

UNIVERSITY OF ALASKA ANCHORAGE

CSCE A470

CAPSTONE PROJECT

Database and Backend for Summer Engineering Academies Registration System

Author:

Zakary Stone

Supervisor:

Prof. Scott Hamel, PhD

Anchorage AK, May 2016



**Computer Science &
Engineering Department**
UNIVERSITY *of* ALASKA ANCHORAGE

© Copyright 2016
by
Zakary Stone

zakarystone@gmail.com

Version 1.0

II

Abstract

The capstone project presented in this document is for the development of a web based summer camp registration system. This project is created for the University of Alaska Anchorage (UAA) Summer Engineering Academies (SEA). The application is meant to be a tool for parents, instructor, staff, and administrator of the SEA program. This application will make it possible for parents to sign up and register their children for SEA sessions. The application will also allow an administrator to manage and process applications for SEA sessions. This project is being designed so it can grow and possibly supply other forms of support to the users.

The development of this project is divided into two different teams. The front end of this project which will be the user experience and user interface. The backend of the project will be the business logic and database. This report documents the projects backend work. The server will be hosted by UAA Information Technology Services (ITS).

ACKNOWLEDGMENTS

This capstone report is being submitted by myself, but the whole project could not have been completed without the help of others. I would like the people in my life who made it possible for me to complete this project.

My first thanks would go out to my mom, Melanie Stone. She has been there supporting me in all endeavors. She also raised me with a love mathematics and logic which lead here today.

On this project, I would also like to thank Dr. Scott Hamel. Without his encouragement I wouldn't have been working on this project.

I would also like to thank the members of the Summer Engineering Academies Registration System student team. The team members, Christopher Brooks, and Johannes Meyer have been a great help and an inspiration.

I would also like to thank BP for their generous donations to the Summer Engineering Academies. This project could not happen without them

Table of Contents

CHAPTER 1 - INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 APPLICATION	2
1.3 MOTIVATION	4
1.4 RECENT DEVELOPMENTS	5
CHAPTER 2 - SYSTEM INTEGRATION AND MODELING	7
2.1 INTRODUCTION	7
2.2 TECHNOLOGY	7
2.3 SYSTEM DESIGN	10
2.4 COMPONENTS, DEVELOPMENT TECHNIQUES, AND TOOLS	11
2.5 DEVELOPMENT METHODOLOGY	12
CHAPTER 3 - DESIGN AND TESTING	14
3.1 INTRODUCTION	14
3.2 USER INTERFACE	14
3.3 TESTING METHODOLOGY	16
3.4 SOFTWARE DEVELOPMENT TOOLS AND TECHNIQUES	17
3.5 AGILE CODING	18
CHAPTER 4 - USER MANUAL	20
4.1 INTRODUCTION	20
4.2 SOFTWARE REQUIREMENTS	21
4.3 GENERAL LAYOUT	21
4.4 MENUS AND NAVIGATION	22
4.5 FUNCTIONS	23
CHAPTER 5 - SUMMARY	37
5.1 INTRODUCTION	37
5.2 IMPLICATIONS	38
5.3 FUTURE DEVELOPMENT	38
	39
BIBLIOGRAPHY	39

List of Figures

1.1	Students in the UAA summer engineering academy robotics session	2
1.2	A typical collaboration of the MVC components in a web application	3
1.3	HUGE is a simple PHP framework that provides an MVC structure and user authentication.	4
1.4	A portion of the Qualtrics survey that parents fill out for their application.	5
1.5	Development tools being use for the project.	6
2.1	Composer is used to manage all of the dependencies within the SEA Registration System.	8
2.2	Diagram describing how we use git for source control.	9
2.3	Database tables that contain the most important information.	10
2.4	The Database Design of the SEA Registration System.	11
2.5	A diagram describing the modified waterfall development cycles.	13
3.1	The navigation bar contains the most common actions of the application	15
3.2	Administrative filtering of participants	16
3.3	Partial user acceptance test form. Final will include all tasks.	17
3.4	Debugging in PhpStorm	18
3.5	The modified waterfall agile development process we've taken	19
5.1	The site is live!	38

Chapter 1

Introduction

1.1 Introduction

The UAA Summer Engineering Academies (SEA) offer a series of week-long hands-on instructional sessions for students entering grades 6 through 12. Each academy is designed to encourage interest and provide exposure to science, technology, engineering, and mathematics (STEM). The academies provide the sessions, including robotics (Figure 1.1) and creative coding, at low-costs to be available to students who typically have limited opportunities for these experiences.

The registration process to sign up for the academies is currently completed through multiple third party services including a survey tool and web store. The information is then collated in an excel spreadsheet and then managed using visual basic scripts. With the growing popularity of these summer academies, SEA is in need of a unified information management system to simplify the registration process and administration tasks.

For this capstone project, we have assembled a team of students with complementary skills and availability to complete it. Our goal as a team is to gain practical experience in each phase of the systems development lifecycle, to develop skills relevant to professions in our chosen fields of study, and to grow as UAA students and Information Technology enthusiasts.



Figure 1.1: Students in the UAA summer engineering academy robotics session. [1]

1.2 Application

For this project, we will be designing and implementing a new registration system for the UAA Summer Engineering Academies. This will be designed based on requirements and needs set forth by the programs director. The new SEA Registration System will improve the registration process by allowing clients, staff, and administrators to access and modify pertinent data from an Internet-based application. The application will include a database backend to contain data in an easy to access, organized way.

As a team we plan to develop and create many deliverables for the new registration system. First of which is the web application itself along with the database to support it. The second deliverable is a User's manual, while most operation should be simple there may be some more complicated operation that require explanation, such as installation and some advances administrative operations. The last is a developer's manual, which will describe the system as a whole and how one should go about making changes, for when improvement or modifications are needed in the future.

The system will be developed using free server tools that many UAA CSCE students have been exposed in some of the degree required courses. These are PHP for the application server and MYSQL for the database server. PHP and MYSQL are used in many open source because there are free tools, resources are readily and easily found online, and support can be found in numerous online communities.

This project will be developed using a Model-View-Controller (MVC) (Figure 1.2) software architectural pattern. Developing a system in an MVC pattern provides many benefits including better code and file organization, easier implementation of security, and easier separation of tasks for group development. [2] For this project, in order implement the MVC architecture pattern, we have chosen to utilize the HUGE framework by panique on GitHub (Figure 1.3). We are using this over other more popular frameworks because it is pure PHP, so using it doesn't require learning a special language.

