MEAN.machine

Business Career Center Application

Project Report

Names: Benjamin Behrend, Donna Tabique,

Jaime Wallitschek, Kawika Bader, Zhengping Duan

Date: 08/10/2015

Table of Contents

Executive Summary

The following solution offers a responsive web application method of conveying W. P. Carey Career Center information to students. The primary purpose is to overcome the current supply shortage of employed coaches to enrolled and applicable ASU students. The methods used to deploy this software project were Agile and SCRUM management based styles. Weekly sprints were implemented by the team to complete milestones and receive constant feedback throughout development. User testing was implemented to improve user experience and improve knowledge transfer cycle between coaches and students. The current web portal and tools are not able to adequately communicate and empower the aggressive and competitive demand of Business School students. Use of a responsive and application based solution will allow simplification for students and increased productivity for the Career Center. Note that this solution is not fully developed and meets only the agreed upon scope and no current integration with ASU technologies.

Team Member Introductions

Benjamin Behrend acted as the SCRUM's Product Owner. His role included acting as the representative between the Career Center and the development team. Assisting all team mates, he provided support and research assistance weekly. He also setup an Amazon Web Service account to host the team's produced application. Ben brings to the team over fifteen years of experience in IT as an Infrastructure Engineer with specialization in networking and security. He will be graduating in December with dual majors in Computer Information Systems and Business Data Analytics.

The contributing Scrum Master for team was Donna Tabique. As the Scrum Master, she acted as the team leader who prioritized the tasks for weekly sprint commitments. She also constantly acted as the knowledge center for the full application scope. Donna was a contributor who supported the team during configuring. She customized a generator to build an application shell based on the MEAN stack and lead troubleshooting of issues. She also focused on front-end development by creating prototypes and customizing a responsive web application to fit ASU branding guidelines.   This was accomplished using HTML and CSS, and added

functionality using AngularJS. She also created documentation for releases and user testing feedback. Donna is studying Computer Information Systems with a minor in Psychology. She will be studying abroad in the fall semester to obtain a certificate in International Business. In addition, she is currently an IT Security Analyst intern at Avnet and focuses on enterprise audit and compliance.

Jaime Wallitschek is graduating in August of 2015 and double majoring in Computer Information Systems and Accounting. The role Jaime represented was one of the three SCRUM Team Members. Her primary contributions to the product related to database code and creating reference documentation for the client relating to MongoDB. Documentation was also created by Jaime throughout all of the sprint meetings. Zhengping Duan represented the team's second Team Member in SCRUM activities. His focus lied with testing during all sprints executed. However, due to unbalanced workloads during implementation he also assisted with MongoDB, and the responsive web calendar. Zhengping will be graduation in August with a double major in Computer Information Systems and Supply Chain Management.

Kawika Bader supported the scrum team as the third team member, who was able to provide various technical perspectives and ideas. He was able to provide this insight through the use of past professional work experience in the software field, specializing specifically in web technologies. He enabled team members through peer to peer programming, assisted troubleshooting, and assisted in research to equip the team. Kawika Bader joins the team with over 5 years of technical support and software development experience, as well as a passion for innovation and current web technologies. He currently works as a Software Developer, working directly with web applications and modern web languages and software practices. He will be completing his degree in Computer Information Systems with the successful completion of this course, during the summer session of the 2015 school year.


Client Introduction

This product is to be delivered to Arizona State University, W. P. Carey's Business Career Center. The two primary contacts for this client were Sarah Hill and Michelle Stelter. Providing support for approximately 1,100 students the career center attempts to improve student experiences and success while enrolled at ASU. This is completed by providing eight career

coaches that specialize in assisting groups of students by major. There are currently twenty-seven major specific programs and nine possible certifications available to W. P. Carey student population. Student access to career coaches also depends on the enrolled major and appointment availabilities.

Problem

The current demand for career coaching and education is not currently managed by the amount of staff on hand. The ratio of coaches to students creates an un-manageable challenge for the current ASU staff. There is specific problems with arranging meetings with students and maintaining contact the large student population effectively. Primarily, the existing web site lacks features and functionality to allow students to monitor degree progress. The use of technology solutions is not being fully utilized by the W. P. Carey's Career Center. There current student portal does not follow ASU branding policies and was critically confusing to navigate by students.

