

UNIVERSITY OF ALASKA ANCHORAGE

CSCE A470

CAPSTONE PROJECT

**Summer Engineering Academies
Registration System
Web Application and User Interface**

Author:

Christopher Brooks

Supervisor:

Prof. Scott Hamel, PhD

Anchorage AK, May 2016



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

© Copyright 2016
by
Christopher Brooks
christopher.brooks19@gmail.com
Version 5.3

Abstract

The purpose of this capstone project is to provide a web application interface and database system to allow University of Alaska Anchorage's Engineering Departments Summer Engineering Academies summer program to manage the student registrations and related data. Additionally, the system should allow for the users to update their existing information, as well as register children and set their camp session preferences.

The system has been analyzed previously. During the Fall 2015 semester the project team conducted extensive research and analysis of the system's requirements and business processes.

The resulting system will be required to maintain a database of students enrolled in the summer camp. It will provide an intuitive web application user interface. It will conduct the necessary administrative actions required for the successful operation of the program.

Acknowledgments

I would like to thank our sponsor. Dr Scott Hamel, who administers the Summer Engineering Academies. The program is an excellent way to introduce our youth to science and engineering, and I am glad I was able to participate in improving it. I would also like to thank him for being an excellent sponsor throughout the entire project.

Thank you to the Military and Veterans Student Services, who I had the pleasure of working with for over two years. They are an outstanding group of people who have helped me to navigate the college environment and become a successful student.

Finally, thank you to my wife Kristi for being an amazing partner and supporting me throughout my college experience.

Tables

Table 2.1: Projected dates for the projects major milestones.

Table 2.2: The projects Functional and Non-Functional requirements.

Figures

Figure 1.1: SEA's Current Logo

Figure 1.2: A portion of the Qualtrics Survey platform.

Figure 1.3: Students participate in one of the camps robotics classes.

Figure 1.4: The MIT license is detailed in full at the *Open Source Initiative*, at <https://opensource.org/licenses/MIT>

Figure 1.5: Major technologies employed in the application.

Figure 2.1: For a project to be successful it is necessary to maintain project scope. Scope cannot be expanded without sacrificing quality, increasing cost, or extending time to completion.

Figure 2.2: A simplified GANTT chart showing the projects task scheduling.

Figure 2.3: *phpMyAdmin* is the database administration tool at the heart of most AMP (Apache, MySQL, PHP) stacks. It has been used throughout the project to manage the MySQL database that is essential to the successful operation.

Figure 2.4: Early Web Application page showing design and styling of a page to resend account activation and validation email.

Figure 2.5: A teams overlapping skill set allows for each developer to assist with work for the other should it become necessary.

Figure 3.1: The navigation bar for the web application.

Figure 3.2: The user account icon and dropdown menu.

Figure 3.3: The google login page. Pertaining user actions are placed in the upper right hand corner of the page.

Figure 3.4: The development methodology cycle.

Figure 3.5: Git based version control has allowed for rapid development without negative effects on team members individual work.

Figure 3.6: A sample of class and tree based documentation generated from PHP code.

Figure 4.1: The ambitious scope of the project is reflected in this early site layout.

Figure 4.2: An early site map for the application.

Figure 4.5: A sample of class and tree based documentation generated from PHP code.

Figure 5.1: Current login page for the web application.

Contents

Abstract	3
Acknowledgments	4
Tables	5
Figures	6
Chapter 1	9
Introduction	9
1.1 Introduction	9
1.2 Motivation	11
1.3 Application	12
1.4 Recent Developments	12
Chapter 2	14
System Integration, Modeling & Methodology	14
2.1 Web 2.0	14
2.2 Application Scope	14
2.3 System Requirements	15
2.3 Components	17
2.4 Provide a Rich User Experience	18
2.6 The Agile Coding Methodology	18
Chapter 3	20
Design, Testing, and User Interface	20
3.1 User Interface	20
3.2 Software Development Techniques and Tools	23
3.3 Github Update	24
3.4 Agile Methodology	24
3.5 Documentation	25
Chapter 4	26
User Manual	26
Introduction	26
Icons Used in this User Manual	26
System Overview	27
History of the Solution	28
Connecting to the SEA Registration Application	29
Structuring of the SEA Registration Application	32
Functions	33
Admin	38

Features Planned for April 9 th Beta Release.....	45
Chapter 5	47
Summary and Conclusion	47
5.1 Summary	47
5.2 Implications	48
5.3 Recommendations and Improvements	48
5.4 Conclusion.....	49
References	50
Appendix A	51
Appendix B	52

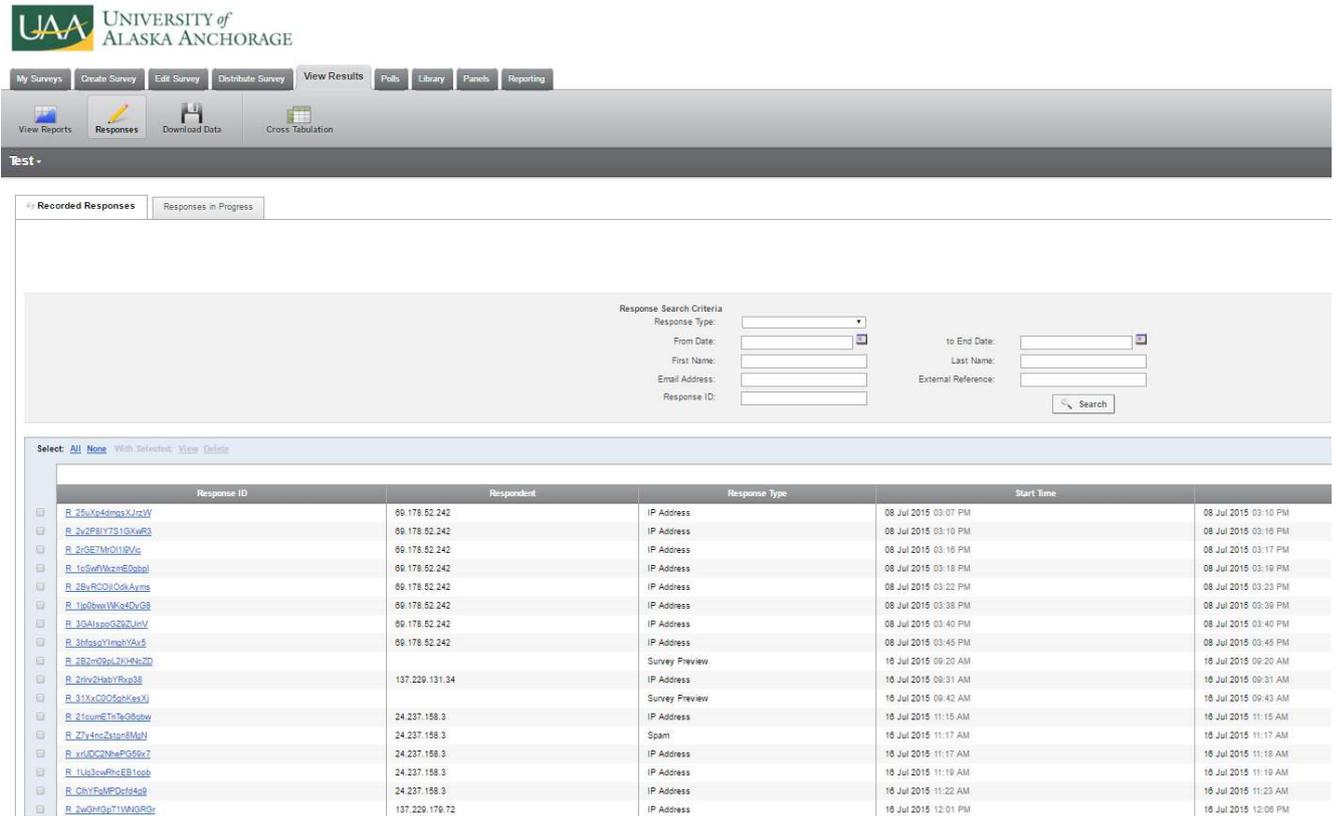


Figure 1.2: A portion of the Qualtrics Survey platform.

The administration of the data relating to student applications, safety waivers, payment information, and session preferences has led to a system of increasing complexity that has become difficult for the administrators to maintain.

The inception of this Capstone Project began in 2015 with a detailed requirements gathering and systems analysis process. This process involved interviews with the Director of Summer Engineering Academies, Dr. Scott Hamel, who is also a professor for UAA's College of Engineering. In addition to the interviews of the key administrators of SEA, the business process model of the existing system was analyzed.

The improvement of the current registration system in place would allow for decreased time spent on administrative tasks by camp administrators. It would provide for a less confusing interface for the parents who are required to use the system to register their children for the camps, as well as adjust preferences for camp waitlists. Additionally, all other interaction is down through email. When a student is accepted in to a camp session, their parent is sent an email. The email gives them instructions on how to proceed. These emails are currently manually created and sent by camp staff. The improved system would allow for automatic email distribution lists.



Figure 1.3: Students participate in one of the camps robotics classes.

1.2 Motivation

The motivation for developing the project is to increase the productivity and reduce the workload and execution of redundant tasks. Additionally, the system will streamline the process for applicants. Allow them to directly interact with their student records and preferences. Previously, the degree of separation created by the use of the Qualtrics surveys that were then manually entered into a spreadsheet before finally an email could be sent updated a parent of the status of their account or their child's placement into a camp, created an undesirable delay for parents to receive information and make informed decision about their child's placement. For the support and administrative staff who run Summer Engineering Activities, improving these response times would alleviate extraneous communication with parents. By decreasing the amount of time spent on manual tasks, greater time can be spent on the fundamentals of the camp and one-on-one customer support.