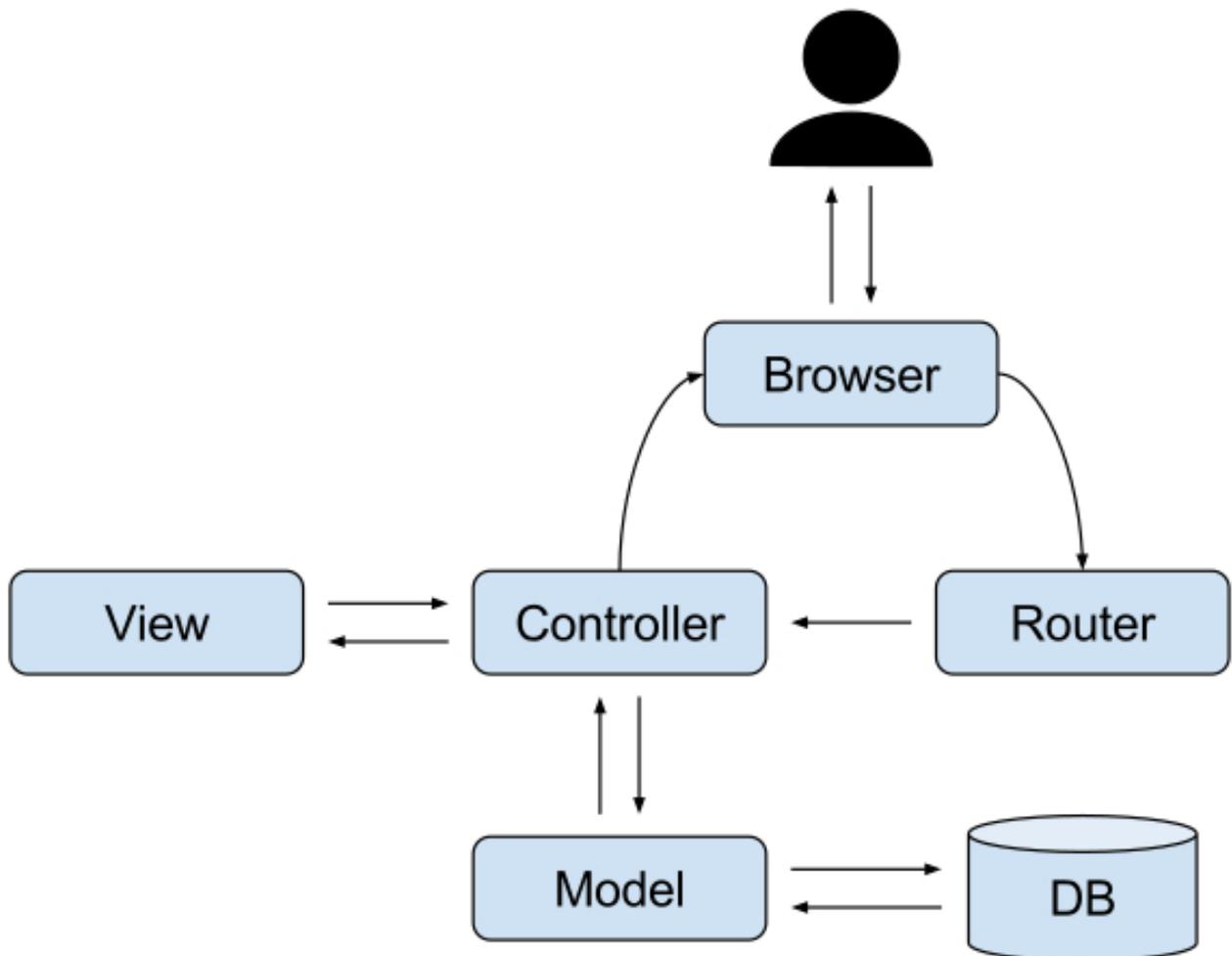


Figure 1.2: A typical collaboration of the MVC components in a web application. [3]



Figure 1.3: HUGE is a simple PHP framework that provides an MVC structure and user authentication. [4]

1.3 Motivation

Registration for SEA sessions is handled using multiple third party services and managed using an excel spreadsheet. Currently parents register participants and session choices through a Qualtrics survey. (Figure 1.4) Administrators then have to download the data from Qualtrics on a daily basis and merge the information into a series of excel spreadsheets. Then when it's time to for participants to be selected for camps, complicated Visual Basic scripts are used to perform the assignment, or “lotter”. While this method works, it can be prone to error and time consuming for administrators who need to make changes to the data. Additionally, parents often become confused about the process.

SEA is in need of a unified information management system to simplify and streamline the registration process and administration tasks. The major problems with the current system are:

1. Parents cannot modify or review applications after submission.

2. Changes to an application must be made manually by administrators.
3. In order to view the latest applications in the system administrators must download a new set of raw survey data from Qualtrics to merge into excel.



Session Preferences

Please rank your preference for the academy sessions.

Grades 7-8

	Sessions									
	EG-M1 - Alternative Energy (June 6- 10)	RB-M1 - Robotics (June 6- 10)	RB-M2 - Robotics (June 13-17)	WA-M3 - Wing Aerodynamics (June 20-24)	WA-M4 - Wing Aerodynamics (June 27-July 1)	CC-M5 - Creative Coding (July 11- 15)	RB-M6 - Robotics (July 18-22) **Located in MatSu**	SD-M7 - Structure Destruction (July 25-29) **Located in MatSu**	BB-M8 - Breaking Buildings (August 1-5)	GA-M8 - - Gaming (August 1-5)
1st Choice	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2nd Choice	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3rd Choice	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4th Choice	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 1.4: Here is a portion of the Qualtrics survey that parents fill out for their application.

1.4 Recent Developments

For this project, I will be developing the backend portion of the system. This will include implementing the business logic and building the database. The business logic will be implemented mostly within the models and partially in the controller of the MVC structure since the MVC structure is designed to separate the interface from the substance of applications like so. [5] We have chosen to build the database in MySQL since it is an open source database [6].

This project will likely be released with the Apache 2.0 license. This is a free license like most of the popular open source licenses. The reason for selecting Apache 2.0 is its explicit statement of trademark usage. [7]



Figure 1.5: Development tools being use for the project. [8]

Chapter 2

System Integration and Modeling

2.1 Introduction

The SEA Registration System is being built with several technologies and tools that the developers have used before, and some for the first time. An important aspect of developing functional applications is selecting the right tools for the job, as well as adequately designing the system before beginning coding.

2.2 Technology

This web application is being developed with PHP on the back end, and JavaScript, HTML, and CSS on the front end. PHP is a server side scripting language used by many of today's top Internet companies, including Facebook and Wikipedia, to build dynamic, secure web applications. [9] [10] We will be using Composer to manage dependencies, so that the libraries we use for features such as Captchas and Email work with the system.

Libraries and Frameworks

- **PHP Huge by panique** – User authentication framework coded in PHP that uses the Model View Controller software architectural pattern. We decided on using Huge over other frameworks such as CakePHP or Laravel, which, while large and well documented, use their own flavor of Markup syntax and require the use of custom binaries. Huge, on the other hand, is an incredibly simple MVC framework coded in pure vanilla PHP, and uses regular HTML mixed with normal PHP commands for markup.
- **PHPMailer** – API for sending emails using PHP.
- **Captcha by Gregwar** – Tried and tested PHP Captcha library.

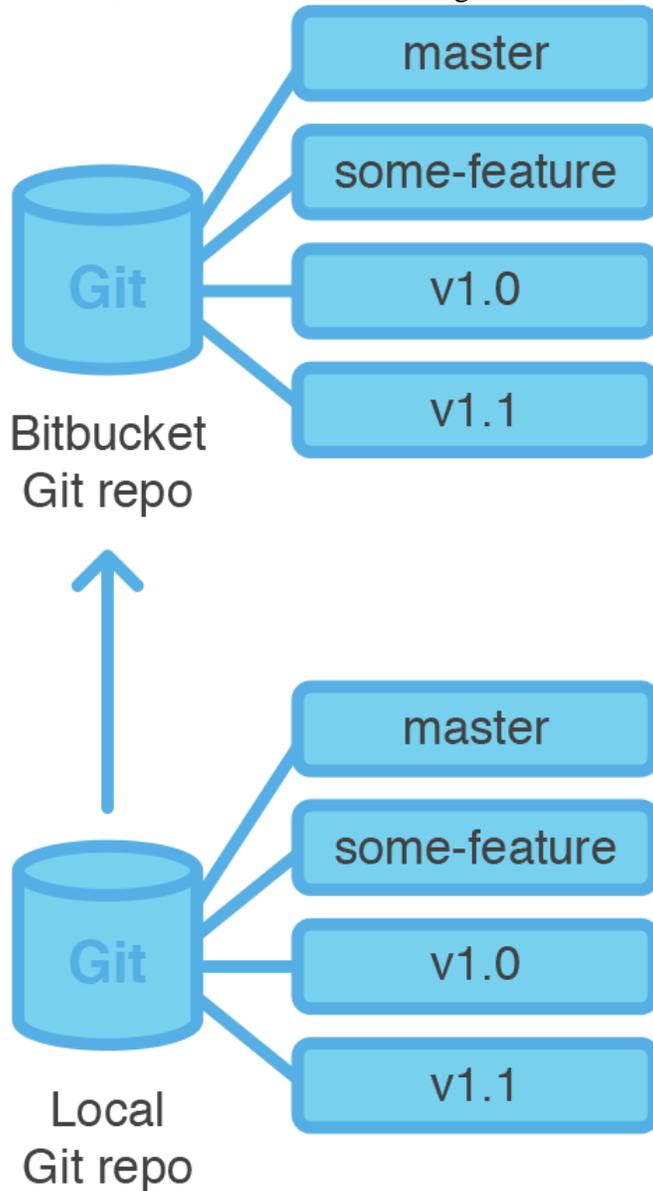


Dependency Manager for PHP

Figure 2.1: Composer is used to manage all of the dependencies within the SEA Registration System. [11]

Source Control

Source control was accomplished through the use of Git and a BitBucket private repository. BitBucket was chosen over GitHub for the free private repository feature. [12] As we work, we check out the project to our local systems. After completing a feature or stopping for the day, all changes are committed and then pushed back to bitbucket. This allows other developers to see what was done, and to add their own changes in turn.



Pushing to the Bitbucket repo

Figure 2.2: Diagram describing how we use git for source control [13]

2.3 System Design

The SEA Registration System includes a web front end for easy registration for parents and administrators, with a database backend to store the data. As my focus has been on the back end, I will present that here. After several iterations, we have simplified the database to just six tables. In the planning stage of the project, we anticipated a separate table for Parents, Instructors, and Staff, but after meeting with a faculty database consultant they suggested combining these tables into a single user's table. Almost all of the fields are shared between these various users, so it did not make sense to repeat them across three tables. Following our goal of DRY (Don't Repeat Yourself) programming, we condensed code anywhere possible.