Current W. P. Carey Student Portal:
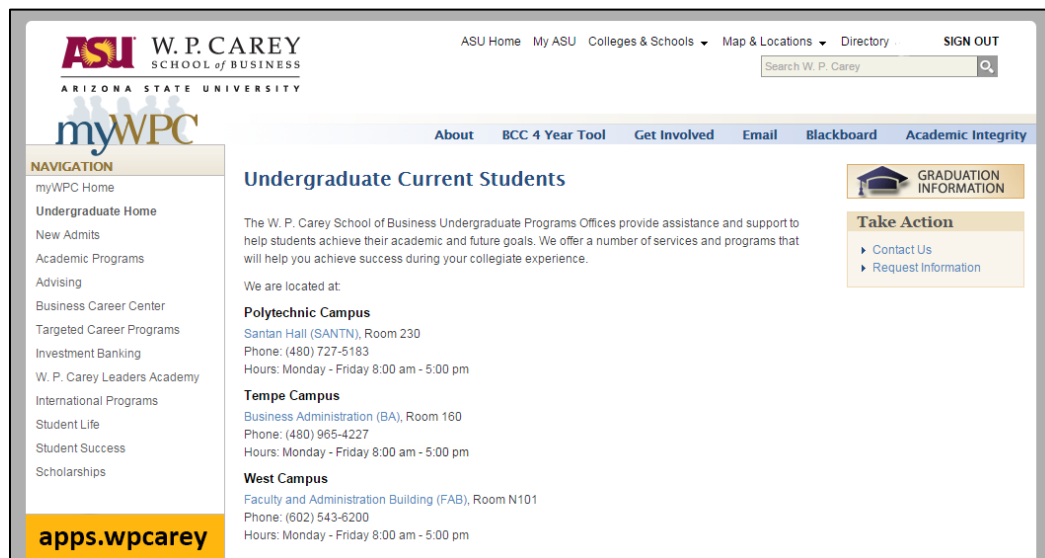
Proposed Solution

To allow for more real time interactions, the Career Center would like a user friendly web application targeting the maximum amount of students possible. Specifically the solution targets incoming freshmen to the W. P. Carey school but also includes already enrolled students of the school. The initial product will be a responsive web application that contains features and

functionality outlined below.   The solution will include client documentation and include a public code repository of the created application. The MEAN technology stack will be used to implement the solution in a six week timeline.


Functionality

Actual

Using a Amazon Web Service the team has delivered an application that is working on a public address. The coding languages used to create the final product include something, something, and something. More details here on actual product. Include any missed items not incorporated in final version of product. Include screenshots of complete application home page.

Planned

Using the MEAN stack, the team decided to plan on implementing a web responsive application. By implementing this instead of a native application the team was able to do something. The client provided a primary list of requirements that included page and user trackable data. There were number of primary pages initially requested to be included in the application. These included "Home", "Grad Club", "Calendar", "More", and "Tools". Additionally it was provided that the information desired from the application was primarily user demographic and tracked usage along with page activity.

