the requirements that the Smart Hospital had for this project was the need to manage events. Currently, the Smart Hospital allows students to sign up for events using MySignUp.com, which managed the events in a list-like fashion. While MySignUp.com did provide a view that listed events by time, the website provided a weekly, monthly, and daily view that allowed users to drag and drop events. Users can easily create an event, and on the weekly and daily view, adjust what specific time on that day the event will occur, as well as how long it will take.

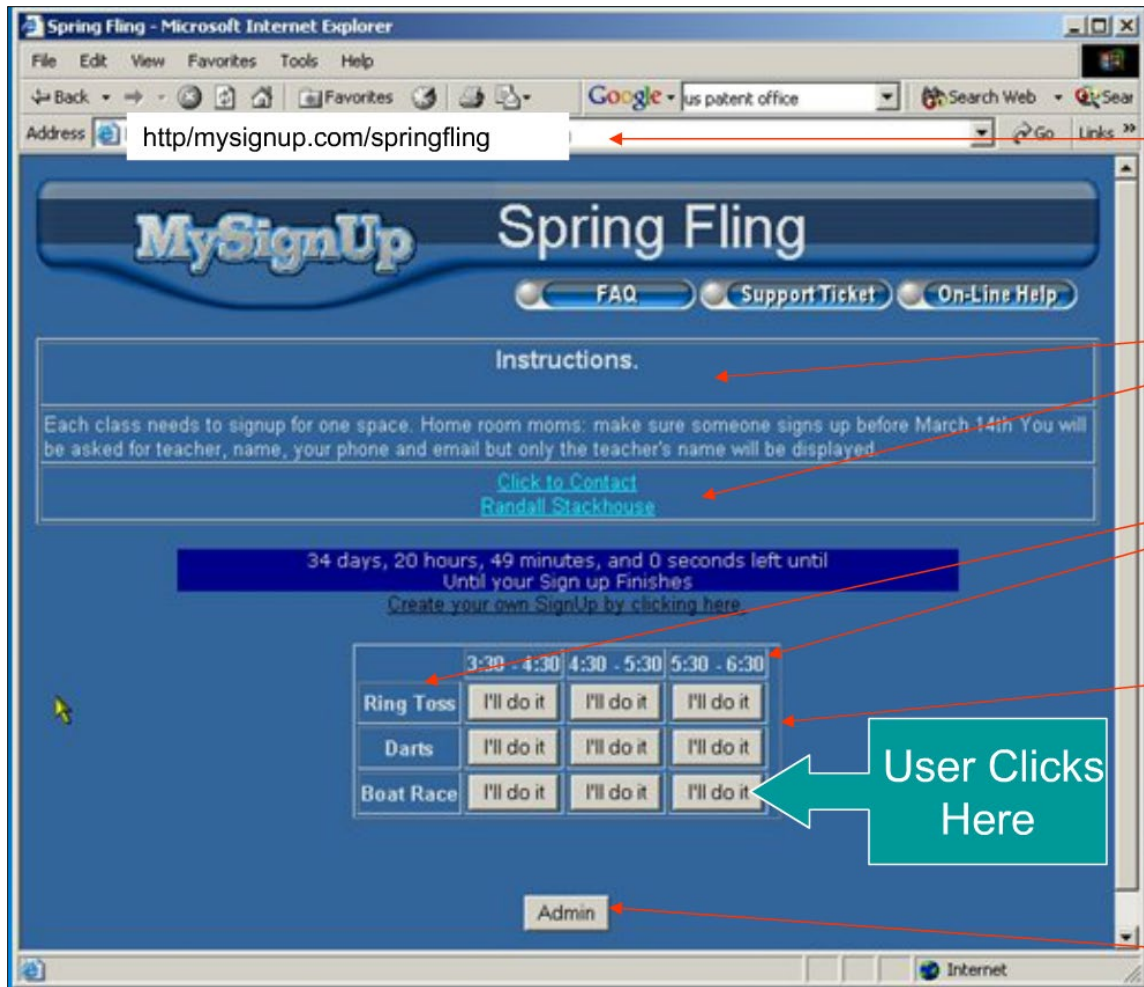


Figure 2.3: MySignUp.com Sign-Up Sheet

<input style="border: none;" type="button" value=" < "/> <input style="border: none;" type="button" value=" > "/> <input style="border: none;" type="button" value=" today "/>		January 2017						<input checked="" style="border: none;" type="button" value=" month "/> <input style="border: none;" type="button" value=" week "/> <input style="border: none;" type="button" value=" day "/>		
Sun	Mon	Tue	Wed	Thu	Fri	Sat				
1	2	3	4	5	6					
8	9	10	11	12	13					
15	16	17	18	19	20					
22	23	24	25	26	27					
29	30	31	1	2	3					
5	6	7	8	9	10					