1.3 Application

The proposed project solution is to develop a modern web application featuring a front-end with a SQL database backend. The application will replace both the Qualtrics surveys that are currently used for parents to register for the camp as well as sign-up for available classes. The system will also replace the majority of data that is currently maintained in the Excel spreadsheets. In addition to those core functionalities the system will incorporate a lottery function that is used to randomly select students for placement in their preferred camp sessions. The application will be developed under the MIT License. The frameworks chosen for the applications web frontend are Twitter Bootstrap for the HTML, CSS, and JavaScript. Bootstrap was chosen because it's responsive design allows for a consistent experience across desktop, tablet and mobile browsers.

The MIT license was chosen in part because it is the license that the underlying framework (HUGE) is distributed under ^[5].



Figure 1.4: The MIT license is detailed in full in **Appendex A** and at the *Open Source Initiative*, at <https://opensource.org/licenses/MIT>.

1.4 Recent Developments

As the systems analysis for the project begin in the Fall 2015 semester, sufficient time was available to design and develop the requirements for the system, and create a traditional development plan. As a result, a modified waterfall methodology was chosen for the projects development.

Our development cycle is clearly delineated:

- Systems analysis done Fall 2015
- Development begins this semester

Our users understand and can clearly state the requirements. There is some degree of uncertainty in time and cost estimates. The problem we are solving is well understood. Our capstone project requires extensive documentation (project plan created Fall 2015). We wish to minimize scope creep to stay within our original requirements. We have targeted 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. For these reasons, we will be using a modified waterfall methodology, specifically the sashimi model. The phases of our project will include:

- Requirements Analysis
- Planning
- Design

- Development
- Testing



Figure 1.5: Major technologies employed in the application.

Chapter 2

System Integration, Modeling & Methodology

2.1 Web 2.0

The project is built on the fundamental principles of the Web 2.0 concept. The system is based on user interaction, dynamic web applications, and database storage for non-static user interaction. Database management is a core function of Web 2.0 development and is reflected in the development of many recent successful web applications. Two of the largest examples are Google's search platform and Amazon's product database. ^[8]

The core competencies of Web 2.0 development include:

- Services designed with cost-effective scalability in mind
- Packaged software is no longer as relevant as it was during Web 1.0
- Maintain control over databases that are unique and difficult to recreate
- Involving users in the development of the product
- Software that spans multiple devices
- Lightweight user interfaces

2.2 Application Scope

Based on interviews with the administrative users of the current Summer Engineering Academies registration management system, it has been concluded that the project will consist of an improved registration system with a web application based front-end user interface and a database for managing all relevant user, administrative, and class information. Users (Parents, guardians, etc.) will be able to register students through an intuitive to use web application. The

application will be designed with common web application and mobile (iOS and Android based applications and web sites) metaphors. The web application will then interface with the database which will store contact information and session preferences. The system administrator will then be able to view, edit, and conduct other relevant administrative tasks with the data. The system will not present users with unnecessary interfaces, redundant paths to the same information, or any other additional application and user information that is judged to be outside the needs of the end user. ^[8]

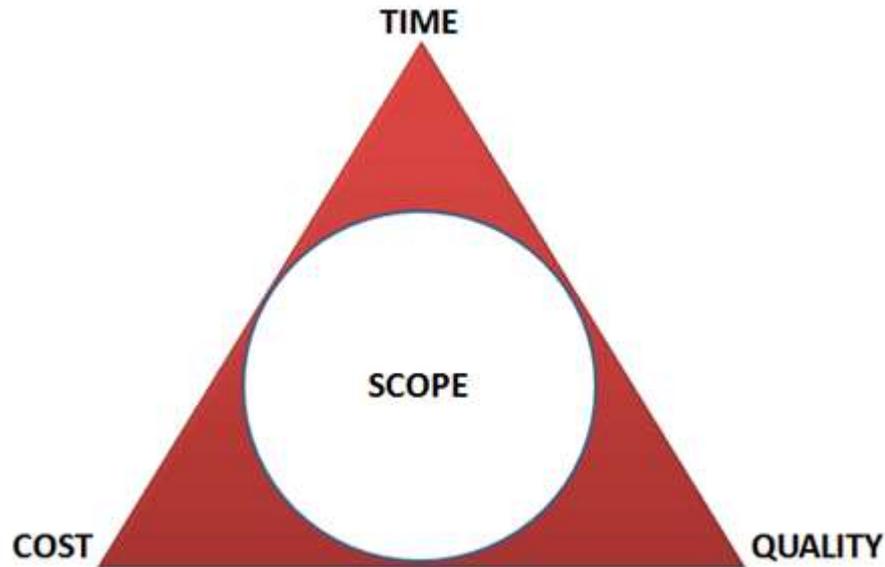


Figure 2.1: For a project to be successful it is necessary to maintain project scope. Scope cannot be expanded without sacrificing quality, increasing cost, or extending time to completion.

The system that the application and development will be operating in is an end user environment. The web application will be used by end users to access their accounts, as well as update information and signup their children for their desired classes. The selection of camp preference, registration for camps, and any changes to the student's information will be conducted through the end user web application.

Administrative tasks will be conducted through the front end web application. Administrators will have access to increased privileges that will allow them to create users, run the lottery that is used to determine student placement in camps, and manually adjust any user information that could be in error.

The registration process for the Summer Engineering Academies requires that parents or guardians fill out safety waivers through an external application, *DocuSign*, as well as pay for the classes they are accepted into through the University of Alaska webstore, provided by TouchNet.

2.3 System Requirements

The project development is based on previous systems analysis which has identified the following system requirements:

Functional Requirements:

- Create user login system
- Enable parents to submit camp applications
- Enable parents to modify camp applications
- Enable parents to view status of camp applications
- Enable parents to respond to camp application acceptance
- Integrate with **DocuSign** to track liability and media forms
- Integrate with **TouchNet** to track payments
- Enable administrators to manipulate system information
- Enable administrators to input and manage paper applications
- Export data needed for mail merge actions
- Enable administrators to assign campers to camps and waitlists
- Identify anomalous camper information
- Provide instructors with needed email lists

Non-Functional Requirements:

- Communication will be encrypted
- The system will be easily maintainable
- Regular backups will occur automatically
- The system will be easily accessible
- Private information will be kept secure

The project has the following projected key milestones:

Project Milestones	Target Date
Project Approved for Implementation	12/11/15
Project Kickoff	01/16/16
Solidify Requirements	01/21/16
Requirements Signoff	01/31/16
ERD Approved	02/13/16
Database Completed	03/01/16
Web Application Completed	04/01/16
Testing Completed	04/20/16
Faculty Walkthrough	04/23/16
Final Presentation	04/30/16
Project Completion Signoff	04/30/16

Table 2.1: Projected dates for the projects major milestones.

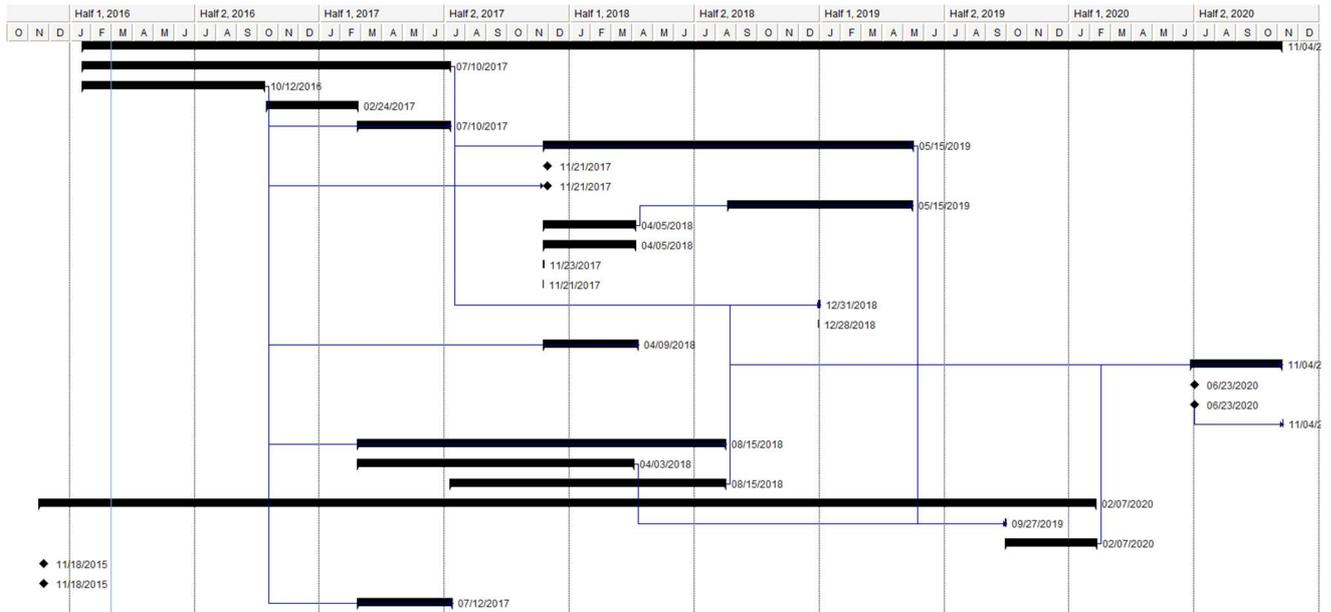


Figure 2.2: A simplified GANTT chart showing the projects task scheduling.