Paper Prototypes:
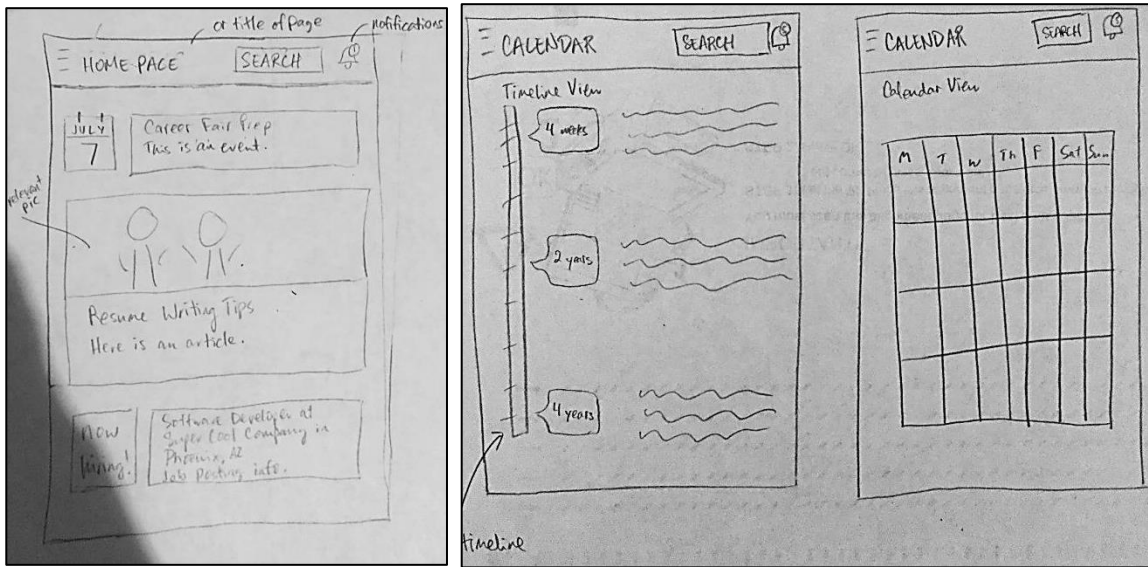
Digital Prototypes:
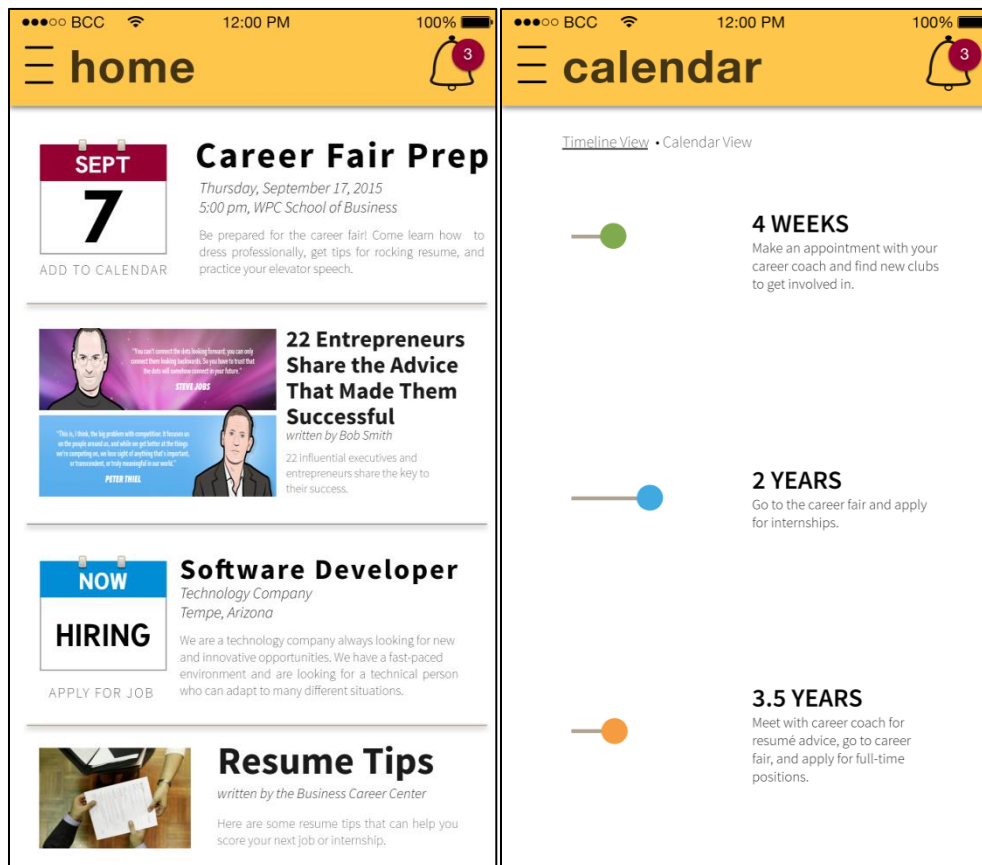
Technology Stack Information

Stack Overview

The technology stack chosen for the application is abbreviated to "MEAN". Each letter in this abbreviation stands for a component of our technology stack. "M" correlates to MongoDB which is a public collection based database structure. This was chosen to allow for a "SQL-less" project to be created and focus on a document-oriented database focus that is cross-platform capable.