Figure 2.4: Monthly Calendar View of Calendar Page

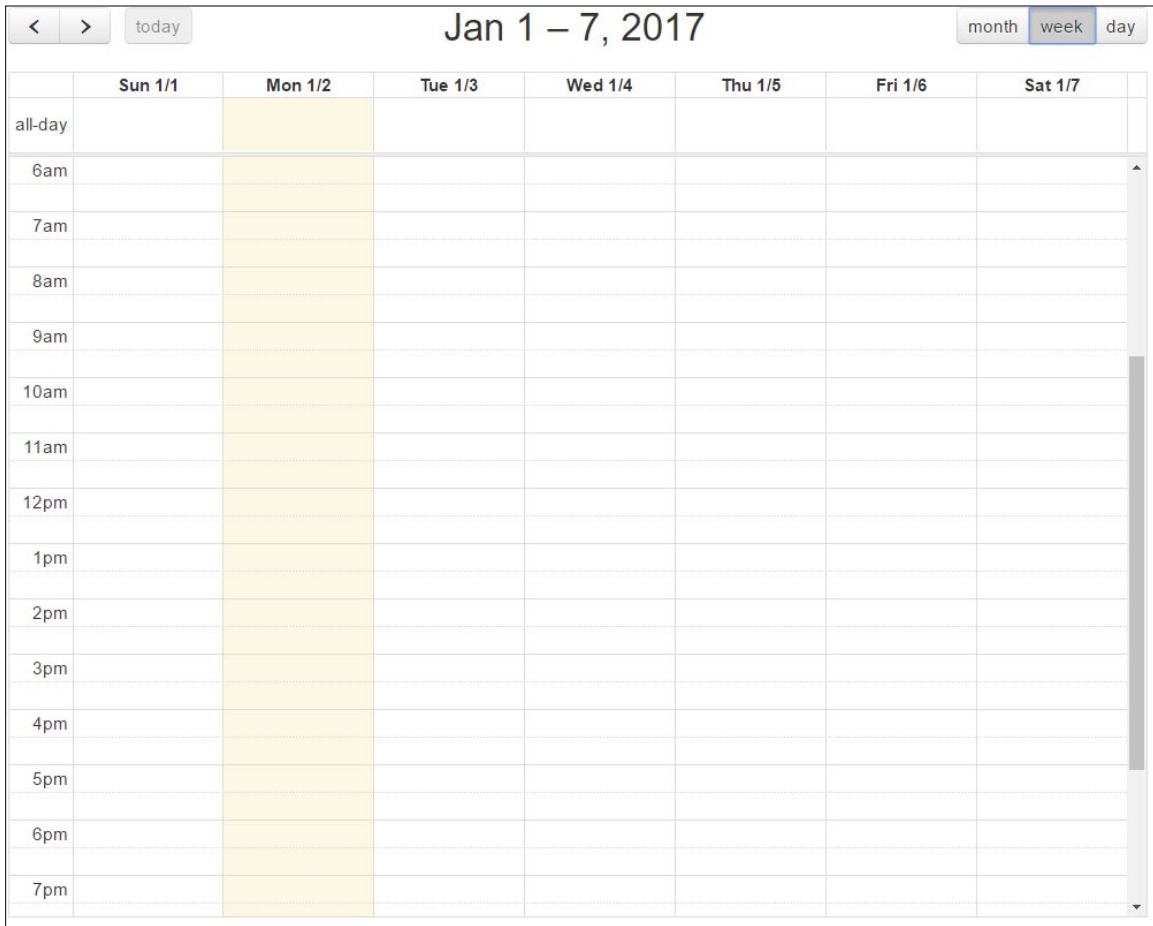


Figure 2.5: Weekly View of Calendar Page

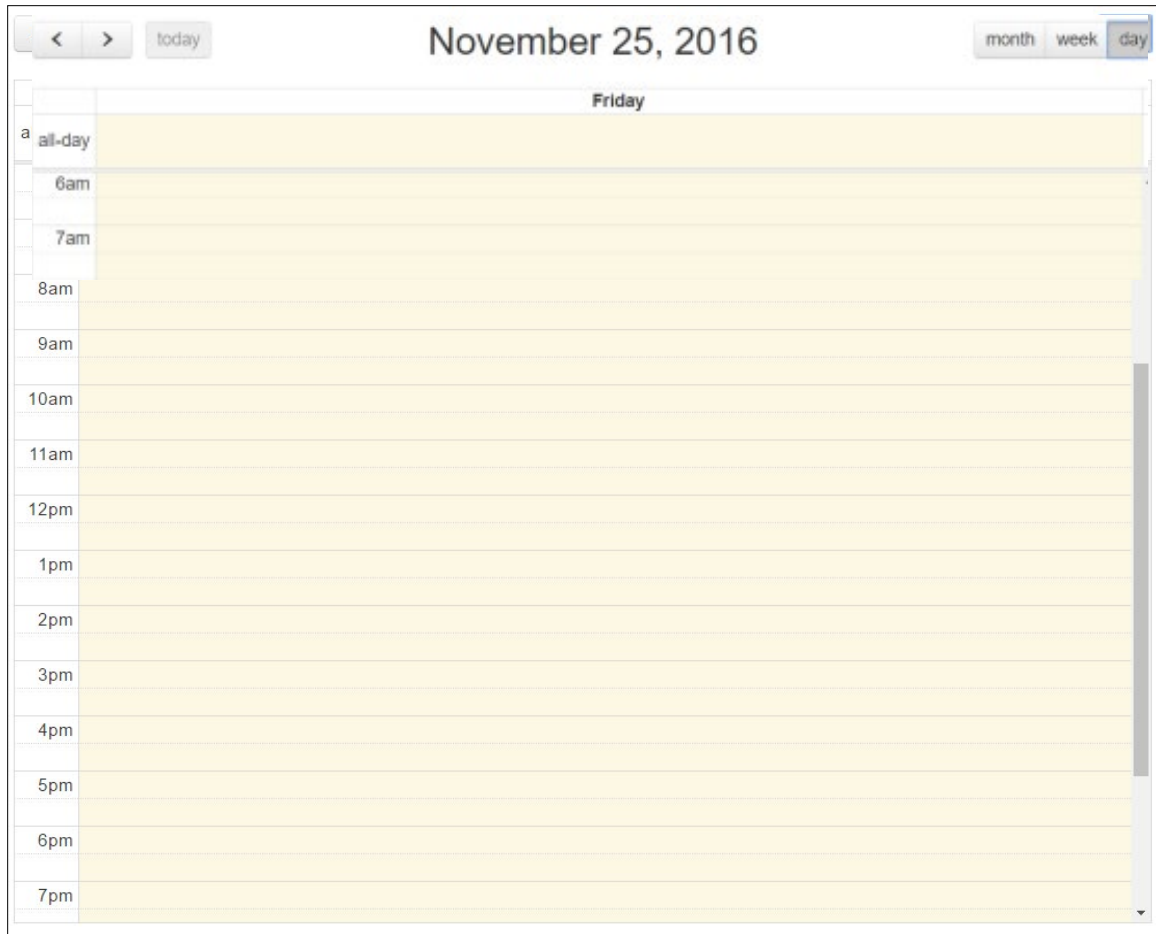


Figure 2.6: Daily View of Calendar Page

2.1.3 *Quartzy.com*

Lab inventory can be difficult to manage manually due to the huge number and variety of items. Quartzy.com is used by the Smart Hospital to help them accomplish this task. One of the features that Quartzy.com provides is to allow users to add inventory items that can then be stored and displayed in a list view. It also allows inventory to be searched, either by its name, vendor, owner, and so on. The item can be edited and its quantity can be updated whenever an item is checked in or checked out.