2.3 Components

The application is being developed with a wide range of languages, tools, and technology. The primary languages and markup that the application will utilize are:

- PHP 5.6
- HTML 5
- CSS 3
- jQuery 2.x
- AngularJS 1.x
- JavaScript ECMA 6
- SQL

Scripting languages such as PHP, Python, and Ruby play a large role in the development of advanced web applications. These languages have been called “The duct tape of the Internet.” [8]

A development server will be provided by University of Alaska Anchorage’s Information Technology Services.



Bringing MySQL to the web

Figure 2.3: *phpMyAdmin* is the database administration tool at the heart of most AMP (Apache, MySQL, PHP) stacks. It has been used throughout the project to manage the MySQL database that is essential to the successful operation.

2.4 Provide a Rich User Experience

As early as 1992, the web has been used to deliver “applets”, or interactive content that is framed within a webpage. When Java was introduced in 1995 it was framed around the delivery of these dynamic experience for web. The introduction of JavaScript allowed for user interaction dynamically within the HTML on the page the user is viewing.

The potential of the web to deliver full scale applications was fully realized when Google introduced their revolutionary email client, Gmail. Web applications with rich, dynamic user interaction followed. [7]

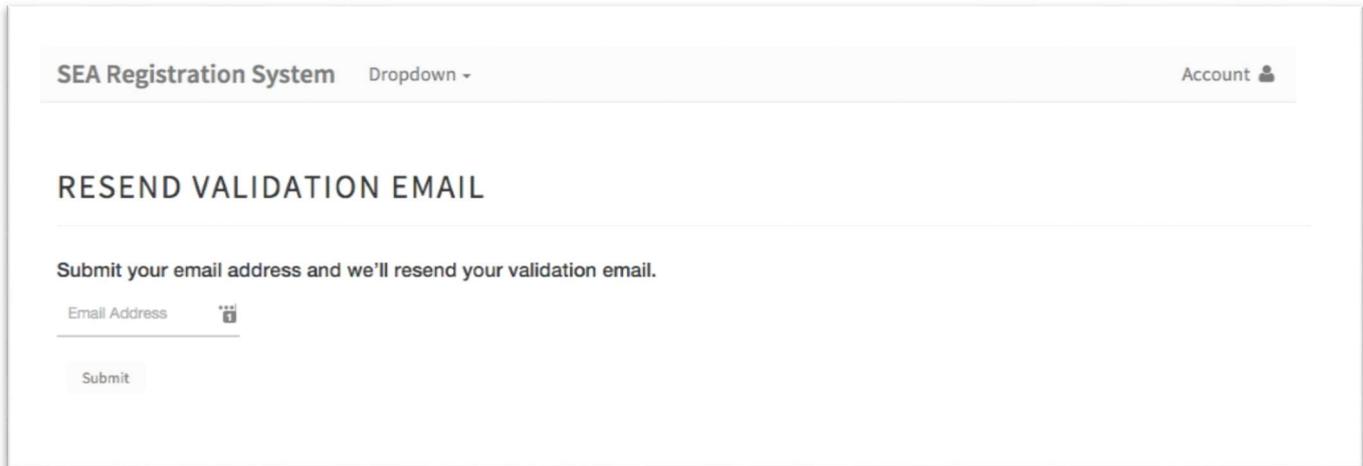


Figure 2.4: Early Web Application page showing design and styling of a page to resend account activation and validation email.

2.6 The Agile Coding Methodology

The agile coding methodology is a people oriented, results focused development methodology. It focuses on results. It’s core concept is the regular delivery of working software. Agile focuses on harnessing the strengths of its team members. Members of the team share multiple roles and often shift positions when their assigned task has been completed but other tasks essential to the current deliverable remain. It promotes adaptability to changing circumstances.

<p>Strengths</p> <ul style="list-style-type: none"> PHP Programming Database Design Server Configuration Java Programming C/C++ Programming HTML 5/CSS 3 JavaScript ASP.NET 	<p>Strengths</p> <ul style="list-style-type: none"> Web Design HTML 5/CSS 3 ASP.NET Database Design Microsoft Office Microsoft Project
<p>Weaknesses</p> <ul style="list-style-type: none"> User Interface Design 	<p>Weaknesses</p> <ul style="list-style-type: none"> PHP Programming

Figure 2.5: A teams overlapping skill set allows for each developer to assist with work for the other should it become necessary.

Chapter 3

Design, Testing, and User Interface

3.1 User Interface

The user interface for the application is a HTML5, CSS3, and JavaScript based web application. It is built with a responsive, mobile first methodology. This interface was chosen due to the target demographic of the project and the ease of interfacing a web based application with a server side database.

Front end

For the applications front end many of the popular modern web technologies were implemented. The applications HTML was developed using Twitter Bootstrap 3. Bootstrap is the industry standard for responsive mobile-compatible web sites and applications. A responsive website is a web site that is designed to be viewable on multiple devices, and at multiple resolutions, while retaining a consistent appearance and maintaining usability. Twitter Bootstrap was chosen because it allows for easy integration with existing technologies. Most users who use a mobile device regularly should be able to quickly identify the actions they wish to take.

Often when a website or application that was design to be viewed in a desktop environment will become unusable when it is viewed from a tablet or smartphone. Due to the significantly different screen to resolution ratios of those devices.



Figure 3.1: The navigation bar for the web application.

For example, a typical desktop monitor is between 19”-21” with a display resolution of 1920 pixels wide by 1080 pixels tall (often referred to as ‘1080p’, or ‘full HD’). This provides a ratio of the number of pixels to the physical size of the display. This is referred to as Pixels Per Inch, or PPI. PPI is often incorrectly referred to as Dots Per Inch, or DPI, which is a similar

measurement of resolution but for print media. A 19" display with a resolution of 1920px by 1080px would yield a PPI of 154.

A modern smartphone with a resolution of 1920px by 1080px and a display size of 5 inches would yield a PPI of 440.

When a page designed for a large desktop (which has a lower PPI) is displayed on a much smaller phone (which has a higher PPI) of the same or similar resolution the content appears much smaller than intended.

A responsive design is one that adapts its layout to the PPI of the device it is being displayed on. This is most commonly achieved by adjusting the Cascading Style Sheets (CSS) to display the content in an easier to understand format. Another technique is the use of columns in the layout of the webpage. The websites content is then placed within the columns, and the columns reposition themselves based on the windows size and physical size of the device it is being

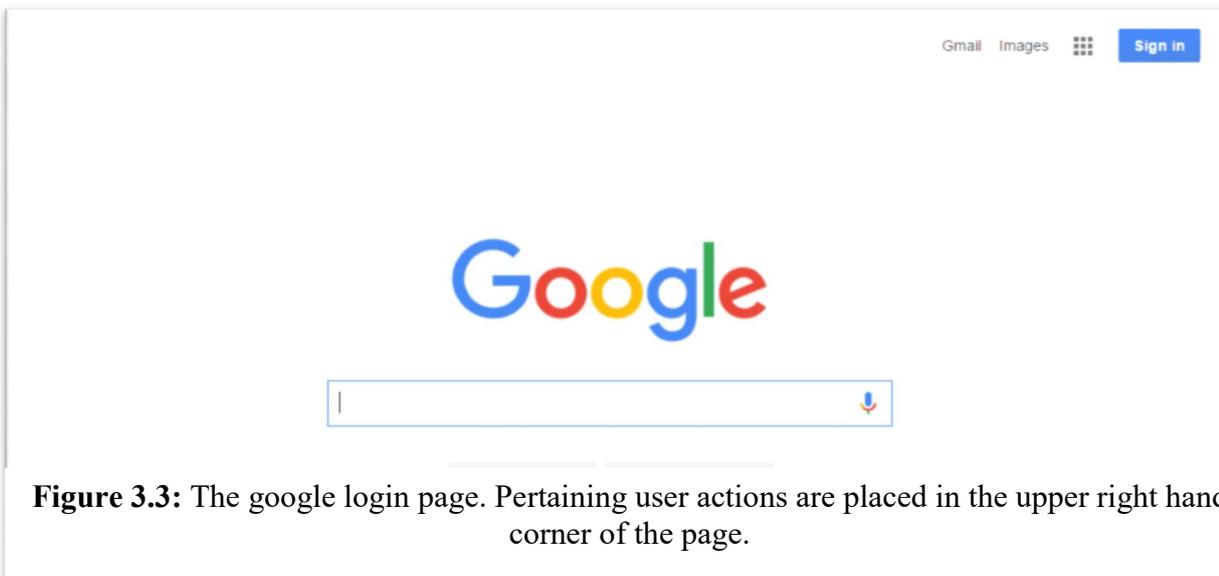


Figure 3.3: The google login page. Pertaining user actions are placed in the upper right hand corner of the page.

displayed on. Bootstrap employs the column method.

In addition to the responsive design of Bootstrap, it also includes custom CSS, JavaScript, and jQuery libraries. These libraries provide the HTML content on the page with a more complex appearance and advanced dynamic user interaction.

In the application I employed the columns to maintain a consistent appearance across devices. The CSS and JavaScript components were used to provide advanced user interaction elements, such as a dynamic navigation bar that allows users to login, manage their account, change their password, etc. When an administrator is logged in then they are provided a navigation bar with actions and pages that require their elevated level of permissions. The web application is intuitive to navigate due to its use of popular user interface design metaphors. A navigation bar is placed at the heading of the page, that allows users to navigate to the desired action or information.