The SEA Registration System stores data through the actions of models (introduced later in this document) and then presents them to users through views. Data is stored in the following tables:

Table Name	Description
users	Name, contact details, login information, account type, and all other information pertaining to Parent, Instructor, Admin, and Staff users.
student	Names, ages, emergency contacts, and any other information of participants (students) created by Parent and Admin users. Will contain Lottery placement as well in future releases.
session	Name, date range, age range, price, and capacity of all SEA Sessions.
student-session	Session preferences of Students. Will also contain placement and waitlist positions in future releases.
waiver	Media and liability waiver signature status.
payment	Session payment status.

Figure 2.3: Database tables that contain the most important information

A visual representation of the relationships between tables follows. See legend for help interpreting the Entity Relationship Diagram.

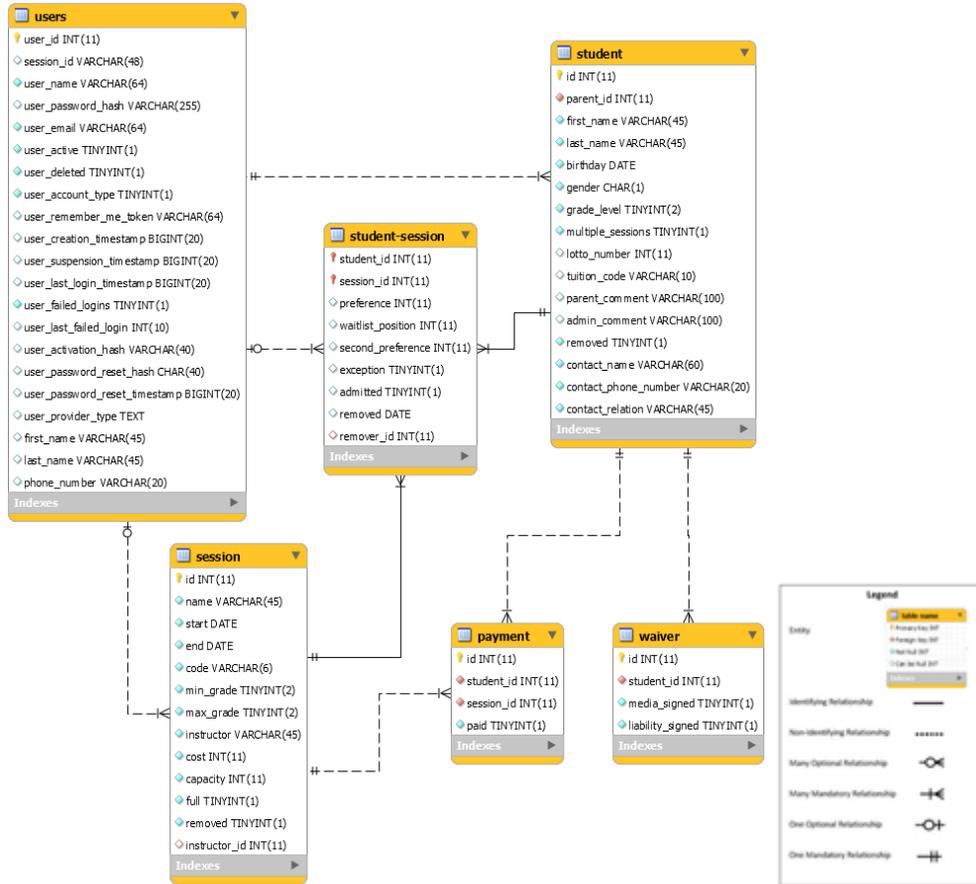


Figure 2.4: The Database Design of the SEA Registration System.

2.4 Components, Development Techniques, and Tools

The SEA Registration System will include both a registration application for parents, and an administrative backend. Data for users and participants will be stored in a database using MySQL. We are using the Model, View, Controller style of development, as explained in the first section. This allows us to separate concerns, and develop features quickly.

Tools

- **PhpStorm** - Powerful IDE for PHP, HTML, CSS, and JavaScript which includes on-the-fly code analysis, error prevention, code completion, and automated refactorings. Helps both front and back end development.
- **PHPMysqlAdmin** - Open source tool written in PHP for administering MySQL over the web. Used to create, modify, and administer our MySQL database.
- **Sublime Text 3** - Powerful text editor that includes thousands of plugins for quickly editing front end code.
- **MySQL** - Open source relational database management system. Central component to the widely used Linux, Apache, MySQL, PHP stack, which we have used for development.

CHAPTER 2. SYSTEM INTEGRATION AND MODELING

- **Git** - One of the most widely used version control systems, Git is open source and distributed. Even if all of the development computers were to fail, the source code is safe and secure.
- **BitBucket** - A web-based hosting service used alongside Git to host our project during development. Offers free private repositories, graphical representation of branches and code commits, and a bug tracking system.
- **Slack** - Easy to use communication tool that combines the benefits of Email, IRC, and file sharing. Developers and project manager communicate through slack near-daily.
- **Wamp** - Short for "Windows, Apache, MySQL, PHP", the developers use wamp to serve the web application on local Windows machines.
- **IIS** - Windows server, what UAA uses to host the site.
- **Chrome** – An advanced web browser, used for viewing and testing the application.
- **Chrome Dev Tools** - A set of web authoring and debugging tools built right into Google Chrome.

2.5 Development Methodology

We have identified various factors that will have an impact on which development methodology would work best for our project.

- Phases are clearly delineated (plan this semester, design in January, develop, test, and deploy in February, April, and May) Some overlap will be necessary
- Our users understand and can clearly state the requirements
- There is little uncertainty in time and cost estimates
- The problem we are solving is well understood
- Our capstone project requires extensive documentation
- We wish to minimize scope creep to stay within our original requirements
- We need to have the project 100% complete by April 8
- This will be the first project of this size for our team members
- The system architecture is unlikely to change
- We don't necessarily need to provide our sponsor with visual signs of progress, beyond standard milestones

After taking into consideration the various factors that will affect project success, we have concluded that a modified waterfall would best fit our development needs. In the waterfall systems development style, the phases of the project are clearly delineated, and include:

- Requirements Analysis
- Planning
- Design
- Development
- Quality Assurance (including system testing)
- Implementation and User Support

In the traditional waterfall model, these phases are separated by formal project reviews. These reviews determine whether the project advances or not, or gets cancelled. While our project will partially follow this process, with planning occurring in fall 2015 and the following phases in spring 2016, some overlap of phases will be necessary. Because of this, we have decided to develop the SEA registration system following a modified waterfall style. A diagram to describe the modified waterfall development model is provided.