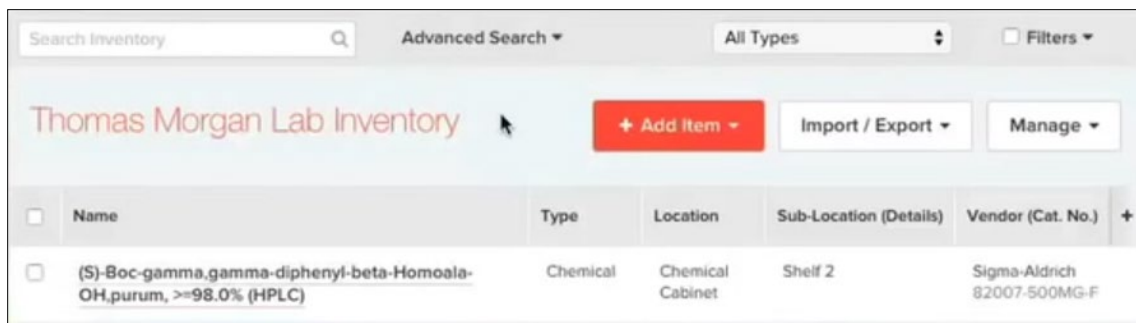


Figure 2.7: Quartzzy.com Inventory Item Interface

The current project's inventory page is similar in design to the quartzzy.com page, which displays all of the items in a list view, and allows the user to edit the fields of the item, including the quantity of that item. It also allows the user to delete items from the list. For example, if the item is not going to be in the lab anymore, the user can choose to delete the item completely from the database. Search functionality is also implemented, allowing the user to search for a particular item.