The second technology component is the "E" or ExpressJs. The ExpressJS is minimalistic and un-opinionated to allow easy coordination with our "N" component of the "MEAN" stack. The "N" or NodeJS technology allows for a fast building scalable network application. It is event driven IO model meant to be lightweight while still providing data-intensive applications that run in real-time. The final component of our technology stack is "A" for AngularJS which allows for an extended HTML library to be used in the application

MEAN Acronym Visual:

MongoDB database	ExpressJs work	AngularJs work	NodeJs work

General Code Information

Information needed to understand code here…

Testing and Evaluation Activities and Findings

Target Audience

Testing was directed at current ASU students ranging from Freshman to Senior specifically enrolled in the W. P. Carey School of Business. Current students were evaluated and an emphasis was placed on finding Freshman students to interview. The value hypothesis (value provided by product if used by customers) is blah. Then the growth hypothesis (how new customers discover product) is blah.

Performed Methods

The evaluation of the application product was performed at the three week and five week marks. User testing was performed using a test script references to desired outputs. Room was provided for the students to provide commentary and "screen shot" results as possible.

Results and Findings

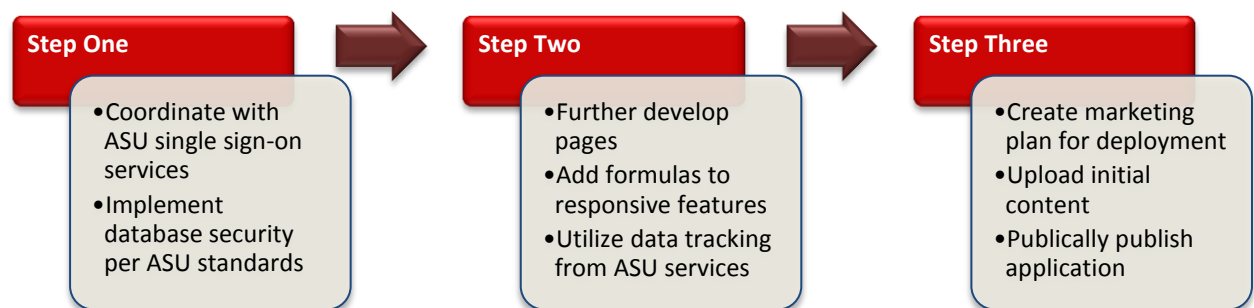How results were gathered

Wave one results

Wave two results

Next Steps

Non-Technical Overview

The work to be completed shall be overviewed here in a general (non-technical) summary. The agreed upon solution prepared by the MEAN.machine team is not complete and will require further development steps. The product is not ready for actual student usage and has been summarized to need three general types of work remaining. Step one will include work to synchronize the application with ASU technology services. The second step will require completing customization of the application and improvement of the user interfacing. Then, step three will focus on rolling the application out to students and adding of content. Please see the below section for details on the requirements of future prior to the application usage.

Visual Summary:

Step One
- Coordinate with ASU single sign-on services
- Implement database security per ASU standards

Step Two
- Further develop pages
- Add formulas to responsive features
- Utilize data tracking from ASU services

Step Three
- Create marketing plan for deployment
- Upload initial content
- Publically publish application

Technical Overview

Step One

One of the first steps to completing the application should be coordination of the code with ASU single sign-on services. This will ensure that only actual ASU students will be able to register for the application. This should be accomplished by communicating with the W. P. Carey Technology Services (https://wpcarey.asu.edu/technology-services). Their current portal allows for help requests to be submitted online under the "Service Request Forms" header of the navigation pane. While in contact with the Technology Services team, security aspects should start to be developed. ASU standards for student data storage and administrative access were not included in the project's initial scope due to time constraints. This means that the current structure while attempting to use best practices is still not compliant will all ASU standards. There is a current online portal (https://getprotected.asu.edu/content/securityreview) for requesting a formal Security Review from ASU. Questions and requests may also be sent to the address, "security.review@asu.edu".