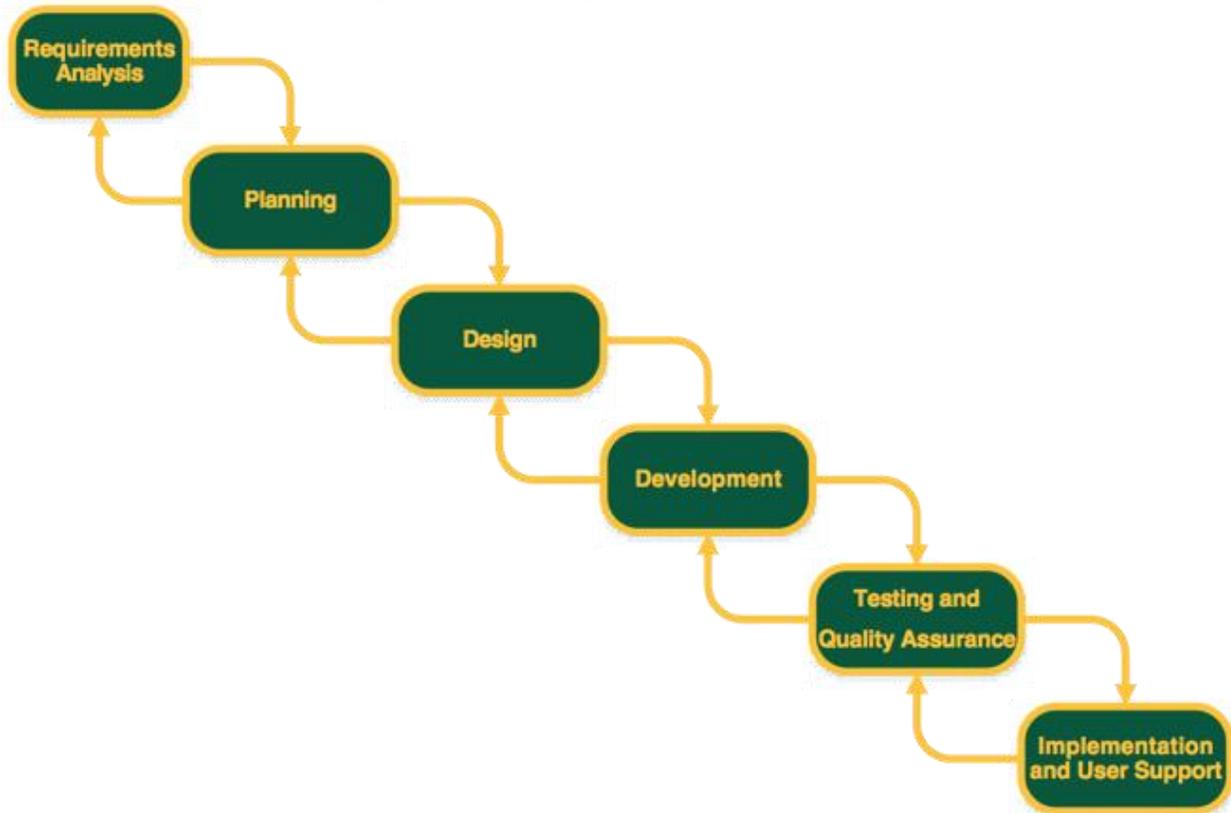


Figure: 2.5: A diagram describing the modified waterfall development cycles.

Chapter 3

Design and Testing

3.1 Introduction

The SEA Registration System will streamline the registration process for parents, and allow the administrator to efficiently manipulate participant information. To work effectively, it is vital that the system has a sound, logical user interface, and is free of bugs and other issues. Because of this, we have gone through various design and testing phases to make sure the application meets the original requirements and works as expected.

3.2 User Interface

The SEA Registration System will include both a public registration application that parents can use to sign up participants, as well as an administrative application for the collection and manipulation of user and participant data. Depending on the account type that is currently logged in, different features will be presented to the user. There is a consistent header throughout the application that allows for navigation for both parent and administrator users.

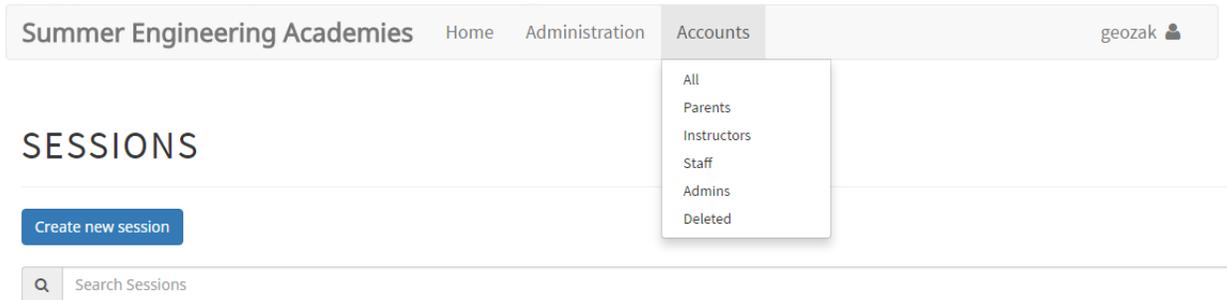


Figure 3.1: The navigation bar contains the most common actions of the application

Parent users will be able to sign up for accounts using a standard registration form, with inputs for important contact information such as name, email, and phone number. Because email is so important for communicating with parents, it is used to activate the account and confirm that it was entered correctly. After creating an account, the parent will then be able to add and edit participant information. Depending on the entered grade level, sessions will then presented to be ordered by preference. An option to sign up for multiple sessions will also be included.

The administrator user interface will include features for registering parent accounts and adding participants and sessions, as well as filtering accounts, sessions, and participants by various criteria.

Summer Engineering Academies					
Home Administration Accounts				geozak 	
STUDENTS					
<input type="text" value="Search Students"/>					
Student ID ^	Parent ID	Name	Date of Birth	Gender	Grade Level
1	1	Johannes Meyer	1992-03-01	M	8
2	6	Little Johnny	2008-02-02	M	6
3	6	jon jim	2001-01-10	M	8
4	6	Bob Johnson	2004-03-31	M	10
5	6	Jane Austen-Texas	2002-04-20	F	10
6	15	Ima Testchild	2003-06-12	F	8
7	14	Thisisa Kidofyours	2004-03-13	M	8
8	14	Example Test	2003-02-22	M	9
9	18	Samantha Veilleux	2010-02-20	F	1
10	21	Little Johnny	2001-03-22	M	9
11	23	Little Sarah	1999-07-02	F	11
12	25	Little Bobby	2000-03-22	M	11
13	27	Bobby '; DROP TABLE participants--;	2016-04-08	O	5

Figure 3.2: Administrative filtering of participants.

3.3 Testing Methodology

Unit Testing

As we are using the Agile method of development, testing occurs as we build and implement new features. This allows us to identify issues that require correction before moving on to other parts of the system. For each new feature built, we make sure that certain input creates the expected output, as well as all functions work correctly. For any feature that adds data to the database, we compare what was entered in the input fields to what was added to the database. This is often called “black box” testing, as the inputs and outputs are seen, but the inner workings aren’t. Testing is also being performed to make sure that if any inputs given from the front end are properly sanitized to make sure the application is immune to security issues such as sql injections. Controls are set in place on both the front end and the back end to make sure that users cannot enter malicious code wherever they are allowed to enter text.

User Acceptance Testing

While unit testing throughout the development process allows us to avoid small bugs becoming major issues, we still need to make sure that users find the system intuitive to navigate, easy to use, and is free of errors. To accomplish this we will be conducting user acceptance tests with various user groups including fellow students and faculty. Test users will create accounts, and then run through the application to make sure all functions work as intended. We will provide a test form to be filled out, including questions about general ease of navigation, as well as specific questions about the results of various tasks such as adding participants and changing account details. After collecting the completed forms, we will make changes based on their feedback and then conduct a subsequent round of user acceptance testing.

Step	Description	Expectation/Objective	Cross Reference Tasks	Priority	Comments
1	User				
1.1	Sign Up User Account	User Account Created		Critical	
1.2	Edit User Information	Edits Saved	1.1	Medium	
1.3	View Sessions	Sessions Displayed	1.1	High	
1.4	Filter Sessions Based on <criteria>	Sessions matching criteria displayed	1.3	Medium	
1.5	Enroll In Session		1.4	Critical	
2	Admin				
2.1	Log In	Admin account logged in		Critical	
2.2	View Users	Display Users	2.1	High	

Figure 3.3: Partial user acceptance test form. Final will include all tasks.

3.4 Software Development Tools and Techniques

The back end of the SEA Registration System was developed using a variety of tools. PHPStorm was used for the bulk of the PHP coding, as it includes many features such as syntax highlighting, code refactoring, and auto complete. [14] MySQL, in conjunction with PHP MyAdmin, is being used to create and administrate the database. Because it is unwise to try to handle all security of an internet connected application from scratch, we are using the Huge framework for user authentication. This framework also allows the Model, View, Controller software architecture. Models handle the modeling of the data, along with the business logic. Views contain the styling and user interface that is presented to the user. Controllers coordinate the entire application by accepting input from users, sending data to the relevant models, and presenting views back to the user. [15]