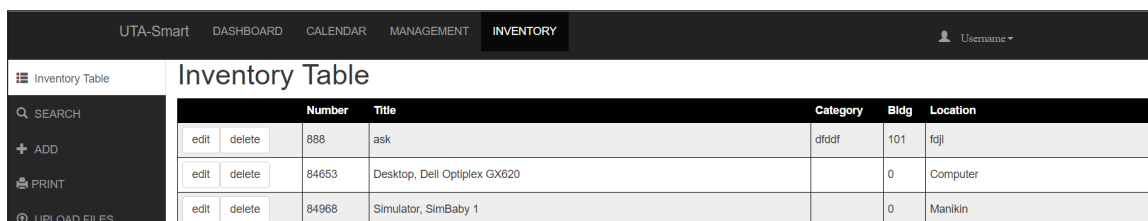


Figure 2.8: Website Inventory Page

2.2 Other Computer Applications and Web Interfaces Built for Hospitals

2.2.1 Hospital Patient Database System Concept by S.A. Asabe, et. al (2013)

This project proposed a system that focused on an important concept used in the current project, which was storing information in a database. The authors of this paper used Microsoft Visual Studio for the programming end, and Microsoft Office Access for the database end, which were not used in the current project. In conclusion the authors of the project found that their database provided a way for the hospital faculty to manage their information effectively, while providing adequate space for documents that needed to be physically filed due to security concerns. The hospital agreed with the researchers' hypothesis, that it would be much more effective to have an electronic system that managed their files [1].

2.2.2 The Intelligent Agents in Hospital Appointment Scheduling System by A. Hylton III and S. Sankaranarayanan (2012)

This is an Android application, called the "Intelligent Agent Based Scheduling System," which is created to solve the problem of patients scheduling appointments with doctors. The paper tries to address the problem of patients needing to make an appointment with a doctor, yet not being able to find a close enough appointment time that they prefer, or needing to see a doctor immediately. This application implements a priority system and checks databases for available times of doctors. Registered patients have higher priority than unregistered patients, and senior citizens, less physically able people, infants, pregnant women, or children are given the highest priority, whether they are registered or unregistered. The next available slot is assigned to the patient with the highest priority, compared to other patients. For example, if a registered senior citizen tries to take an appointment with the doctor at a certain time and if a registered medical staff tries to take

an appointment with the same doctor at that same time, the system will give preference to the senior citizen and allow them to register for that time, and have the staff to take the next available appointment time [2].

2.2.3 A Dynamic Platform for Workflow Management using Web Services: a Hospital Scenario, by N. Prinyapol, et. al (2009)

This is a web platform that was developed for the sake of assigning medical nurses to specific patients they must see on their shift and what they must do with these patients. Normally, these assignments are made face-to-face and through papers, involving doctors writing orders for patients and giving these orders to an in-charge nurse, who then organizes a nursing care schedule to assign nurses to patients to record vital signs. Nurses must then switch out with other nurses when their shift is done. This process requires that the information needed to take care of the patients under the previous nurse's care be passed along to the nurse on the next shift who is assigned to take care of those same patients. This can cause problems, since reading the notes that are being passed along can cause reading interpretation errors, and so delays can be created since these interpretations must be checked with the doctor. The proposed web management system will try to solve these problems by first matching the required tasks that must be done to the nurses who prefer performing these tasks. These matches are then cross checked with the nurse's "work profile service," which lists the nurse's general responsibilities and tasks, set by the supervisors or managers in order to ensure that these tasks are appropriately fitted to the best nurse. A schedule is then created, following constraints placed on it by supervisors or managers. This schedule tries to prevent illogical scheduling such as having random gaps in a nurse's work schedule, or assigning tasks to a nurse that managers or supervisors believe are best fitted for that nurse. This platform was created using a database and the

XML and HTML languages. It was not tested yet, however; the authors hope to test it at a certain hospital [3].

CHAPTER 3

METHODOLOGY

The Scrum methodology model was used as the development model to complete the project. The sections below outline the methodology in detail, as well as challenges that were faced in the duration of this project.

3.1 Requirements Gathering

The first step in Scrum methodology is to gather some initial requirements from the customer. These requirements will be used to create a demo of the product. The customer can preview this demo, and give feedback on the features and functionality that he likes or dislikes, or whether additional features should be implemented. Some of the features that the customer for this project, Mr. Soohyun Kim, the Simulation Manager of the Smart Hospital, wanted from the system included:

- A way to schedule workstations for students on certain days and times
- A way for faculty and students to register for these events, and upon doing so, send a confirmation to the faculty/students via email indicating that they have signed up
- Store inventory items in a database and load the inventory items and other information about them (such as that item's quantity)
- A way to scan tags on items using the Smart Hospital barcode scanners and have it automatically inserted in the database

- A notification system that notifies the faculty of when a certain category of items is below a certain level, so they can restock
- Automatically keeps track of the hours that students and employees have worked on and generates a report based on this information

A System Requirements Document was created, which allowed these requirements, along with their priority and source, to be listed. This helped prioritize which requirements needed to be worked on first. The System Requirements Document also made it necessary to explicitly list other requirements that needed to be fulfilled in order to complete the customer's requirements; for instance, since there was a requirement to store information that can be specific to faculty and students, such as events, a requirement of having website accounts for each user was listed. Upon completion of the System Requirements Specification Document, the participation in Scrum meetings started.