Step Two

The physical code for the application can be found all on a public repository that is hosted by the public site, GitHub. Before the application can provide a responsive interface as desired, pages will require further development. Specifically the team imagines writing formulas or algorithms to allow the content inside the application to update by student logged-in. This would require adding an algorithm to the Timeline page so that the display is generated by the user's input graduation date.

Also, currently the data for the application is being stored in a MongoDB database. This is not synced with existing ASU storage methods. Similar to the mediation in step one, the development team will need to coordinate with ASU Technology Services to utilize the trackable data options. This portion of work could be done in conjuncture with step one but is less crucial than ASU sign-on updating. If sprint timelines allow for more work to be taken on, this could be performed earlier but is expected to take time due to ASU politics.

Step Three

The work summarized in step three relates to the actual roll out of the application to users for registration and initial usage. A possible tool recommended for this is the Xamarin platform

(http://xamarin.com/). This tool allows for applications to be built, tested, and monitored all across platforms that include Target iOS, Android, Windows, and MAC. The Xamarin application tool allows for a development team to implement the solution using native user interface controls unlike the current MEAN technology stack.

Future Enhancements

1) Implement a badge system
   a) Allow users to track progress during each academic year with points that could relate to physical prizes. This would provide a form of "gamification" to the application and encourage repetitive check-ins from users. Prizes could include "swag" from the Career Center and also encourage students to go on campus and visit in person with Career Center staff at W. P. Carey.
2) Social sharing with Grad Club page
   a) Specifically requested by the client, a Grad Club page should allow for social sharing between W. P. Carey students. This page could include stories, articles, art, or other possibly user submitted ideas. Code has been commented into the current application structure to allow easy integration for this page feature.
3) Push notifications (mobile only)
   a)
4) Upgrade Calendar page for Mobile options
   a) Create an option to push application data onto personal phone device. This would allow students to synch their phone calendar with the application calendar possibly. Or possibly send calendar notifications to a phone or personal email other than "asu.edu".
5) Allow tracking of appointments
   a) Provide a notification inside application if user has any scheduled appointments with the W. P. Carey Career Center. This could either be done through synchronization with the current Advising SOS application or other methods. Could also coordinate with the third option, "Push notifications", to allow students to receive text message reminders one day before, two hours before, and at appointment start time.

Appendix

Change Log

Version 0.1

Focus began by clarifying future sprints and user stories for the ensuing six week timeline. This required evaluating a solution for the Career Center's problem presentation on July 2nd, 2015. Ben, the Product Owner, created an initial readme.md file on the team's public code repository using GitHub. He also created a highleveltasks.md file that overviewed the general work needed to create the desired solution. Ben also further communicated with the Career Center to create a productbacklog.md file that listed complete user stories onto GitHub. His conversations with the client also resulted in a futurefeatures.md file that includes possible functions to consider with the solution hand off.

Also, the team coordinated to evaluate all possible technological and structural information needed. Conversations were had on technology stack preferences and how to utilize specific skills of each team member. Multiple documents were created to allow the team to narrow the to-be application's scope. First, Donna created paper and digital prototypes for the proposed user interface mirroring the requested template of "BabyCenter". Also, the prototype for the database was created by Jaime with basic understanding of the necessary fields and collections to be created. No functioning code was created but instead Mongo CRUD functionality was researched.

For testing, Zhengping studied all of the basic MEAN stack functions for the future user stories. He also went through GitHub tutorials throughout the week. Then, Kawika helped establish the GitHub and individual development environments. He also provided research advice to teammates based on experience. Kawika helped coordinate all the meetings for the week and assisted in leading discussions. Finally, the team did not publish a release on GitHub by the deadline due to error.

Version 0.2

Research and training by team members was still being conducted into the second week of development. Ben, as the Product Owner, studied all of the MEAN stack elements to assist the team in coordinating the four technologies. To focus the team, Ben prioritized team deliverables for the week's sprint. This was done by continuing communication with the Career Center of the team's weekly deliverables. Donna also prioritized the backlog tasks while helping teammates troubleshoot. During the sprint, Donna configured a Yeoman generator to create the front and back end functionality for the application files. Kawika had initially discovered the tool and assisted in the implementation with Donna to build a basic structure. Zhengping focused on setting up a local environment using Yeoman. At the same time, Zhengping was also troubleshooting the local environment with a Windows platform.