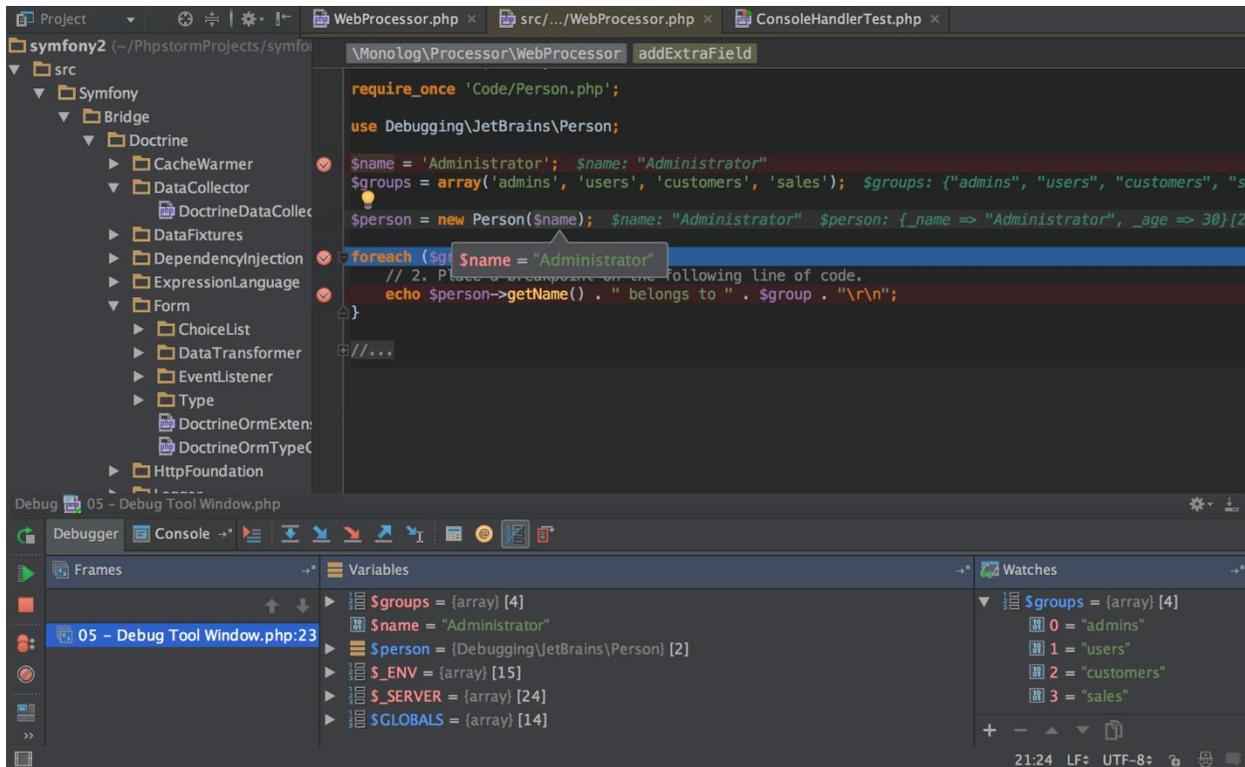


Figure 3.4: Debugging in PhpStorm [16]

3.5 Agile Coding

As the project has progressed, we have begun implementing an agile approach to how we develop the SEA Registration System. The team meets twice a week, and each meeting includes planning what will be completed for the following meeting. This has allowed us to break the project into incremental goals, and work on features one at a time. Each week, the developers get assigned tasks, and work to complete them before the next meeting.

Agile coding is unique in that it involves testing throughout development, rather than just during the testing phase. [17] While we code new features, we test to make sure they are working as expected. Agile coding helps avoid “code and fix” mentality that can happen at the end of larger, traditional projects that include testing at the very end only.

Waterfall to Agile Development

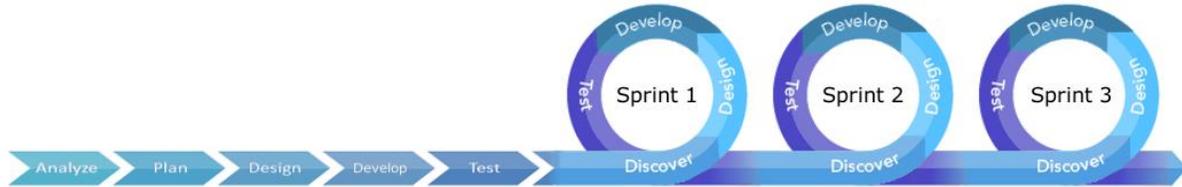


Figure 3.5: The modified waterfall agile development process we've taken. [18]

Chapter 4

User Manual

4.1 Introduction

As the SEA Registration System is still under development, this User Manual pertains to the 0.6 Alpha release version. Not all planned features are yet present, and will be added throughout the next two weeks. An updated edition of this document will be provided alongside the Beta release of the SEA Registration System, currently planned for April 9th. Until this future release, the live application will remain frozen for testing purposes while development will continue on the remaining features on a different project branch.

The SEA Registration System is being developed for the UAA School of Engineering Department's Summer Engineering Academies. It allows parents to register accounts, add applications for their children, select those child's session preferences, and view the status of each application. Additionally, it helps facilitate SEA administrator duties, and replaces the current system composed of using various third party systems. As the popularity of SEA continues to grow from year to year, it has become important to have a simplified, easy to use application.

Development on the SEA Registration System has progressed through the Requirements Analysis, Planning, and Design phases, and is currently in the Development and Testing phase. This 0.6 Alpha pre-release version has been opened to the public so that core features of account registration, student creation, and session manipulation can be tested.

4.2 Software Requirements

Operating System	Microsoft Windows (7 or later suggested) OS X (Version 10.7 “Lion” or later suggested) Linux (Most major distributions)
Web Browser	Chrome (suggested) Firefox

4.3 General Layout

Throughout the application, there is a header menu theme that contains the most important sections of the site.



 *Parent users have account settings to the right of the header, and a home link to the left that leads to any created participant registrations.*



 *Administrators have more options than other user types.*

Text boxes are accompanied by descriptive labels to the left or above their corresponding text areas, as well as placeholder messages indicating text requirements.

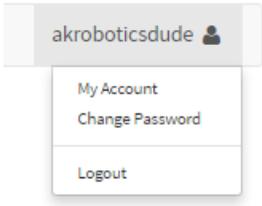
Username

Dates: Through:

4.4 Menus and Navigation

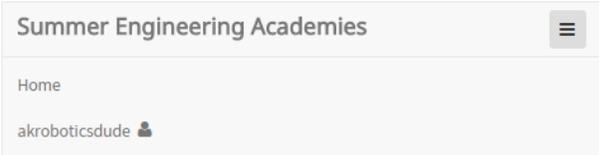
Computer Browsers

Menu styles change depending on the size of the device screen rendering the application. For most computers and larger tablets, menus will appear when items in the header are clicked:



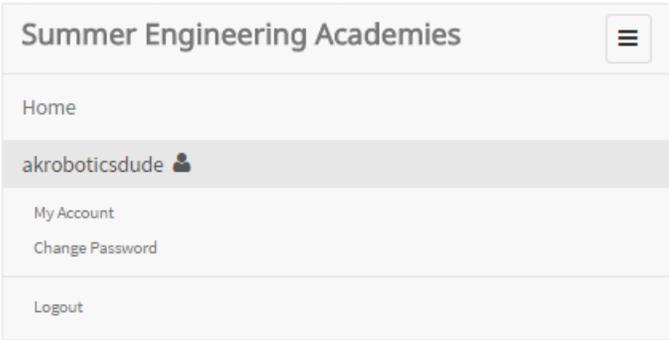
Mobile Browsers

When viewing the SEA Registration devices such as tablets, menus appear icons that, include the menu options:



System from smaller as smart phones, the as typical “hamburger” when pressed, expand to

After the menu is expanded by clicking on the username, further options are displayed below



the username. Throughout the application, this method of opening menus can be used to quickly navigate from page to page without repeatedly zooming in and zooming out when using a smaller device. The SEA Registration System was developed to reach the widest audience possible, regardless of their preferred device.

Shortcuts

From the home “Summer Engineering Academies” header below the banner image.



wherever you are in the application, page can be reached by clicking on the



Icons

While on mobile, look for the icon. It indicates that menu options are available, and, when clicked, will expand to show the additional menus and options.



While subtle, the color difference between the upper (mouse not hovering) and lower (mouse actively hovering) headers indicates an action available

4.5 Functions

The above section presents a high level overview of the application, while the rest of this User Manual will introduce specific functions that can be completed using the SEA Registration System.

Login

At the home page, select the “Login” option, either from the menu option in the header, or the button below the welcome image.



Register

To register a new account, select “Register” from the top menu bar:



The register form will now display. Fill it out, keeping in mind to match all patterns presented in the default placeholder text.

REGISTER

Username
(letters/numbers, 2-64 chars)

Email
(a real address)

Repeat Email
(to prevent typos)

Password
(6+ characters)

Repeat Password
(to prevent typos)

First Name
(upto 45 characters)

Last Name
(upto 45 characters)

Phone Number
(numbers only, 7-20 chars)



Please enter above characters
Enter captcha above

[Reload Captcha](#)

[Register](#)

 *Make sure to add a current email, as registration requires email verification.*

To complete registration, go to the email account specified and follow the verification link. While steps have been taken to ensure delivery, if the link doesn't appear after refreshing your inbox, check the spam folder. If it still hasn't arrived, request a validation email resent.

Resend Validation Email

If the validation email fails to arrive, request a resend at the login prompt.