3.2 Participating in Scrum Meetings and other Group Meetings

3.2.1 What Is Scrum?

Scrum is the development framework that is used by companies whose goal is to allow teams to deliver the highest valued product to customers in a short amount of time. One of the key points in Scrum is that teams try to get a working piece or pieces of the project every two weeks (called "sprints" in Scrum), and from there, try to improve upon these pieces and/or add new pieces to the project. These pieces are presented to the customer, so that the customer can tell the developers what they like or dislike about the prototype and what else would be useful to the customer to add on to the prototype. Scrum meetings are held by the team daily, and are short, typically 15 minutes. In the first Scrum

meeting of a sprint, the goals are established for that sprint. Meetings after this initial meeting for the rest of the sprint require that each team member answers three questions:

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

This is so that the members are setting commitments for themselves. At the end of each sprint, the team performs a "retrospective," where the team looks at what they have accomplished well and what they need to improve on, so the team can perform better in the next sprint.

3.2.2 The Current Project's Scrum Meetings

The team followed the Scrum methodology, but there were significant differences that the team members' schedules required. The team came up with sprint goals in short meetings after class or in a team meeting. The team would then present these goals to the rest of the class and the professor; this helped since the team was able to express any barriers and/or difficulties that they were facing to the professor, who would then give suggestions or would help resolve an issue that may come up in completing these goals. One of the initial goals that the team came up with was deciding what tools would be used for the project. Team members were assigned roles in the project, and later on, work on the actual project was done together, unlike the traditional Scrum meetings.

3.3 Methodology Followed for Actually Creating the Website

3.3.1 Set-up of Tools

The most difficult part of the project was deciding what tools were needed to create the website. The fact that a database was required for a website also created a difficulty, since the team initially did not know how to connect database information with website pages. So, a tool called node.js was suggested and researched, which was a tool that was meant to help connect servers with the application, i.e. the website interface. The team later realized that PHP could be used, since it was a programming language that can be used to directly create connections between the website files and the database. This made learning node.js no longer necessary.

Some of the team members took a database class, and used SQL and SQL Workbench to create databases. So these tools were used in this project to create the database files.

Information being stored on a website typically requires a paid service. Initially, an Amazon Web Services account was created by the customer, at the team's request, to host the website. Yet the team later realized that they could not use the budget given for this project to pay for these services, due to two reasons: 1) the CSE Department did not own the services and 2) Amazon charged for these services based on the number of visitors to the website, rather than on a fixed yearly or monthly basis, which was required since the services would need be terminated by the department after the project was done. The professor therefore set up an account on ASPHostPortal.com, which charged for its services annually. This service not only allowed the team to host an actual website that is publicly available, but also allowed the team to upload their existing SQL database files

and provided a way to create the files needed for the actual web pages, such as HTML files, and so on.

3.3.2 Creation of Database

In order to create the database, the team needed first to create a schema. A database schema shows the relationships between different entities in the database and their attributes; for instance, the equipment in the Smart Hospital may be one entity in the database, and it may have certain fields of information such as its inventory number, the number of the building it is located in and so on. The buildings that the Smart Hospital needs to keep track of may also have fields of information, and some of these fields may be the same as the equipment component, such as the building number, and this is where a relationship is formed, since these fields are common. This building number may be made a primary key and, at the same time, a foreign key. A primary key is used to identify a component in a database since that component has a unique field. The foreign key is used to link two components together by linking the two common fields. Making the building number a primary key in the building component ensures that each building entry in the database is unique, since no two buildings can have the same building number. By making this field a foreign key in the equipment table at the same time, a relationship can be formed between the equipment components in the database and the buildings components in the database. This way, if one wants to get the building number that an equipment was located in, that person is able to find the building number by looking up the building number stored in the equipment table. If this building number was updated, it would be updated in both the equipment fields and the building fields. After creating the schema, the actual SQL files

that implemented the schema were created and edited. The first version of our database was ready for use in the actual website.

3.3.3 Creation of Website Domain and Web Pages

After obtaining a website domain the team was able to start actual work on the website. The general layouts of the web pages were designed, and specific website pages and features were worked on. The registration and login pages were made. The framework for the other web pages were made, such the calendar page mentioned earlier, the inventory page, and a management page that allows the manager to manage other users, by deactivating or activating their account.