User account access is located in the top right corner of the navigation bar. This allows users to access their account information, change their password, and logout of their account. This is similar in design to many popular web based applications, such as Google's Gmail. It should be intuitive to users and immediately recognizable.

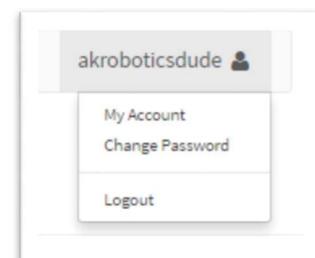


Figure 3.2: The user account icon and dropdown menu.

AngularJS

One of the most advanced frameworks available for building web applications is Google's AngularJS. AngularJS is a collection a very powerful JavaScript functions. It can be used to provide additional functionality to an existing web page, or it be used to create entire websites that utilize Single Page Application designs. For this project I used AngularJS to provide dynamic filtering updating of various administrative pages on the application. With the features I implemented users can select the search box on the page, and the table of data they are searching is immediately updated with matching results, as they type each character.

If a similar search were to be implemented using HTML and JavaScript, results would normally only be displayed after submitting the search string. The dynamic approach provides information to the user quicker.

AngularJS was also used to display dynamic data provided to the client side of the application. Through various AngularJS Directives, elements can be updated and visibility adjusted. For search results an element that displays the number of results that match the input string was dynamically hidden or displayed based on the length of the input. The number of results was used to adjust the class of the element, allowing the element to display a red warning when no results were found.

JavaScript was used to provide dynamic forms validation. jQuery was used to check inputs on forms against their validation pattern. As the user moves between fields on the form the validator checks the input, and displays a warning if it is invalid. Without this scripting the errors would only be displayed after the form is submitted, often meaning the user would lose all he data they had type into the form and be forced to reenter it.

CSS

CSS was used to provide custom appearance to elements in the application. It was also used to adjust elements for proper displace and a variety of effects. As the application is designed to serve as an administrate tool, CSS and JavaScript effects that are often used in commercial client-facing websites, were not used here. As with any program the more code that is added, and the more calculations that must be made (such as graphically effects, which are often high-CPU processes) the lower the performance. It was decided that the trade off in performance of the system was not worth the cost in performance and usability.

Whenever possible the design attempted to follow current popular design concepts. This refers to the placement of certain primary action buttons, actions that occur in response to user input, consistent design and placement throughout the application.

The user account icon is an example. In many of the current popular web applications, when a user's name, email, or username is displayed on a page the user expects to be able to accomplish actions related to their account. This is commonly placed in the upper right corner of the page. Anticipating that users will first go there in an attempt to accomplish their desired actions provides for a better user experience.

When designing other portions of the web application user intent were more difficult to anticipate. Actions that are logically placed to the developer may be unknown to the end user. The students page that displays all student accounts to the administrator is an example. Design

considerations included: Where does the user expect to have to click in order to access the student information? What about the student's parent? Is clicking on the user's ID number a logical place for the action to view the student? Should selecting the sessions the student is in take the user to those sessions, or a list of the current student's session? These are some of the questions we had to answer to design and develop an intuitive user interface.

3.2 Software Development Techniques and Tools

The project will implement the following quality assurance techniques:

- PHPunit, which will allow the use of custom written test cases
- JSHint, a popular JavaScript linter
- W3C HTML linter and validation

A linter is a code analysis tool that examines the input for syntactical errors as well as style guides and good coding practices.

Software development techniques and tools.

User acceptance testing is testing that is conducted with project sponsors and key members of the development team to insure that the product is meeting their expectations. A significant portion of the project is made up of the administrative tasks that the sponsor currently carries out. As a result, this project is will make use of extensive user acceptance testing to insure that it is meeting the expectations of the sponsor.

After the project is migrated to the production environment, further testing will be conducted to insure compatibility and functionality.

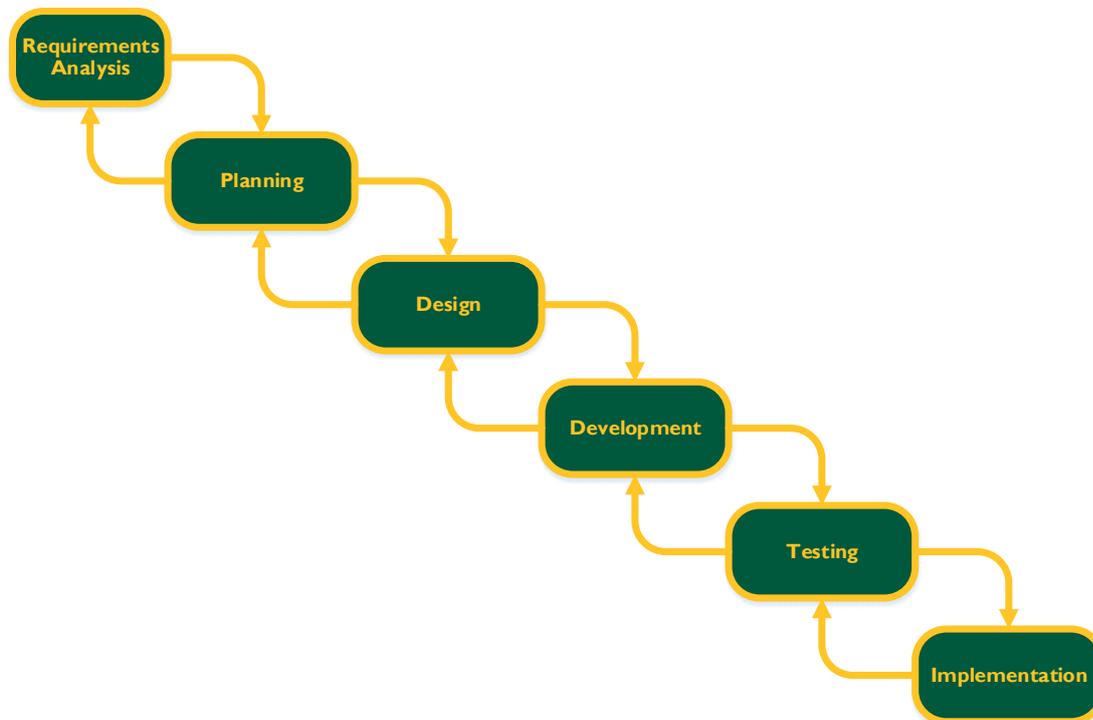


Figure 3.4: The development methodology cycle.

The testing methodology is based on unit testing on the database and framework. User acceptance testing will be used before delivery of the product to the customer.

3.3 Github Update

Git based version control allows for developers to work on their assigned tasks on a branch of the code base, without affecting the other team members code or the master code base. This has allowed for continuous development of the system even when other developers are having difficulties with their own development system.

The project is hosted on a BitBucket git repository with multiple branches for development, feature creation, and live deployment to the test server.

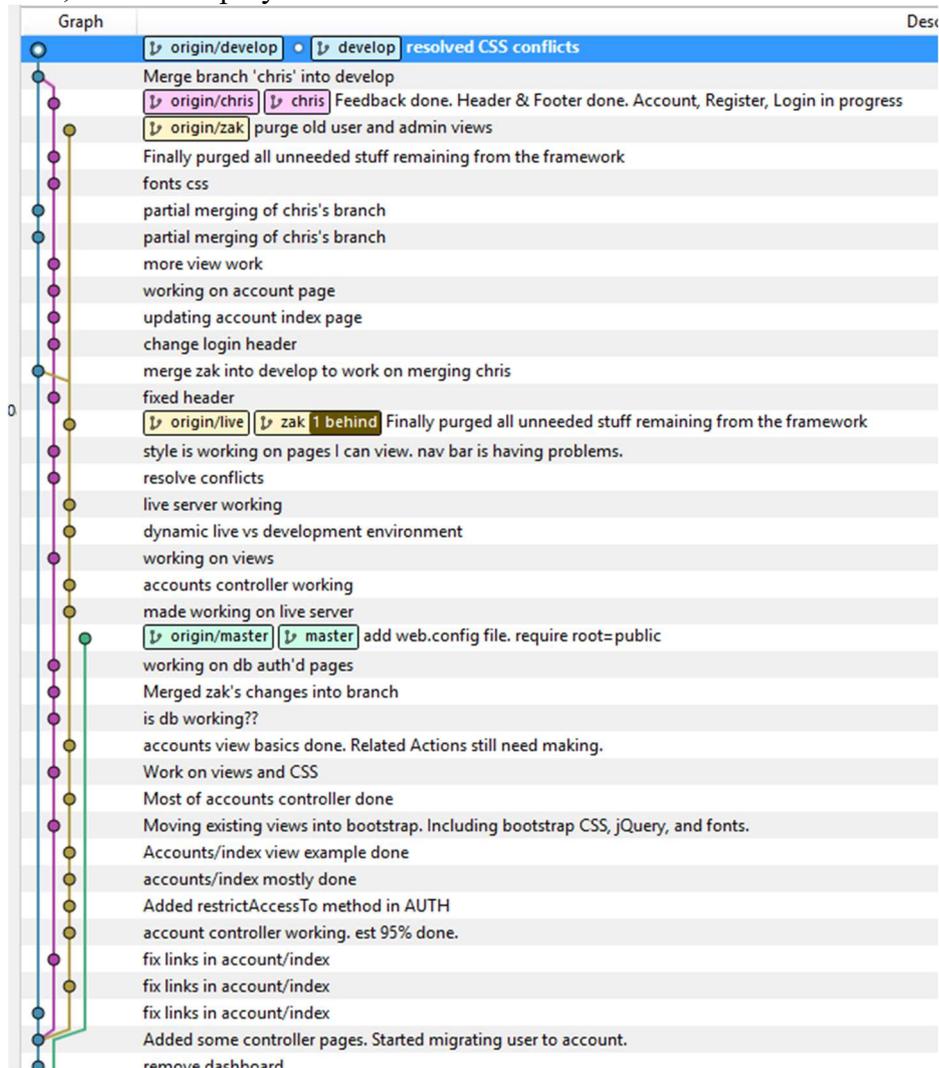


Figure 3.5: Git based version control has allowed for rapid development without negative effects on team members individual work.