Donna also customized the application to fit ASU branding guidelines. Kawika then implemented a home page based off of the previously made prototypes. Jaime studied MongoDB along with Nodejs and Expressjs code for implementing the solution's database. Two specific collection types were created to store user data and page related content. Jaime also wrote support documentation relating for the future hand-off of the solution. This sprint Donna published the GitHub Version 0.2 release along with the documentation created to date.

Version 0.3

Version 0.03 was the third week long sprint completed by the team. As a reoccurring theme, Ben lead the team's prioritization of deliverables for the sprint week with Donna prioritizing backlog tasks. Also, communication was constant between Ben and the Career Center representatives. To host the application, Ben also started building the Amazon Web Services (AWS) environment.   Kawika assisted in troubleshooting the AWS and ad-hoc GitHub issues during the week.

For the front end, Donna created a basic timeline for the Four Year Plan page using blank coding. She then also developed extra sign up fields for the login page's functionality. For the Grad Club page, Kawika began the needed base code. Along with this page, he also created all needed page frameworks for other group members to add code. Throughout the week routing developed into a confusion area for the team and required Kawika's assistance to resolve issues.

For the database, the basic structure was exported into CSV (Comma Separated Values) files from Jaime's local machine. Zhengping and Jaime worked to integrate the Mongo code,

comprising of the database, with the application. Along with the two exported collections, the MongoDB reference document was compiled for the future hand-off. Working with user interfacing as well, Zhengping also designed the responsive details of the calendar page design using JavaScript, HTML, and jQuery.

Other documentation created during this sprint includes a user testing script for the current working progress of the application. Zhengping helped perform testing for all of the user login functions and experience. Donna also conducted user testing and compiled user feedback into documentation. During the week, she created a basic shell for the team's upcoming report. Then finally, Ben performed quality assurance by monitoring the software engineering on a MAC and Linux. Donna again published the GitHub version release for the team.

Version 0.4

Continuing into week four Ben prioritized the team's sprint tasks and was communicating deliverables to the Career Center. The AWS environment was completed by Ben by resolving blank issues that had been outstanding. Documentation was recorded throughout the updating of the environment to include at the hand-off. Jaime created a basic future implementation plan as a section in the final report. She also compiled a content and link reference for teammates to easily locate W. P. Carey information that currently isn't easily locatable. Kawika assisted the team by further defining the scope with removal of low priority features from the back log. This allowed the team more time with understanding jQuery routing.

Donna completed a large portion of code and assisted Kawika in troubleshooting the UI/Angular routing for adding additional files. The design and functionality implementation for the Tools and More pages was also complete by Kawika. Meanwhile, Donna configured CSS and JS files with integration of the customized Timeline page features. A basic sidebar was also created based off of week three user testing feedback. Another response was that she also developed even more sign up fields for the Login page interfacing. Zhengping continued testing routing for the application's JQuery code. He improved the format to the Calendar page to increase user experience from feedback based as well. Donna again published the fourth version release on the team's public GitHub repo.

Version 1.0

      For the final sprint, Ben prioritized all team deliverables and communicated with the client. As a final clean up, Donna polished front end interactions along with testing and fixing the sidebar. Troubleshooting jQuery was related to routing by Donna. During coding Donna completed comments for her code and completed additional support documentation. Zhengping also completed his portion of support documentation. This was while Ben was completing the environment set up documentation. The AWS environment was completed with all code from the previous week's deliverable presentation. Then, all members of the team contributed to the project report during the week. A presentation was also developed showcasing the solution on August 10th, 2015.

Resource Links

GitHub

    1. https://github.com/asu-cis440-summer/career-center-MEAN.machine

Amazon Web Services (AWS)

    1. USE MOST UP-TO-DATE LINK

MongoDB

    1. https://www.mongodb.org/

    2. http://mongoosejs.com/

    3. https://github.com/mongodb/mongo

Expressjs

    1. http://expressjs.com/

    2. https://github.com/strongloop/express

Angularjs

    1. https://angularjs.org/

    2. https://github.com/angular/angular.js

Nodejs

    1. https://nodejs.org

    2. https://github.com/joyent/node

    3. https://github.com/mongodb/node-mongodb-native