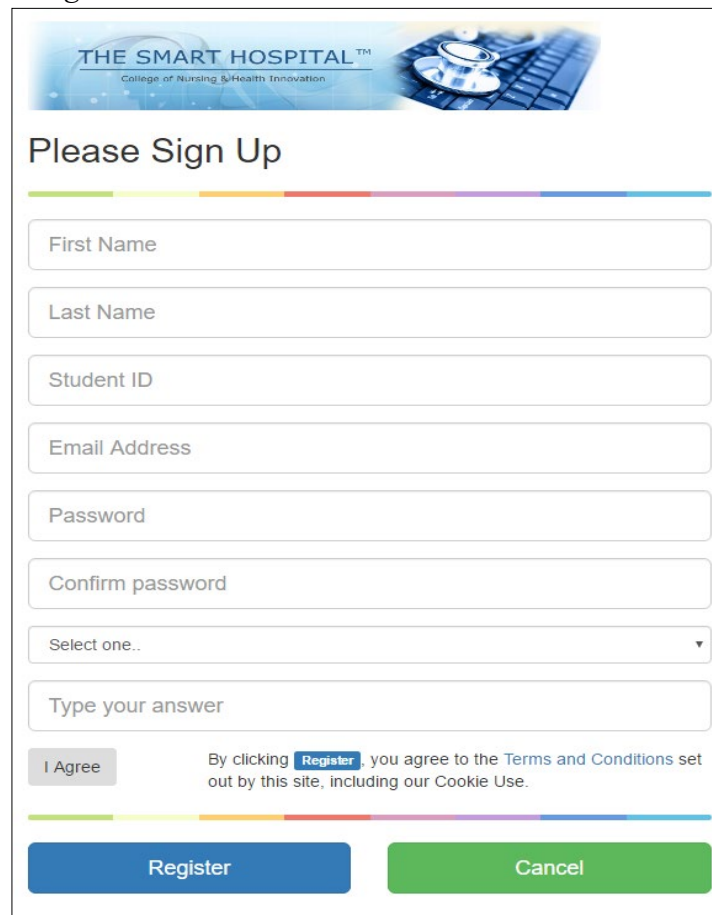
CHAPTER 4

RESULTS

The team was able to achieve the goal of demonstrating that it was possible to accomplish what the Smart Hospital was asking for. Using HTML, Bootstrap, CSS, PHP, and SQL, the team was able to create a website that proved that managing inventory, events, and users can be done from a single web platform.

4.1 Website Components

4.1.1 Registration Page



The screenshot shows a registration form for 'THE SMART HOSPITAL™ College of Nursing & Health Innovation'. The form is titled 'Please Sign Up' and features a colorful horizontal bar. The fields include: First Name, Last Name, Student ID, Email Address, Password, Confirm password, a dropdown menu labeled 'Select one..', and a text input labeled 'Type your answer'. Below the fields is a checkbox labeled 'I Agree' and a line of text: 'By clicking [Register](#), you agree to the Terms and Conditions set out by this site, including our Cookie Use.' At the bottom, there are two buttons: a blue 'Register' button and a green 'Cancel' button.

Figure 4.1: Website Registration Page

New users must first be registered in the system before they can log in to view the main website pages. All fields on the project's registration page are required, as well as the user clicking on the "I agree" button, and if the user tries to leave a field blank and clicks "Register," then a pop-up will appear, asking the user to fill out the field. The data in the fields, as well as the selected security question, is stored in the database upon completing the form and clicking "I Agree," then "Register."

4.1.2 Login Page


The image shows a web browser window displaying the login page for 'THE SMART HOSPITAL'. The header features the hospital's logo and a stethoscope on a keyboard. Below the header, the text 'Please Sign In' is displayed. There are two input fields: 'Email Address' and 'Password'. Below these fields are two buttons: a green 'Sign In' button and a blue 'Register' button with a dropdown arrow. A link for 'Forgot Password?' is located at the bottom right of the form area.

Figure 4.2: Website Login Page

Upon completing registration, the user can login using their email address and password. The data in these fields are checked against existing email addresses and passwords for the users stored in the database. If the user's email address and password stored in the database match the email and password supplied in the fields, then the user can login; otherwise, the page is refreshed and the user must either retype their email or password or click on "Register" to create a new account. Due to time limitations, the team

is still working on getting a functioning feature that allows the user to click on the "Forgot Password?" button so that the user can enter in their email to get either a new password or their current password sent to them so they could login.