Your account is not activated yet. Please click on the confirm link in the mail. [Click here](#) to resend the link. ×

LOGIN

Username

Password

Remember me

[Log in](#) [Forgot your password?](#)

After clicking on the “Click here” link, enter the email used to register into the validation form.

RESEND VALIDATION EMAIL

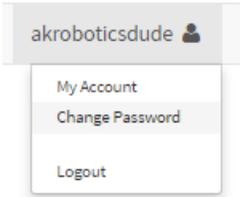
Submit your email address and we'll resend your validation email.

Email address:

[Submit](#)

Change Password

To change your password, click the username in the upper right corner, and select Change Password



 Passwords must be at least 6 digits in length.

Edit Account Information

To edit account information, click the username in the upper right corner and select “My Account”

Summer Engineering Academies Home akroboticsdude

- My Account
- Change Password
- Logout

ACCOUNT INFORMATION

Username:	<input type="text" value="akroboticsdude"/>	Save Username
Email:	<input type="text" value="akroboticsdude@johannescmeyer.com"/>	Save Email
First Name:	<input type="text" value="Ima"/>	Save First Name
Last Name:	<input type="text" value="Testuser"/>	Save Last Name
Phone Number:	<input type="text" value="9073338978"/>	Save Phone Number

[Change Password](#) [Delete Account](#)

Create Participant

Participants, also called Students or Children, can be created from the home page of any Parent account. Late in this guide, Administrative participant creation is presented. For parents, clicking on “Create Participants” will allow you to create applications for your participants.

Summer Engineering Academies Home akroboticsdude

PARTICIPANTS

[Create Participant](#)

Fill out the New Participant form, making sure to follow the input guidelines shown in the placeholder text.

CREATE A NEW PARTICIPANT

First name:

Last name:

Birthday:

Male Female

Grade Level (as of fall 2016):

Emergency Contact

Name:

Phone Number:

Relation:

[Create Participant](#)

Select Session Preferences

After a participant is entered into the system, you will be prompted to select their session preferences.

SESSION PREFERENCES

Session Preferences for Ima Testchild

Session Preference 1:
 RB-M1 Jun 06 to Jun 10 Energy
 None
RB-M1 Jun 06 to Jun 10 Energy
 EG-M1 Jun 06 to Jun 10 Alternative Energy
 RB-M2 Jun 13 to Jun 17 Lego Robotics
 WA-M3 Jun 20 to Jun 24 Wing Aerodynamics
 RB-M3 Jun 20 to Jun 24 Robotics
 WA-M4 Jun 27 to Jul 01 Wing Aerodynamics
 CC-M6 Jul 18 to Jul 22 Creative Coding
 SD-M6 Jul 18 to Jul 22 Structure Destruction (MatSu)
 SD-M7 Jul 25 to Jul 29 Breaking Buildings
 CC-M7 Jul 25 to Jul 29 Creative Coding (MatSu)
 SD-M8 Aug 01 to Aug 05 Breaking Buildings

Session Preference 5:
None

Session Preference 6:
None

Session Preference 7:
None

Session Preference 8:
None

Session Preference 9:
None

Session Preference 10:
None

Session Preference 11:
None

Are you interested in more than one session if available?

Are you interested in more than one session if available?



Select the “Are you interested in more than one session if available?” option to you would like the participant to be placed in multiple sessions as available.

[For more details about the sessions please visit the Main Site.](#)



For more details about the sessions, visit the Summer Engineering Academies main site at: https://www.uaa.alaska.edu/collegeofengineering/sea_academydescriptions.cfm

After choosing preferences, the participant can be found in the “Home” page.

PARTICIPANTS

Ima Testchild	Basic Information	Session Preferences
---------------	-------------------	---------------------

Create Participant

Edit Participant Information

Select “Basic Information” to edit the participant information.

PARTICIPANT INFORMATION

First Name:	<input type="text" value="Ima"/>	<input type="button" value="Save First Name"/>
Last Name:	<input type="text" value="Testchild"/>	<input type="button" value="Save Last Name"/>
Birthday:	<input type="text" value="06/12/2003"/>	<input type="button" value="Save Birthday"/>
Gender:	<input checked="" type="radio"/> Male <input type="radio"/> Female	<input type="button" value="Save Gender"/>
Grade Level: (as of fall 2016)	<input type="text" value="8"/>	<input type="button" value="Save Grade Level"/>

Emergency Contact

Name:	<input type="text" value="Ima Testcontact"/>	
Phone Number:	<input type="text" value="8873334444"/>	
Relation:	<input type="text" value="Hairdresser"/>	<input type="button" value="Save Emergency Contact"/>

Delete Participant:

Delete Participant

Within the “Participant Information” page, there is an additional “Delete Participant” option. If needed, select this to delete the application.

Edit Participant Preferences

From the “Home” page, select “Session Preferences” alongside whichever participant you wish to edit preferences for.

Admin

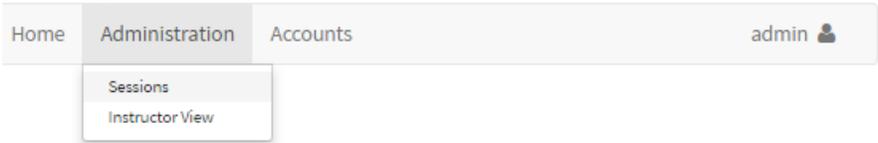
Logging in as an admin user will present the “Admin” page.

From this page, the administrator can currently:

1. View and Create Sessions.
2. View Instructor Information.
3. View and Create Parent, Instructor, and Staff Accounts.
4. View and Create Administrator Accounts.



View Sessions



All currently created session are then displayed.

SESSIONS

Edit	Name	Dates	Grade Levels	Instructor Name	Instructor ID	Cost	Capacity
RB-M1	Energy	06-Jun-2016 to 10-Jun-2016	6 to 8	Elaine Todd	None	100	30
EG-M1	Alternative Engery	06-Jun-2016 to 10-Jun-2016	6 to 8	Dr. Matt Kupilik	None	100	36
EG-H2	Alternative Engery	13-Jun-2016 to 17-Jun-2016	9 to 12	Dr. Matt Kupilik	None	100	36
RB-M2	Lego Robotics	13-Jun-2016 to 17-Jun-2016	5 to 8	Elaine Todd	None	100	24
WA-M3	Wing Aerodynamics	20-Jun-2016 to 24-Jun-2016	6 to 8	Dr. Jifeng Peng	None	100	36
RB-M3	Robotics	20-Jun-2016 to 24-Jun-2016	7 to 8	Elaine Todd	None	100	24
SD-H5	Advanced Structures	11-Jul-2016 to 15-Jul-2016	9 to 12	Dr. Scott Hamel	None	200	28
WA-M4	Wing Aerodynamics	27-Jun-2016 to 01-Jul-2016	6 to 8	Dr. Jifeng Peng	None	100	36
RB-H4	Robotics	27-Jun-2016 to 01-Jul-2016	9 to 12	Dr. Scott Hamel	None	100	24
RB-H5	Arduino Robotics (MatSu)	11-Jul-2016 to 15-Jul-2016	9 to 12	Michael Backus	None	200	24
CC-M6	Creative Coding	18-Jul-2016 to 22-Jul-2016	6 to 8	Dr. Kenrick Mock	None	100	30
SD-M6	Structure Destruction (MatSu)	18-Jul-2016 to 22-Jul-2016	6 to 8	Dr. Scott Hamel	None	100	36
SD-M7	Breaking Buildings	25-Jul-2016 to 29-Jul-2016	6 to 8	Dr. Scott Hamel	None	100	36
CC-M7	Creative Coding (MatSu)	25-Jul-2016 to 29-Jul-2016	6 to 8	Zak Stone	None	100	30
SD-M8	Breaking Buildings	01-Aug-2016 to 05-Aug-2016	6 to 8	Dr. Utpal Dutta	None	100	36

Create Session

Create Sessions

Click "Create Session" to begin adding new sessions. Make sure to follow the guidelines presented with the placeholder text.