3.4 Agile Methodology

The agile coding methodology is a people oriented, results focused development methodology. It focuses on results. It's core concept is the regular delivery of working software. Agile focuses on harnessing the strengths of its team members. Members of the team share multiple roles and often shift positions when their assigned task has been completed but other tasks essential to the current deliverable remain. [8]

Agile as a project management methodology takes the core concepts of the Agile development methodology and applies them to a broader project management scope. It focuses on delivering results to the sponsor often. It emphasizes a less structured overall plan to allow for adapting to changing needs of the sponsor and situation. [14]

3.5 Documentation

Documentation will be created for the database, php infrastructure, and user interface. This documentation is currently in progress and we have very little to show. Documentation will include user quick guides that demonstrate how to become familiarly with the system. A detailed user manual will be created using the existing documentation as well as lessons learned during development. The complete user manual will be provided with the final version of this report.

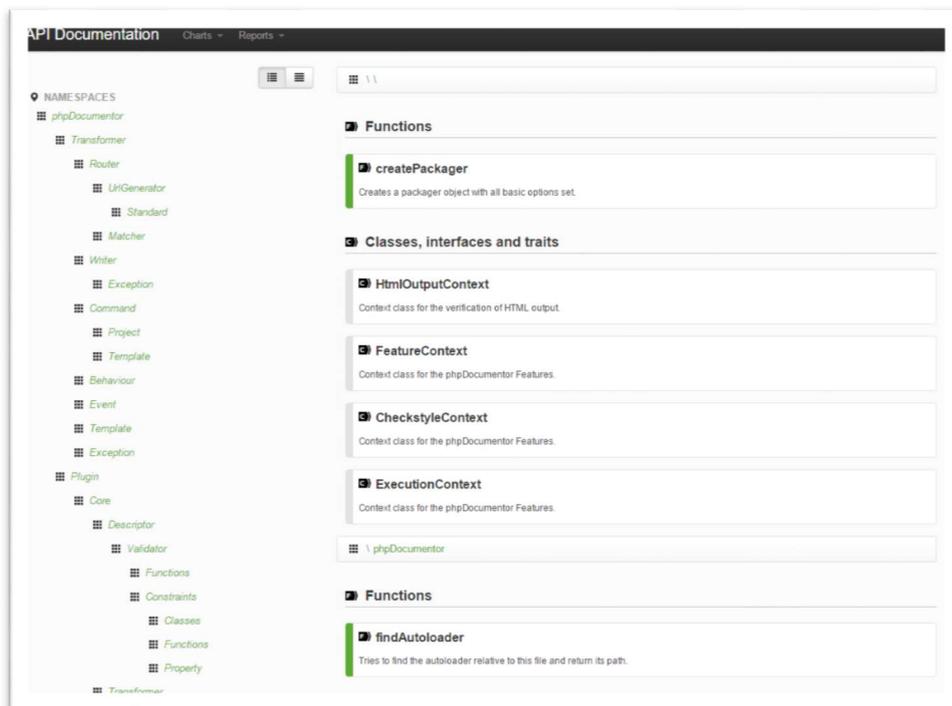


Figure 3.6: A sample of class and tree based documentation generated from PHP code.

Chapter 4

User Manual

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.

Icons Used in this User Manual

Important information contained within this user manual is communicated through the use of visual icons. The various icons are as follows:

	Important information
	Good to know - Tricks
	Risk or Warning
	Mandatory action
	Actions reserved for the administrator
	Actions reserved for parent, staff, or instructor users.

System Overview

Description

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.

Project Stakeholders

Project Manager	Johannes Meyer jcmeyer2@alaska.edu (907) 268-0143
Back End Developer	Zakary Stone, Developer zrstone@alaska.edu (907) 317-8216
Front End Developer	Christopher Brooks, Developer cmbrooks@alaska.edu (907) 854-6717
Project Sponsor	Dr. Scott Hamel, Director University of Alaska Anchorage Summer Engineering Academies sehamel@uaa.alaska.edu (907) 786-1070
Faculty Advisor	Dr. Dennis Drinka, Professor University of Alaska Anchorage College of Business and Public Policy drinka@uaa.alaska.edu (907) 786-4120

Support

For support during the development phase please contact the development team by email, telephone, or social media. After the final application has been released, the developers may be unable to provide assistance depending on the issue. UAA's Information Technology

Services will be hosting the application after the final release, and should be able to answer most queries.

	During Development:	After Development
Name	Johannes Meyer	UAA IT Services
Email	jcmeyer2@alaska.edu	callcenter@uaa.alaska.edu
Phone	(907) 268-0143	(907) 786-1800
Twitter	@johannesmeyerak	@uaanchorage

Referenced Documentation

For more information about the Summer Engineering Academies, and the UAA Engineering department in general, please go to <http://www.uaa.alaska.edu/collegeofengineering/summeracademies.cfm> .

History of the Solution

While attendance at the Summer Engineering Academies has continued to grow, more sessions, instructors, and staff have been added to handle the increased numbers. The current system of using excel spread sheets and visual basic scripts to store and sort data is quickly becoming overly complex. The administrators continue to run into issues relating to the deficiencies of storing hundreds of records in “flat” files. Small errors can be difficult to fix, and, if the current administrator were to move on, their replacement would be hard pressed to understand how the system functions. Further, Parents can get confused during the many phases of the registration, application, and lottery processes, leading to administrator time being taken up with otherwise avoidable customer service issues.

The SEA Registration System is being created to help parents easily register for the Summer Engineering Academies at UAA, as well as for Administrators to have a simple interface in which to conduct Session placement and other administrative tasks. This Registration System serves as a way for users to sign up participants for the various programs, and for the system administrator to perform typical business procedures.

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.

Project Milestones	Target Date
Project Approved for Implementation	12/11/15
Project Kickoff	01/16/16
Solidify Requirements	01/21/16
Requirements Signoff	01/31/16
ERD Approved	02/13/16
Database Completed	03/01/16
Webapp Completed	04/01/16

Testing Completed	04/20/16
Faculty Walkthrough	04/23/16
Final Presentation	04/30/16
Project Completion Signoff	04/30/16

The project has progressed through development and will be entering the testing phase shortly.

Connecting to the SEA Registration Application

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



Although most modern browsers are acceptable, the developers suggest accessing the SEA Registration System using an updated Chrome browser.

Web Address

Connecting to the Summer Engineering Academy will vary depending on which project phase the connection is occurring in. After the web application has been completed and moved from development servers to the production server, it will be accessed through the Summer Engineering Academies homepage contained within the wider University of Alaska Anchorage Web Site as shown by the following address:

https://www.uaa.alaska.edu/collegeofengineering/sea_application.cfm

Until the application is completed, and while in development, it can be accessed through the development server by going to the following address:



 *The application URL may change after final implementation. Contact support for more information.*

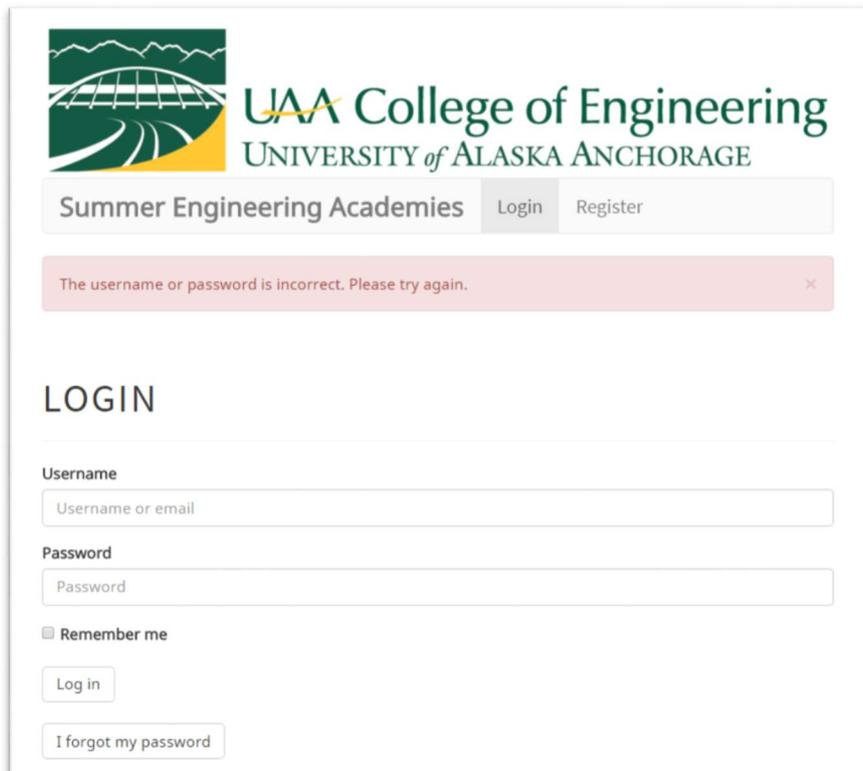
Authentication

There are several different account levels that have access to the SEA Registration System. Users are authenticated by providing a username and password at the login page. Sample parent and administrator accounts have been provided for testing purposes. Please inform the Project Steering Committee when the links are no longer needed, so they can be removed for security purposes.