4.1.3 Dashboard

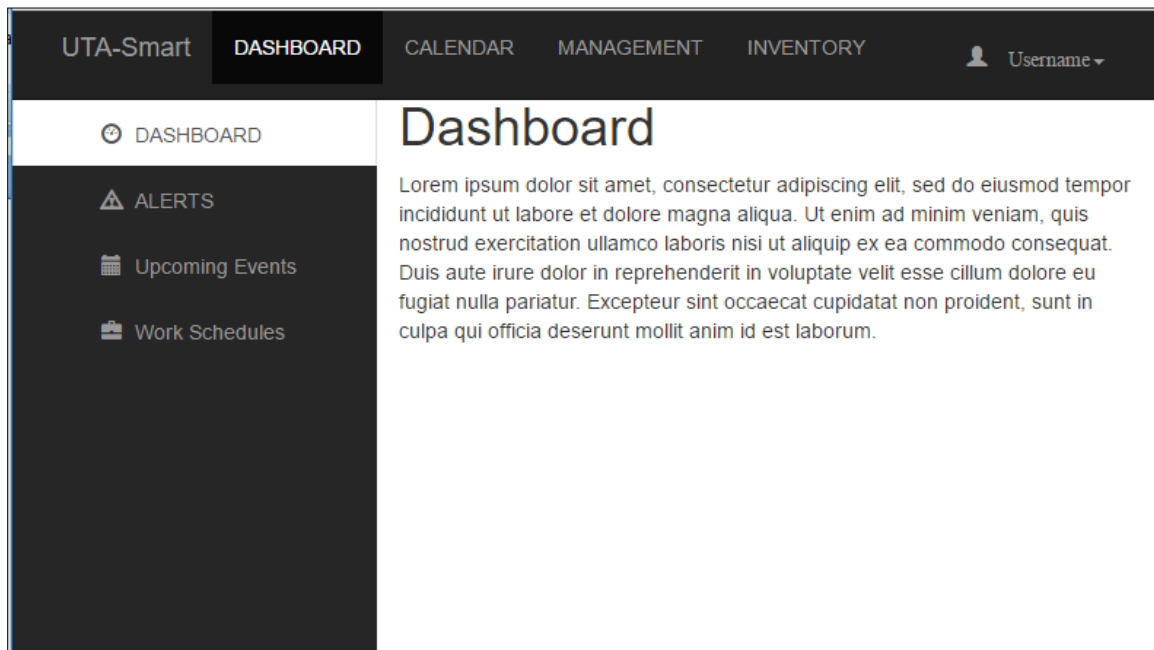


Figure 4.3: Website Dashboard

This page is intended to display relevant information for the students and staff, such as upcoming events. However, the team has not been able to create additional functionality on the dashboard, such as displaying these relevant events and alerts to the user or allowing the user to create work schedules. They focused more on the design of other pages and features, such as the management and inventory page, after creating the basic design for the dashboard, since they felt that the other features, such as inputting inventory items, were of a higher priority.

4.1.4 Inventory Page

	Number	Title	Category	Bldg	Location
edit delete	888	ask	dfddf	101	fdjl
edit delete	84653	Desktop, Dell Optiplex GX620		0	Computer
edit delete	84968	Simulator, SimBaby 1		0	Manikin

Figure 4.4: Website Inventory Page

The Inventory page allows the user to input lab items along with other information about that item, such as their item number, what category they are in, the building that the items are located in, and the location of these items. A user can also add inventory items using the barcode scanner by clicking on the appropriate fields in the "ADD" page and scanning the item barcode.

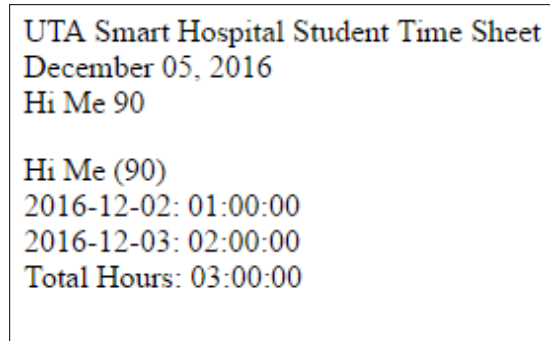
4.1.5 Management Page

	ID	First Name	Last Name	Status
edit activate deactivate	90	Hi	Me	Student
edit activate deactivate	123456789	Uta	Smart	Student

Figure 4.5: Website Management Page

Currently, the design of the page is done, as well as the activating and deactivating of users; if a user is deactivated, then the user cannot login until the admin clicks on the activate button, allowing them to login again. The edit button currently does not work. The

Generate Report currently works, allowing a user to generate a report that shows the total number of hours that they have spent in the lab (signed into the website as of now), similar to this:

A rectangular box containing a sample generated report. The text inside the box is as follows:

UTA Smart Hospital Student Time Sheet
December 05, 2016
Hi Me 90

Hi Me (90)
2016-12-02: 01:00:00
2016-12-03: 02:00:00
Total Hours: 03:00:00

Figure 4.6: Sample Generated Report

4.1.6 Calendar Page

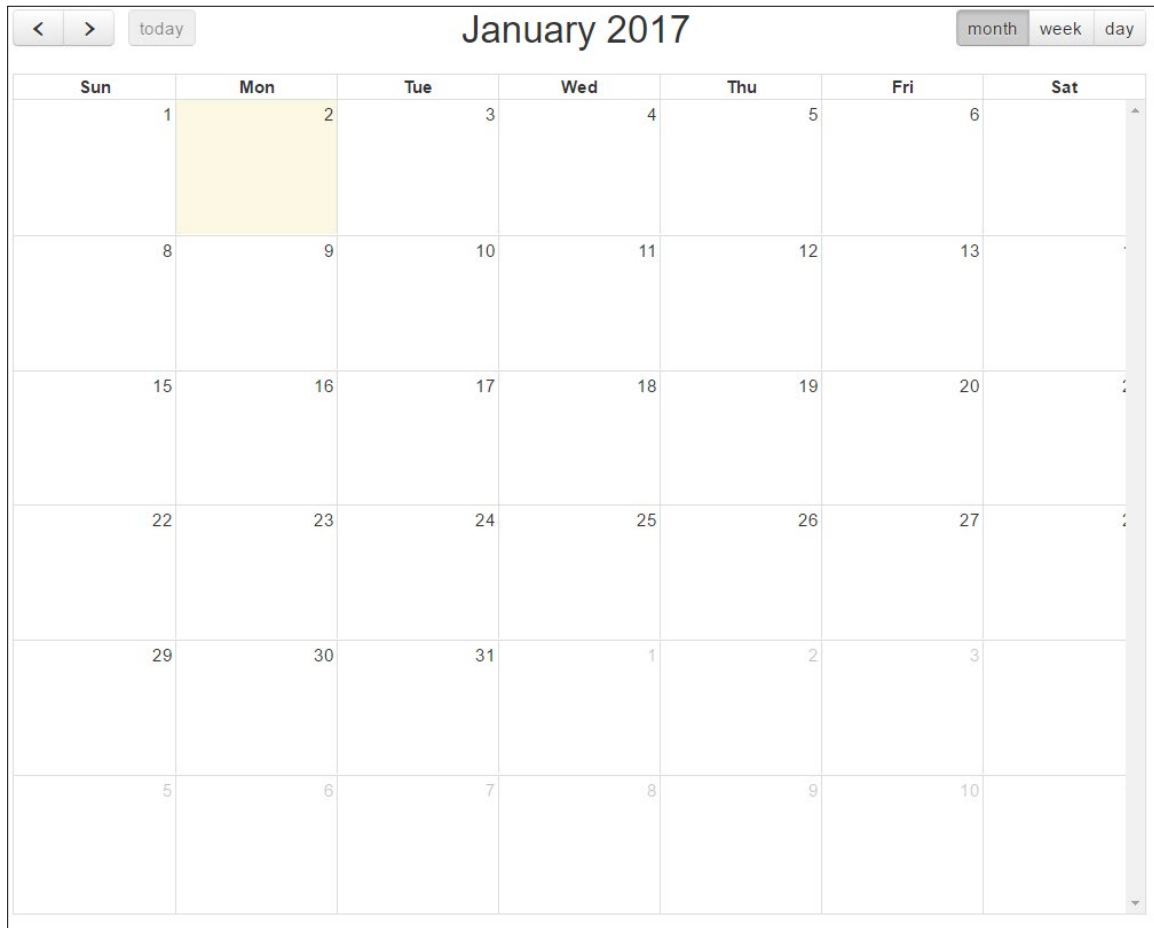


Figure 4.7: Calendar Page

This project's calendar page, as shown in figure 4.7, allows a user to view a calendar in a monthly, weekly, and daily view. The user can also drag and drop events such as a meeting, conference, or lunch event.

CHAPTER 5

FUTURE WORK

Our website has a lot of features, yet there are many other features that need to be implemented that could further help the Smart Hospital, as well as any other healthcare facility that may need a similar framework to help their management. Either a future Senior Design team or other future developers can potentially use our framework and improve upon it in order to create a fully functioning and truly useful website. Some example functions that may need improvement or may need to be included are:

- A notification system that indicates to the faculty when a certain category of item is below a certain level, so they can restock
- Storing events on the calendar that shows students the days and times that a student will use a particular workstation
- Having the edit button work on the management page
- Allowing the user to upload excel sheets containing inventory item data on the inventory page so that this data gets imported into the database
- Allowing the user to print the items on the inventory page

CHAPTER 6

CONCLUSION

This project has shown with the above results that it is possible to combine all of the Smart Hospital's information into one web interface, which was the main goal that the Smart Hospital wanted to see if the team could accomplish in this project. Other features that they asked that were not yet implemented, such as the need for students to register for events, or notifications for low supply stock for particular items, are possible, however with a lot of research and time. The team realized in doing this project that they had to spend most of their time learning what they did not know or solving problems that were difficult for them to solve due to their lack of experience in web development. However, they did get a framework built, and thus it is possible, with more time or experience to deliver a fully functioning platform that would completely serve the Smart Hospital's needs.

REFERENCES

- [1] S. A. Asabe., N. D. Oye, and G. Monday, "Hospital Patient Database Management System A Case Study of General Hospital North-Bank Makurdi-Nigeria." Makurdi, Nigeria: Compusoft, 2013.
- [2] A. Hylton III and S. Sankaranarayanan, "Application of intelligent agents in hospital appointment scheduling system." *International Journal of Computer Theory and Engineering* 4(4), pp. 625. 2012. DOI: 10.7763/IJCTE.2012.V4.545.
- [3] N. Priyanpol, J. Fan, and S. K. Lau, "A Dynamic Platform for Workflow Management using Web Services: A Hospital Scenario," *International Conference on Internet Computing and Web Services (ICICWS'09)*. Hong Kong, China: International Association of Engineers, 2009, pp. 944-949.

BIOGRAPHICAL INFORMATION

Neelim Haider graduated from UT Arlington with an Honors Bachelor of Science in Computer Engineering. He has worked on computer programs, including operating system level programs and embedded level programs, as well as embedded microcontroller alarm systems. His goal is to work in a career where he can apply his computer engineering skills to help people in their everyday lives.