CREATE A NEW SESSION

Session name:	<input type="text" value="session name (upto 45 characters)"/>		
Dates:	<input type="text" value="mm/dd/yyyy"/>	Through:	<input type="text" value="mm/dd/yyyy"/>
Session code:	<input type="text" value="session code e.g. RB-M1"/>		
Grades:	<input type="text" value="min grade (numbers only)"/>	Through:	<input type="text" value="max grade (numbers only)"/>
Instructor name:	<input type="text" value="session instructor (upto 45 characters)"/>		
Cost:	<input type="text" value="\$ Session cost in dollars (numbers only)"/>		
Capacity:	<input type="text" value="Session capacity (numbers only)"/>		
<input type="button" value="Create Session"/>			

Instructor View

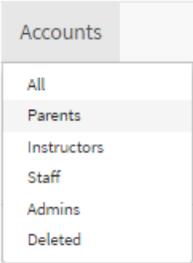
Currently this view presents the administrator with all of the Instructors and their assigned sessions. In future releases, administrators will be able to sort by the column headings to easily find needed details. This is also where Instructors will be able to view all of their assigned sessions, along with attendance sheets containing important information such as participant emergency contacts, ages, and allergies.



Administrators are able to see the same views as all other account types.

View Parents

From the Accounts drop down menu, select Parents



From here click on View/Edit to manipulate any of the fields. To create a new parent account select the “Create new parent” button.

MANAGE ACCOUNTS/PARENT

[Create new account](#) [View all accounts](#)

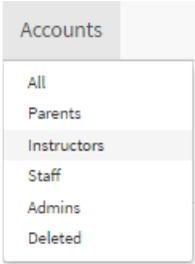
parent

[Create new parent](#)

Id	Username	Email	Activated ?	First Name	Last Name	Phone Number	
1	zrstone	zrstone@alaska.edu	Yes	Zak	Stone	9073178216	View/Edit
2	jimBronieHomie	jim@johannescmeyer.com	Yes	Jim	Bronie-Homie	6664206969	View/Edit
6	SteveBob	stevebob@johannescmeyer.com	Yes	bill	Bob	2222222222	View/Edit
7	cis	cis460authentication@gmail.com	Yes	cis	zak	1234567890	View/Edit
14	anotherstest	anotherstest@johannescmeyer.com	Yes	Another	Test	3624685555	View/Edit
15	akroboticsdude	akroboticsdude@johannescmeyer.com	Yes	Ima	Testuser	9073338978	View/Edit
16	OneMoreUser	onemoreuser@johannescmeyer.com	Yes	Onemore	User	74635422558	View/Edit

View Instructors

From the Accounts drop down menu, select Instructors.



From here click on View/Edit to manipulate any of the fields. To create a new instructor account select the "Create new instructor" button.

MANAGE ACCOUNTS/INSTRUCTOR

Create new account View all accounts

instructor

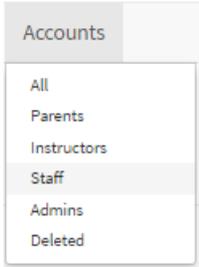
Create new instructor

Id	Username	Email	Activated ?	First Name	Last Name	Phone Number	
8	8instructor	elaine@johannescmeyer.com	No	Elaine	Todd	3335552222	View/Edit
9	9instructor	kupilik@johannescmeyer.com	No	Matt	Kupilik	6643332222	View/Edit
10	10instructor	peng@johannescmeyer.com	No	Jifeng	Peng	9073332222	View/Edit
11	11instructor	shamel@johannescmeyer.com	No	Scott	Hamel	9073787654	View/Edit
12	12instructor	zrstone@gmail.com	No	Zak	Stone	9076547865	View/Edit

i Instructor usernames are automatically generated by the system to include their user id number followed by their account type.

View Staff

From the Accounts drop down menu, select Staff.



From here click on View/Edit to manipulate any of the fields. To create a new staff account select the "Create new staff" button.

MANAGE ACCOUNTS/STAFF

Create new account View all accounts

staff

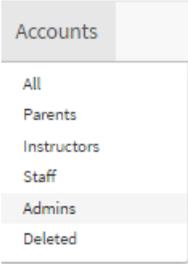
Create new staff

Id	Username	Email	Activated ?	First Name	Last Name	Phone Number	
5	5staff	staff@johannescmeyer.com	No	Tom	Planks	3333333333	View/Edit
17	17staff	ChraTheRocket@johannescmeyer.com	No	Crah	Tchrocket	3332224444	View/Edit

i Staff usernames are automatically generated by the system to include their user id number followed by their account type.

View Administrators

From the Accounts drop down menu, select Admins.



All current administrators will be displayed. From here click on View/Edit to manipulate any of the fields. To create a new administrator account select the “Create new administrator” button.

MANAGE ACCOUNTS/ADMIN

[Create new account](#) [View all accounts](#)

admin

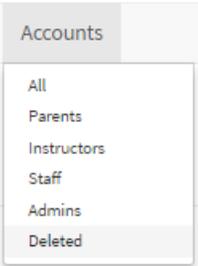
[Create new admin](#)

Id	Username	Email	Activated ?	First Name	Last Name	Phone Number	
3	geozak	zakarystone@gmail.com	Yes	Zak	Stone	9073178216	View/Edit
13	admin	admin@johannescmeyer.com	Yes	Example	Administrator	3141592653	View/Edit

 *It is best practice to have new administrators set their own password. At the very least, require them to change it after turning the account over.*

View Deleted Accounts

When accounts are deleted, they are held in the database and are able to be reactivated at the administrator’s discretion. To view deleted accounts select “Deleted” under the accounts menu.



All deleted accounts will then be displayed, and from there can be edited and/or reactivated by clicking the “View/Edit” option.

MANAGE ACCOUNTS/DELETED

Create new account View all accounts

deleted

Id	Username	Email	Activated ?	Role	First Name	Last Name	Phone Number	
4	4instructor	test@johannescmeyer.com	No	instructor	Johannes	Mayers	9072680143	View/Edit

- While it may be tempting to edit usernames that are offensive in some way, it is better to instead deactivate the account and reach out to the user to inform them of any rules broken. Depending on the severity, they may then be allowed to request a new username.

USER ACCOUNT INFORMATION

User ID: 4

Account type: Save Account Type

Username: Save Username

Email: Save Email

First Name: Save First Name

Last Name: Save Last Name

Phone Number: Save Phone Number

New password:

Repeat new password: Submit new password

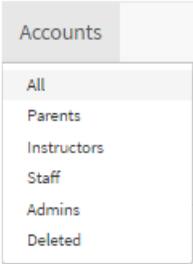
Change Account Status:



Admins have the ability to delete, or deactivate accounts, and re-enable or reactivate them.

View All Accounts

From the Accounts drop down, select All Accounts.



Deactivate Accounts

At times the administrator will be forced to deactivate accounts, whether because a user is acting maliciously, breaking UAA’s terms of service in some way, or simply because they’ve requested their account be deleted. With the details of the account in question pulled up, select the “Soft Delete” button in the “Change Account Status:” section of the account details.

New password:

Repeat new password:

Change Account Status:

 *“Soft Delete” is being used in the Alpha release for testing to distinguish between deactivation of accounts (soft deleting) and complete removal from the database (hard delete).*

While account deletion can be reversed, there may be times when the Administrator accidentally removes an account without noticing. To confirm that the action has been requested, a pop up box is presented. Press “OK” to deactivate the account, and “Cancel” to, well, you guessed it: cancel.

USER ACCOUNT INFORMATION

User ID: 4

Account type:

Username:

Email:

First Name:

Last Name:

Phone Number:

New password:

Repeat new password:

Change Account Status:

searegistration.uaa.alaska.edu says:

Are you sure you want to delete this account?

Chapter 5

Summary

5.1 Introduction

This application has been developed so that parents can easily manage their applications for the Summer Engineering Academies online, and administrators can manipulate data quickly and without error. Using modern technologies such as PHP and MySQL on the back end has led to a modern, dynamic application that is secure and easily maintainable. As a team, we have overcome challenges and mitigated risks throughout the project, and worked to stay within scope. The final features are being developed, and testing conducted, to make sure that the application meets the original requirements.



Figure 5.1: The site is live!

5.2 Implications

The SEA Registration System replaces the current system of using Qualtrix surveys, excel spreadsheets, and various other services to administrate participant applications. The three main objectives of the SEA Registration are to:

1. Streamline the registration process for parents.
2. Consolidate administrative functions into a unified system.
3. Reduce administrator workload by automating common business procedures.

These objectives should lead to less frustrated parents and administrators who use the system. The parents will be able to track and manage their participants' sessions online, and administrators will have all of the benefits of a modern application with a database back end.

5.3 Future Development

Having built the SEA Registration System using the Model View Controller architecture, there is the potential for code reuse, and easy maintainability. As the Summer Engineering Academies continue to grow, new features may be added to handle the popularity. Additionally, if the program spreads to other cities in Alaska, it will be easy to expand the application. Throughout development, we have been compiling a list of features that SEA would benefit from in a future version of the registration system.