Username:	admin	akroboticsdud e
Password:	Pa\$\$word	123456



 A username and Password must be entered to progress beyond the Login page.



 Incorrect Authentication details have been entered

You have typed in a wrong password 3 or more times already. Please wait 30 seconds to try again. ×

 *Incorrect Authentication details have been entered 3 or more times, and the user must now wait 30 seconds to try again.*

Structuring of the SEA Registration Application

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.

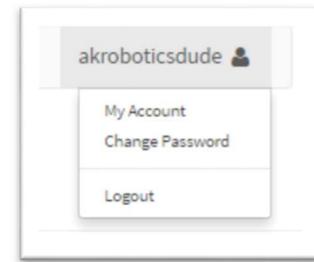
Username

Dates: Through:

Menus and Navigation

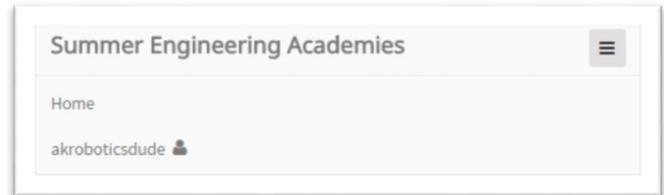
Computer Browsers

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 System from smaller devices such as smart phones, the menus appear as typical “hamburger” icons that, when pressed, expand to include the menu options:

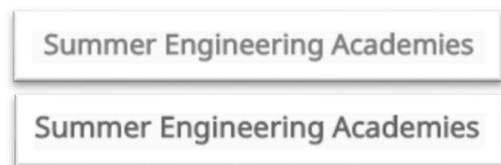


After the menu is expanded by clicking on the username, further options are displayed below the username. Throughout the application, this method of opening menus can be used to quickly navigate from page to page without repeatedly zooming in and zooming out when using a smaller device. The SEA Registration System was developed to reach the widest audience possible, regardless of their preferred device.



Shortcuts

From wherever you are in the application, the home page can be reached by clicking on the “Summer Engineering Academies” header below the banner image.



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

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.

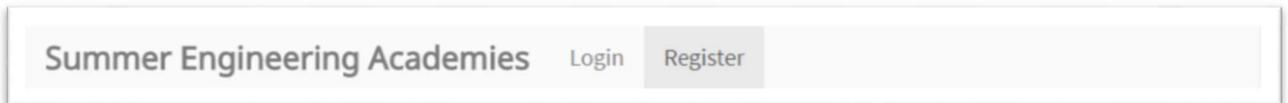
Login

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

Email

Repeat Email

Password

Repeat Password

First Name

Last Name

Phone Number



Please enter above characters

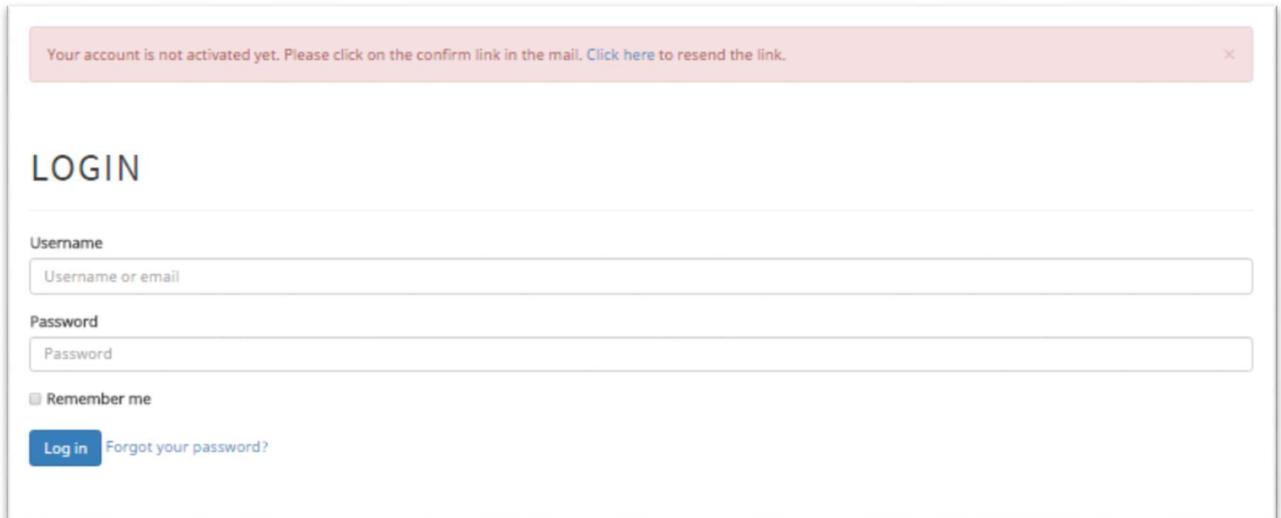
[Reload Captcha](#)

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

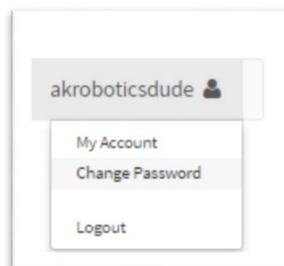
Submit your email address and we'll resend your validation email.

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: akroboticsdude Save Username

Email: akroboticsdude@johannescmeyer.com Save Email

First Name: Ima Save First Name

Last Name: Testuser Save Last Name

Phone Number: 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:

Birth day:

Male Female

Grade Level (as of fall 2016):

Emergency Contact

Name:

Phone Number:

Relation:

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:

None
RB-M1 Jun 06 to Jun 10 Energy
EG-M1 Jun 06 to Jun 10 Alternative Engery
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:

Session Preference 6:

Session Preference 7:

Session Preference 8:

Session Preference 9:

Session Preference 10:

Session Preference 11:

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: <small>(as of fall 2016)</small>	<input type="text" value="8"/>	<input type="button" value="Save Grade Level"/>

Emergency Contact

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

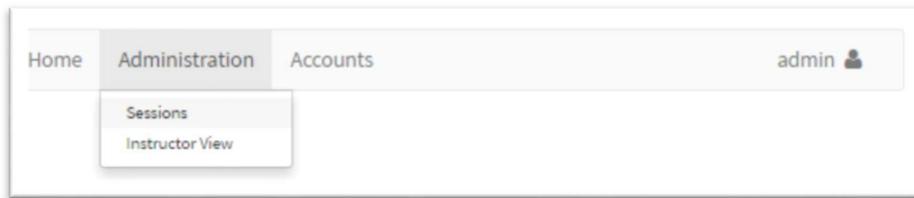
Logging in as an admin user will present the “Admin” page.

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

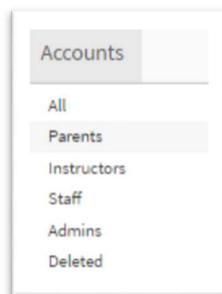
Currently this view presents the administrator with all of the Instructors and their assigned sessions. In future releases, administrators will be able to sort by the column headings to easily find needed details. This is also where Instructors will be able to view all of their assigned sessions, along 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](#)

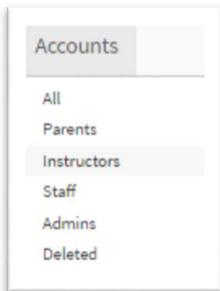
parent

[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

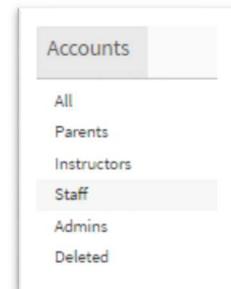
Id	Username	Email	Activated ?	First Name	Last Name	Phone Number	
8	8instructor	elaine@johannescmeyer.com	No	Elaine	Todd	3335552222	View/Edit
9	9instructor	kupilik@johannescmeyer.com	No	Matt	Kupilik	6643332222	View/Edit
10	10instructor	peng@johannescmeyer.com	No	Jifeng	Peng	9073332222	View/Edit
11	11instructor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

 *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

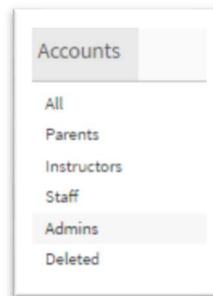
Create new staff

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

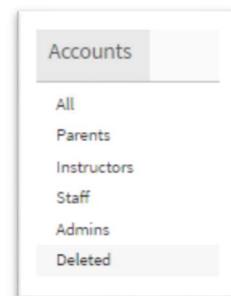
Create new admin

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:

Username:

Email:

First Name:

Last Name:

Phone Number:

New password:

Repeat new password:

Change Account Status: Undo Soft Delete

Save Account Type

Save Username

Save Email

Save First Name

Save Last Name

Save Phone Number

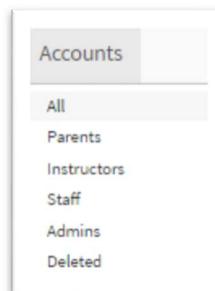
Submit new password



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 whether because a user is acting maliciously,

deactivate accounts, 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.

The screenshot shows a form with three main sections. The first section has two input fields: "new password:" and "Repeat new password:". To the right of these fields is a blue button labeled "Submit new password". The second section is labeled "Change Account Status:" and contains a red button labeled "Soft Delete".



"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.

The screenshot shows the "USER ACCOUNT INFORMATION" form. The form includes fields for User ID (4), Account type (instructor), Username (4instructor), Email (test@johar), First Name (Johannes), Last Name (Mayers), and Phone Number (907268014). There are also fields for New password and Repeat new password. A "Change Account Status:" section has a "Soft Delete" button. To the right of the form are several blue buttons: "Save Account Type", "Save Username", "Save Email", "Save First Name", "Save Last Name", "Save Phone Number", and "Submit new password". A white dialog box is overlaid on the form, titled "searegistration.uaa.alaska.edu says:" and contains the text "Are you sure you want to delete this account?". The dialog box has "OK" and "Cancel" buttons.

Features Planned for April 9th Beta Release

While our original project plan specified the 20th of April as the end of the testing phase, we have condensed the schedule with the aim of releasing the full Beta version of the SEA Registration System in time for the AITP NCC Final Web Competition presentations on April 9th. Many of the features absent from this version due to not yet being fully functioning will be finished and added after the 5 day Web Site freeze. These include:

Instructor Views

Instructor users will be able to view their assigned sessions, as well as attendance sheets containing names, ages, gender, emergency contact, and allergies for the participants in each session they instruct.

Staff Views

Senior staff will be able to see all of the information presented to instructors, sortable by any of the instructors or sessions.

Data Filters

Administrators will be able to filter participant data by:

- first name
- last name
- grade level
- id number
- session selection
- waitlist position
- application status
 - application unfinished
 - application finished
- registration status:
 - payment completed
 - payment not completed
 - release form signed
 - release form not signed

This feature will be incredibly important when the application goes live and hundreds of records are present.

Tuition Waivers

The system will both record and provide waivers for the session fees. Project Manager is currently discussing options with the Project Sponsor.

Lottery Process

The next release will include various lottery functions, including:

- Assigning session placement numbers via an existing lottery algorithm
- Assign participants to sessions in order of preference according to their placement number
- Place participants on waitlists as sessions fill up
- Tools to move waitlist positions up and down as participants decline sessions

Secondary Lottery

After the first round of session placement, parents will be notified of which sessions and/or waiting lists their participants were placed, and have a chance to accept sessions and waitlists, or change preferences for a second round of session placement.

Administrator Overrides

Administrators will have the ability to override the results of all functions contained within the set of lottery and secondary lottery tools.

Import Payment Data

The system will have the ability to import comma separated files (CSVs) in order to enter payment information from TouchNet. Various Application Programming Interfaces (APIs) are being researched that may import directly from TouchNet. At a minimum, the system will accept CSVs, which administrators can generate quickly and easily from TouchNet.

Export Printable Lists

Administrators will be able to generate attendance sheets via a data export tool.

Chapter 5

Summary and Conclusion

5.1 Summary

This project began with extremely ambitious goals. We were building a product with commercial level scope, in a single semester. When the additional responsibility of other classes, work, and family are taken into account this left very little time to devote the level of attention this project demands.

We were however, surprisingly successful given the previously stated constraints. We have completed the database, and the framework for the majority of the system is complete. All currently functioning controllers have user facing webpages. The necessary functions for the creation of user accounts are present and working. Sessions can be added and administrated. User accounts can add students and set session preferences based off of their information and the requirements of the selected session.



Figure 5.1: The landing page of the current live version of the webapp.

5.2 Implications

The long term implications of this project have the potential to make a massive impact in how the Summer Engineering Academies staff administrate their program. The successful implementation of the project will save the staff significant work hours. It could save the University thousands of dollars in licensing fees if they grow to the point of needed to implement an existing off the shelf project. Finally, it will allow the current administrators to spend more time making the Summer Engineering Academies a better program, and less time on reparative administrative tasks.

5.3 Recommendations and Improvements

Communication is key. Perhaps my biggest take away from this project is that a lack of communication can quickly stall a project, especially one with such a short development period as ours. During the project we met several times a week and communicated via email and Slack. However, confusion or miscommunication about small details still developed. The misunderstanding, while unintentional, still lead to duplicated, lost, or ultimately unnecessary work.

Understanding the client's needs and not inserting your own into the development process is also important. Essential development time was spent on features that the team thought was required, but in fact were not wanted by the sponsor.

We aimed extremely high in the scope of our project. As previously mentioned we set out to develop a commercial grade application for immediate use by the University. This is not a small task, especially for two undergraduate Computer Science and Engineering students. The development of this project required quickly learning new languages and implementations, as well as rapid prototyping, debugging, and deployment.

While we completed the majority of the core requirements for this application to be used in its intended application, more time is required to properly finish the remaining requirements. Code needs to be refactored for consistent style across the application. Accessibility requirements needs to be confirmed and thoroughly tested. More extensive white-box testing needs to be conducted to test for the bugs that we did not anticipate.



← View all participants

↑ View parent

✎ Edit Information

SESSION OPTIONS FOR EXAMPLE TEST

Session preferences for Example Test

Session preference 1:
SD-H5 Jul 11 to Jul 15 Advanced Structures

Session preference 2:
EG-H2 Jun 13 to Jun 17 Alternative Engery

Session preference 3:
RB-H4 Jun 27 to Jul 01 Robotics

Session preference 4:
RB-H5 Jul 11 to Jul 15 Arduino Robotics (MatSu)

Are you interested in more than one session if available?

Save Preferences

Session exceptions for Example Test

Add exception

Select Session

Add Exception

Figure 5.2: The depth of the systems abilities is shown here in an administrator page for editing a student's session preferences.

5.4 Conclusion

Working on the Summer Engineering Academies registration has been an insightful learning experience. It required me to combine the skills I have learned while at UAA as well as learn new ones as the project required them. It taught me valuable skills about working with a client and helping them to understand how the available technologies could assist them in solving the problems they are experiencing. It gave me the experience of working with a team on a large scale, industry relevant project. Finally, while we accomplished much of what we set out to do, I encourage future students to help the project realize its full potential.

References

- [1] Avraham Leff, J. T. (kein Datum). Web-Application Development Using the Model View Controller Design Pattern. IBM T. J. Watson Research Center.
- [2] FRATERNALI, P. (1999). Tools and Approaches for Developing Data-Intensive Web Applications: A Survey. ACM Computer Surveys.
- [3] Leff, A. and Rayfield, J.T. (2008) ‘WebRB: A different way to write web applications’, IEEE Internet Computing, 12(3), pp. 52–61. doi: 10.1109/mic.2008.69.
- [4] UAA Summer Engineering Academies, <http://www.ktuu.com/news/news/UAA-s-Summer-Engineering-Academies/26150066>
- [5] panique (2015) *Panique/huge*. Available at: <https://github.com/panique/huge> (Accessed: 27 January 2016).
- [6] El-Abbassy, A. and El-Zewedy, M. (2013) ‘ICONIX approach to MVC: Applying robustness analysis on the model - view - controller architecture’, Journal of ACS : Advances in Computer Science, 7, pp. 37–59. doi: 10.12816/0011486.
- [7] Musser, J., O’Reilly, T., Team, the O.R., Oreilly, T. and Team, O.R. (2006) *Web 2.0: Principles and best practices*. Sebastopol, CA: O’Reilly Radar
- [8] Shore, J., Warden, S. and Chromatic (2007) *The art of agile development*. United States: O’Reilly Media, Inc, USA.
- [9] *Hosted libraries* (2016) Available at: <https://developers.google.com/speed/libraries/> (Accessed: 10 March 2016).
- [10] Q. Inc, Reserved, A.R., Inc, C. 2014 Q. (2015) *Rich Internet applications: The next frontier of corporate development*. Available at: <http://www.eweek.com/c/a/Security/Rich-Internet-Applications-The-Next-Frontier-of-Corporate-Development-732651> (Accessed: 19 March 2016).
- [11] Flanagan, D. (2011) *JavaScript: The definitive guide: Activate your web pages*. 6th edn. United States: O’Reilly Media, Inc, USA.
- [12] *JavaScript: The definitive guide (third edition)* (1998) *Computers & Mathematics with Applications*, 36(8), p. 126. doi: 10.1016/s0898-1221(98)91177-4.
- [13] Leff, A. and Rayfield, J.T. (2008) ‘WebRB: A different way to write web applications’, IEEE Internet Computing, 12(3), pp. 52–61. doi: 10.1109/mic.2008.69.
- [14] Mundra, S. (2016) *Agile project management*. Available at: <https://www.thoughtworks.com/agile-project-management> (Accessed: 1 April 2016).
- [15] Otto, M., Thornton, J. and contributors, B. (no date) *Bootstrap*. Available at: <http://getbootstrap.com/> (Accessed: 1 April 2016).