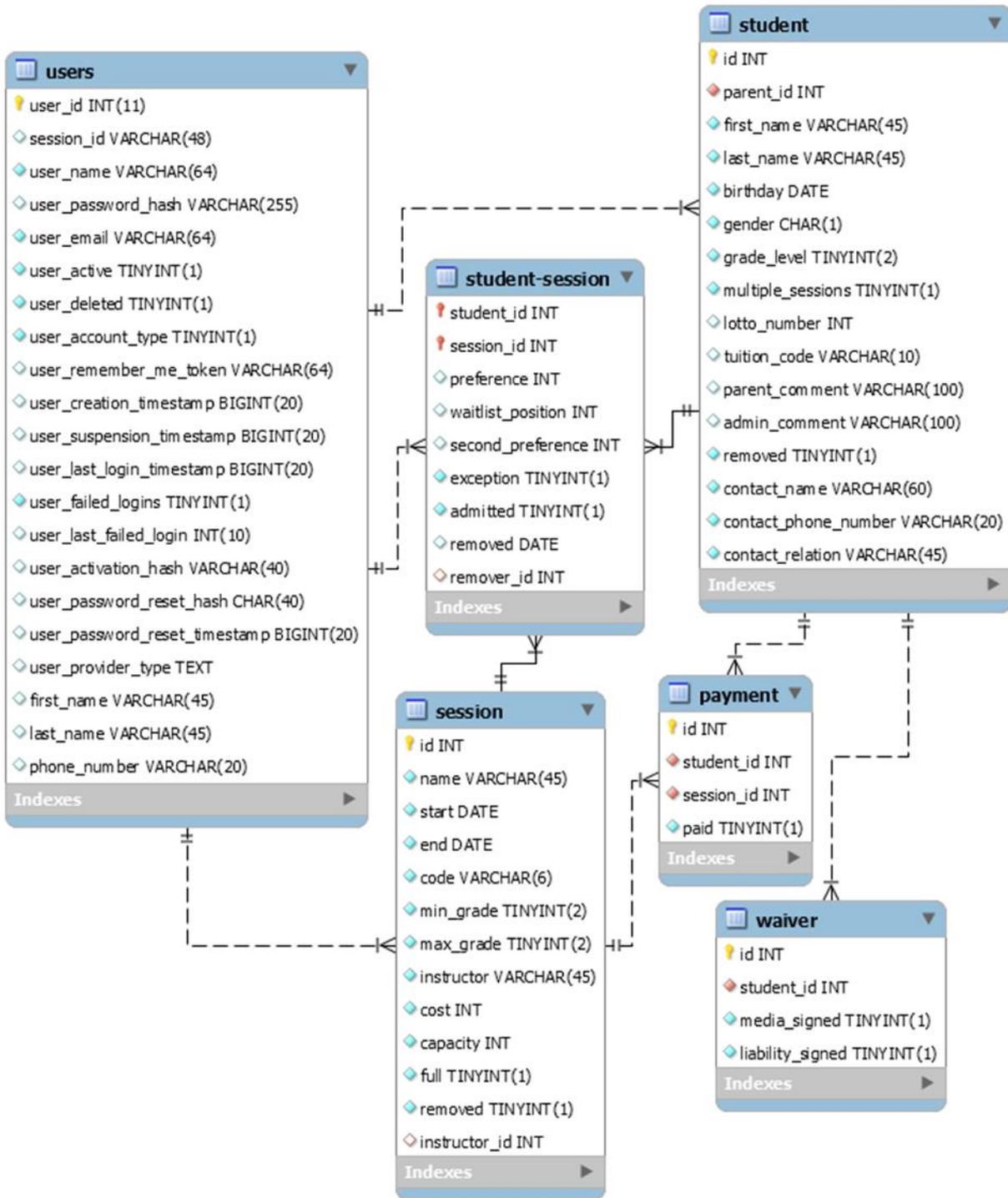
Bibliography

- [1] T. Kalytiak, "Kids find their bliss at UAA summer engineering academies - Green & Gold News", *Green & Gold News*, 2015. [Online]. Available: <http://greenandgold.uaa.alaska.edu/blog/35250/kids-find-their-bliss-at-uaa-summer-engineering-academies/>. [Accessed: 09- Mar- 2016].
- [2] Nirutechnolabs, "php MVC concept advantages", *Stackoverflow.com*, 2016. [Online]. Available: <http://stackoverflow.com/questions/8833689/php-mvc-concept-advantages/29762735#29762735>. [Accessed: 09- Mar- 2016].
- [3] "Model–view–controller", *Wikipedia*, 2016. [Online]. Available: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. [Accessed: 09- Mar- 2016].
- [4] panique, "panique/huge", *GitHub*, 2015. [Online]. Available: <https://github.com/panique/huge>. [Accessed: 09- Mar- 2016].
- [5] J. Diepenmaat, "The Model View Controller pattern in web applications", *Zeekat.nl*, 2005. [Online]. Available: <https://zeekat.nl/articles/mvc-for-the-web.html>. [Accessed: 09- Mar- 2016].
- [6] "MySQL", *Mysql.com*, 2016. [Online]. Available: <https://www.mysql.com/>. [Accessed: 09- Mar- 2016].
- [7] "Licenses", *Choose a License*, 2016. [Online]. Available: <http://choosealicense.com/licenses/>. [Accessed: 09- Mar- 2016].
- [8] "Weather by You! Weather newbie's corner", *Weatherbyyou.com*, 2016. [Online]. Available: http://weatherbyyou.com/diy_wamp.php. [Accessed: 09- Mar- 2016].
- [9] G. Clarke, "Facebook re-write takes PHP to an enterprise past", *Theregister.co.uk*, 2016. [Online]. Available: http://www.theregister.co.uk/2010/02/02/facebook_hiphop_unveiled/. [Accessed: 09- Mar- 2016].

- [10]"Manual:Installation requirements - MediaWiki", Mediawiki.org, 2016. [Online]. Available: https://www.mediawiki.org/w/index.php?title=Manual:Installation_requirements&oldid=299556#PHP. [Accessed: 09- Mar- 2016].
- [11]"Composer", Getcomposer.org, 2016. [Online]. Available: <https://getcomposer.org/>. [Accessed: 09- Mar- 2016].
- [12]A. Zhitnitsky, "Bitbucket VS. GitHub: It's More Than Just Features | Takipi Blog", Takipi Blog, 2014. [Online]. Available: <http://blog.takipi.com/bitbucket-vs-github-its-more-than-just-features/>. [Accessed: 09- Mar- 2016].
- [13]"Git Tutorials and Training", Atlassian, 2016. [Online]. Available: <https://www.atlassian.com/pt/git/migration#!migration-share>. [Accessed: 09- Mar- 2016].
- [14]"PhpStorm Features", JetBrains.com, 2016. [Online]. Available: <https://www.jetbrains.com/phpstorm/features/>. [Accessed: 09- Mar- 2016].
- [15]"Model View Controller(MVC) In PHP Tutorial", Php-html.net, 2009. [Online]. Available: <http://php-html.net/tutorials/model-view-controller-in-php/>. [Accessed: 09- Mar- 2016].
- [16]"PhpStorm IDE :: JetBrains PhpStorm", JetBrains, 2016. [Online]. Available: <https://www.jetbrains.com/phpstorm/>. [Accessed: 09- Mar- 2016].
- [17]Guru99.com, 2016. [Online]. Available: <http://www.guru99.com/agile-scrum-extreme-testing.html>. [Accessed: 09- Mar- 2016].
- [18]J. Taylor, "Using Both The Waterfall And Agile Methods | StatSlice Business Intelligence And Analytics | Business Intelligence And Analytics Dallas | Business Intelligence Training Dallas", Statslice.com, 2016. [Online]. Available: <http://www.statslice.com/using-both-the-waterfall-and-agile-methods>. [Accessed: 09- Mar- 2016].

Appendix A.

Entity Relationship Diagram



Appendix B.

Initial source code example

HomeController.php

```
1  <?php
2
3  class HomeController extends Controller
4  {
5      /**
6       * Construct this object by extending the basic Controller class
7       */
8      public function __construct()
9      {
10         parent::__construct();
11     }
12
13     /**
14      * Handles what happens when user moves to URL/home/index - or - as
15      * this is the default controller, also
16      * when user moves to /home or enter your application at base level
17      */
18     public function index()
19     {
20         if (LoginModel::isUserLoggedIn()) {
21             Redirect::home();
22         }
23         $this->View->render('home/index');
24     }
25 }
```

Appendix C.

Source Control Repository Access

Service Provider: <https://bitbucket.org/>
Login-Email: cis460authentication@gmail.com
Login-Password: seaguest