Appendix A

Diagram A1: Entity Relationship Diagram

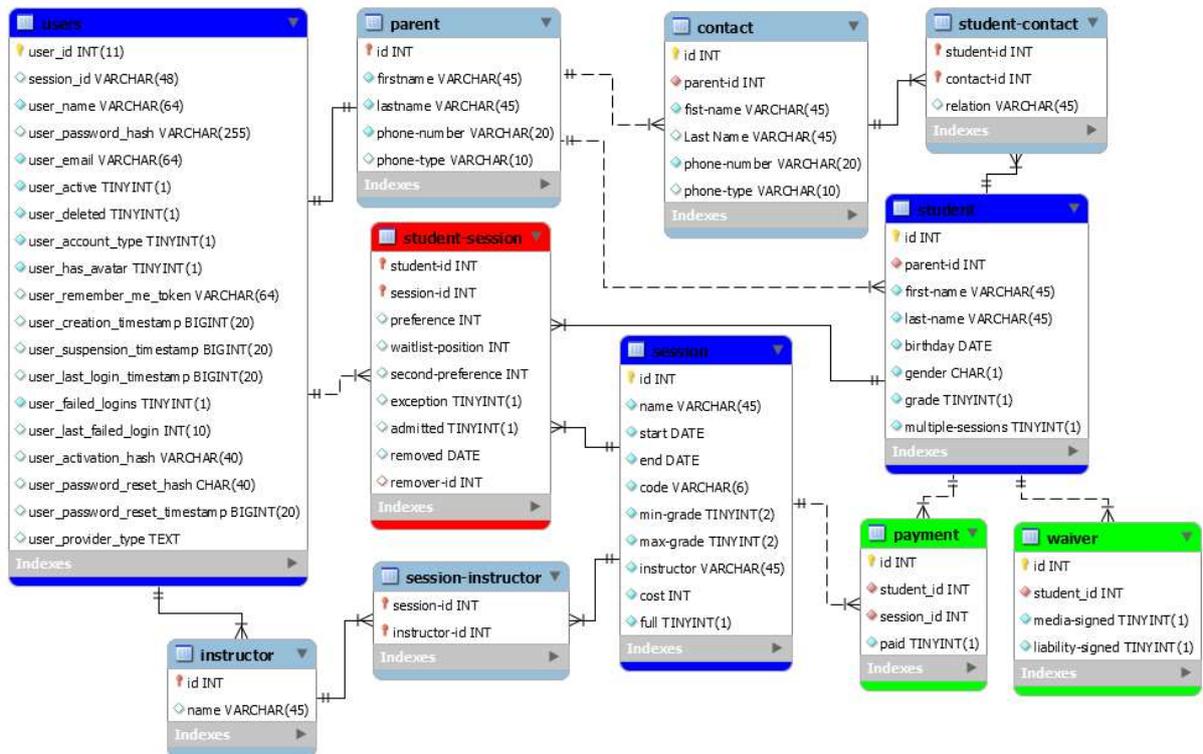


Diagram A2: A work breakdown structre for hte project, intersected with project deliverables.

	Web Application							Database		Documentation				Project Report									
	Account Creation	Application Management	Signup System	Registration Tools	Internal User Tools	Account Management	Lottery System	Email List Controls	Application Review Tools	Paper Application Function	Data Export System	Tables	Stored Procedures	User Manual	Maintenance Support	System Design Specifications	Database Design Specifications	Entity Relationship Diagram	Schema Documentation	Stored Procedures Documentation	Project Management Documentation	Client Acceptance Documents	Design Documentation
Web Application																							
Design Web Application																							
Identify User Requirements																							
Interview Users	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓		✓	✓	✓
Draft Requirements Document																					✓	✓	✓
Review Requirements Document																					✓	✓	
Design User Experience																							
Build User Interface Mockups	✓	✓	✓		✓																		
Review User Interface Mockups with Users	✓	✓	✓		✓																		
Design Diagrams																							
Create GANTT Chart																						✓	✓
Create Business Process Model																						✓	✓
Create Network Diagram																						✓	✓
Develop Wireframe Diagram																						✓	✓
Review Diagrams with Users																						✓	✓
Develop Web Application																							
Implement Lottery System								✓															
Develop Student Application management system																							
Create Student Application Submission Tools		✓	✓																				
Create Session Confirmation																							
Integrate Redirection to Payment Website							✓																
Integrate Redirection to eDocument Signing Website							✓																
Develop Student Registration Management Tools			✓																				
Develop Login System																							
Create Login page							✓																
Create Signup page		✓																					
Develop Permissions System																							
Create Data Export Utility										✓													
Build Account Management Tools																							
Implement Duplicate Data Detection Tool							✓																
Implement Password Change Tool							✓																
Implement Student Data Filters																							
Build Email Filter tools																							
Develop Application Review System																							
Develop Paper Application System																							
Create Paper Application Entry Tools										✓													
Create Paper Application Management tools										✓													
Test Web App																							
Develop Tests																							
Develop Unit Tests																							✓
Develop System Tests																							✓
Develop Integration Tests																							✓
Conduct Tests																							
Conduct Unit Testing																							✓
Conduct User Acceptance Test																							✓
Conduct Integration Tests																							✓
Deploy Web Application																							
Deploy Application to Production Server		✓	✓		✓	✓																	
Deploy Database to Production Server											✓	✓											
Conduct User Training																							

Appendix B

Image B1: Git Repository (hosted on BitBucket) of initial application source files.

Christopher Brooks / SEA-Registration

Source

master | SEA-Registration /

- account
- css
- error
- fonts
- home
- images
- js
- login
- register

 .gitignore	0 B	7 hours ago	continue with wireframes
 blog.html	3.1 KB	2016-01-21	added a bunch of template files
 dashboard.html	9.3 KB	2016-01-21	added a bunch of template files
 index.html	44 B	5 hours ago	update
 profile.html	3.1 KB	2016-01-21	added a bunch of template files
 readme.md	2.5 KB	a minute ago	readme.md edited online with Bitbucket
 signin.html	3.0 KB	2016-01-21	added a bunch of template files
 students.html	2.9 KB	2016-01-21	added a bunch of template files

Image B2: Git Repository (hosted on BitBucket) showing updated content and progress of commits.

<https://bitbucket.org/geozak/sea-registration>

Author	Commit	Message	Date
geozak	16d8f1b	partial merging of chris's branch	2 days ago
geozak	e9afda3	partial merging of chris's branch	2 days ago
chris-brooks	8c9d869	more view work	4 days ago
chris-brooks	f07bdf	working on account page	4 days ago
chris-brooks	07171ab	updating account index page	4 days ago
chris-brooks	35f939e	change login header	4 days ago
geozak	3ca9678	merge zak into develop to work on merging chris	5 days ago
Christopher Br...	12e3b84	fixed header	5 days ago
geozak	87e74ed	Finally purged all unneeded stuff remaining from the framework	5 days ago
Christopher Br...	6f8c85e	style is working on pages I can view. nav bar is having problems.	5 days ago
Christopher Br...	8e0fe0b	resolve conflicts	5 days ago
geozak	1bbe33c	live server working	5 days ago
geozak	0d19eb1	dynamic live vs development environment	5 days ago
Christopher Br...	c89c5c2	working on views	5 days ago
geozak	53aa08a	accounts controller working	5 days ago
geozak	0539755	made working on live server	5 days ago
geozak	15453f4	add web.config file. require root=public.	5 days ago
Christopher Br...	fe77e0c	working on db auth'd pages	2016-02-25
Christopher Br...	96af98a	Merged zak's changes into branch	2016-02-22
Christopher Br...	5ba7b6d	is db working??	2016-02-22
geozak	d576f4e	accounts view basics done. Related Actions still need making.	2016-02-21
chris-brooks	09a68c6	Work on views and CSS	2016-02-21
geozak	8a230d9	Most of accounts controller done	2016-02-21
chris-brooks	0b829b9	Moving existing views into bootstrap. Including bootstrap CSS, jQuery, and fonts.	2016-02-21
geozak	844fafc	Accounts/index view example done	2016-02-19
geozak	fc6db15	accounts/index mostly done	2016-02-19
geozak	38aefc2	Added restrictAccessTo method in AUTH	2016-02-17

Image B3: Git Repository showing project development progress.

<https://bitbucket.org/geozak/sea-registration>

change destination
master
chris

Create pull request

Diff Commits Merged pull requests

Author	Commit	Message	Date
2 commits behind master. Sync now.			
Christopher Br...	da19203	Feedback done. Header & Footer done. Account, Register, Login in progress	3 days ago
geozak	cfdf0a7	Finally purged all unneeded stuff remaining from the framework	5 days ago
Christopher Br...	4cdf759	fonts css	5 days ago
chris-brooks	8c9d869	more view work	2016-02-27
chris-brooks	f07bdf	working on account page	2016-02-27
chris-brooks	07171ab	updating account index page	2016-02-27
chris-brooks	35f939e	change login header	2016-02-27
Christopher Br...	12e3b84	fixed header	2016-02-26
Christopher Br...	6f8c85e	style is working on pages I can view. nav bar is having problems.	2016-02-26
Christopher Br...	8e0fe0b	resolve conflicts	2016-02-26
Christopher Br...	c89c5c2	working on views	2016-02-26
Christopher Br...	fe77e0c	working on db auth'd pages	2016-02-25
Christopher Br...	96af98a	Merged zak's changes into branch	2016-02-22
Christopher Br...	5ba7b6d	is db working??	2016-02-22
chris-brooks	09a68c6	Work on views and CSS	2016-02-21
chris-brooks	0b829b9	Moving existing views into bootstrap. Including bootstrap CSS, jQuery, and fonts.	2016-02-21
geozak	447a301	fix links in account/index	2016-02-15
geozak	84dccb	Added some controller pages. Started migrating user to account.	2016-02-15
geozak	9cc93b2	remove dashboard	2016-02-14
geozak	ce2c677	Removed note&profile controllers,views,&models. Removed avatars.	2016-02-14