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. The team implemented weekly sprints were implemented to complete milestones and receive constant feedback throughout development. User testing was conducted 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 provides a fully functioning application. However, this only includes a previously agreed-upon scope, select features, and no current integration with ASU technologies.

Team Member Introductions

Benjamin Behrend acted as the Product Owner. He was the primary point of contact between the Career Center and the development team. Additionally, he was responsible for prioritizing the user from the Career Center . Assisting all teammates, he provided support and research assistance weekly. He also set up an Amazon Web Service account and produced an application instance 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 the 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 application configuration. She customized a generator to build an application shell based on the MEAN stack and assisted with 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 were related to database code and creating reference documentation for the client relating to MongoDB. Jaime also created documentation throughout all of the sprint meetings. Zhengping Duan represented the team's second Team Member in SCRUM activities. He focused on testing during all sprints executed. However, due to unbalanced workloads during implementation he also assisted with MongoDB, home page formatting, and the responsive web calendar. Zhengping will be graduating 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 and provided various technical perspectives and ideas. He provided insight through the use of current and 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 five 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, 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's, 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 currently unmanageable by the amount of staff on hand. The ratio of coaches to students creates an unfeasible challenge for the current Career Center staff. There are specific problems in arranging meetings with students and effectively maintaining contact with the large student population effectively. The existing website also 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. Their current student portal does not follow ASU branding policies and was confusing to navigate.

Current W. P. Carey Student Portal:
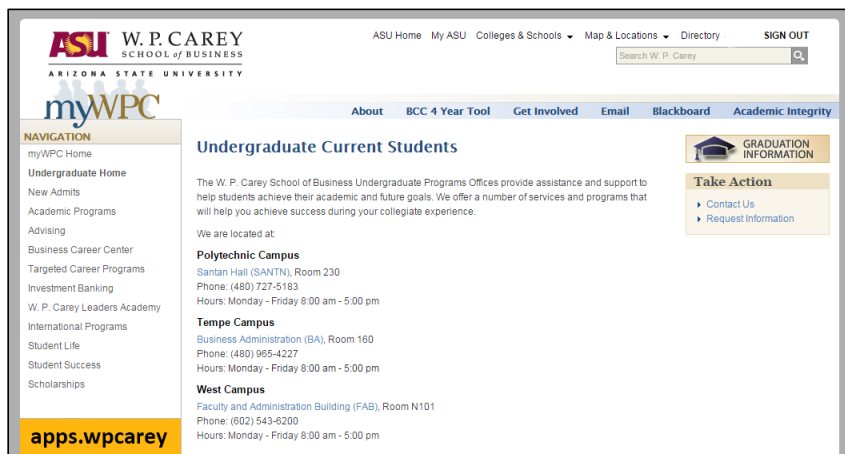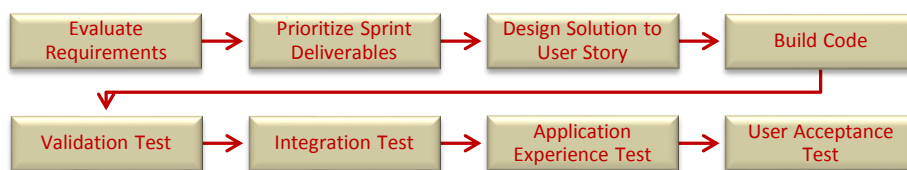

Proposed Solution

To allow for more real time interactions, the Career Center would like a user-friendly web application targeting the highest potential number of W. P. Carey students. Specifically, the solution targets incoming freshmen to the W. P. Carey School but includes already enrolled students as well. The team proposed a responsive web application that contains features and

functionality as outlined below.   The solution will include client documentation and include a public repository of the created application. The recommendation from the team was to use MEAN (Mongo, Express, Angular, Node) as the technology stack to implement the solution in a six-week timeline. It was a team decision that the scope would be limited to plausible tasks in the tight schedule. Future work recommendations will be included in the solution for how to further develop the application.

Weekly Iteration Plan:

Evaluate Requirements → Prioritize Sprint Deliverables → Design Solution to User Story → Build Code

Validation Test → Integration Test → Application Experience Test → User Acceptance Test

Functionality

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 expected to improve the W. P. Carey Career Center's information accessibility on all types of devices and browsers. The client provided a primary list of requirements for the solution that included demand for page and user trackable data. There were number of primary pages initially requested to be included in the application. This included "Home", "Grad Club", "Calendar", "More", and "Tools". Additionally it was provided that the design should allow for students to interact with their school schedule interactively. This could be done with either a check-list or responsive timeline of a four year plan.

Tools, other than the MEAN stack, expected to be used were bootstrap, something and something. The type of information to be included in Tools page would be student's "first internships", "first jobs", "career fair experiences", and etc. The More page was intended to provide an external link to other existing W. P. Carey tools and applications.

Paper Prototypes:

Comment [U1]: Kawika, Donna or Zhengping, please provide more material here
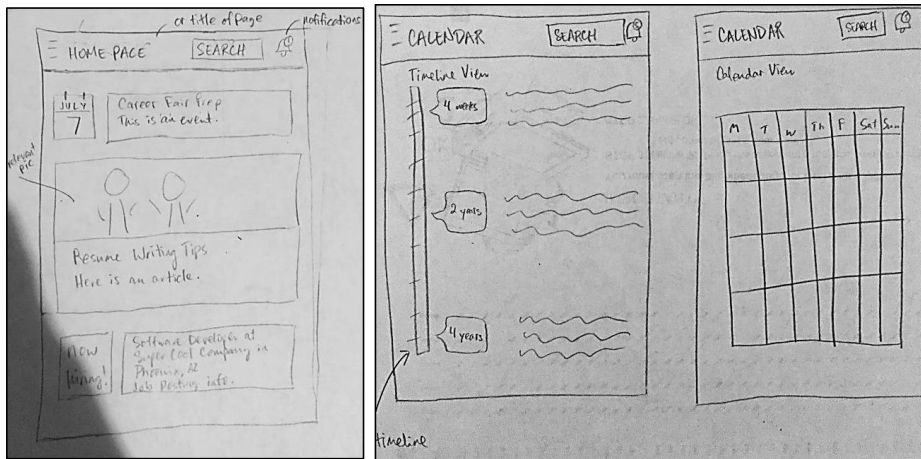
Digital Prototypes:
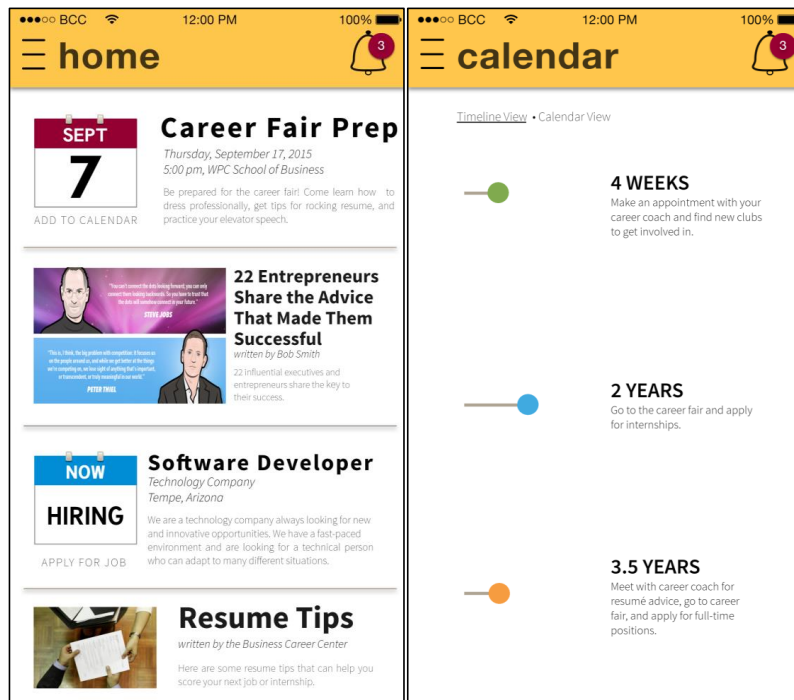
<u>Actual</u>

By hosting Amazon Web Services, the team has delivered an application that is available from the internet, anywhere. The programming languages used to create the final product include HTML, CSS, JavaScript, MongoDB, ExpressJs, AngularJs, and Node.Js. The initial load screen of the application is the Home page that acts like a newsfeed for posted events and articles. Additionally, the team implemented:

- Sign Up page to register for the site
- Sign In page that allows the user to edit their profile
- Calendar page
- Timeline page for the Four Year Plan
- Tools page that includes content

More page that includes external links to other pertinent information for the Career Center

==More details here on actual product. Include any missed items not incorporated in final version of product.==

Computer Home Page View:

==Need screenshot(1 or 2)==

Mobile Home Page View:

==Need screenshot(1 or 2)==

Technology Stack Information

<u>Stack Overview</u>

The technology stack chosen for the application is abbreviated to "MEAN." Each letter in this abbreviation stands for a component of the technology stack. "M" correlates to MongoDB, which is a public collection-based database structure. This was chosen to allow for the solution to be created without SQL and instead focus on a document-oriented database. Another benefit of Mongo is that it is cross-platform capable technology.

The second stack component is the "E" or ExpressJs framework. The ExpressJS is minimalistic and un-opinionated to allow creation of a robust API. There is also easy coordination with our "N" component of MEAN. The "N" or Node.Js technology allows for a

fast building scalable network application in development. It is event driven IO model meant to be lightweight while still providing data-intensive applications. This is also a real-time running platform. The final component of our technology stack is "A" for AngularJS. This full feature single-page-application framework allows for an extended HTML library to be utilized. It was built as a top layer to jQuery that added MVC concepts. This stack element was actually the driving force behind the team's selection of MEAN for solution implementation.

MEAN Acronym Visual:

MongoDB	ExpressJs	AngularJs	Node.Js
database	framework	HTML extension	network

General Code Information

Information needed to understand code here…

Testing and Evaluation Activities and Findings

Target Audience

The target audience are students that are currently enrolled in the W.P. Carey School of business at ASU. More specifically, students that want to track their student progress throughout their academic career with hopes to obtain a successful career upon graduation. Testing was directed at current ASU students ranging from Freshman to Senior and those that were currently enrolled in the W. P. Carey School of Business. An emphasis was placed on finding Freshman students to interview.

Performed Methods

The evaluation of the application product was performed at the three week milestone of the schedule. User testing was performed using a test script that provided users with references to desired outputs. Room was provided for the students to provide commentary on a specific page, general layout, and other opinions.   Each instruction provided a "desired output" to allow each user to base their experience subjectively from the pictured example. A team member was present for questions during the actual implementation of testing with students. In some cases,

the tester had to use a team member's laptop because of server issues online using Amazon Web Services. The published test script can be found at the following link.

Results and Findings

General comments resulted in positive feedback that users would use the application again. Specifically, some excerpts from the testing feedback include "I would use it", "Yes, I would use a web app over a mobile app", and "I'm not sure if would use it on my phone as much because I do not want to use mobile applications". There were other suggestions that students provided without prompting. Initially, a tester suggested that the navigation bar additionally have subcategories for directing searches. Also, the majority of testers expressed a desire for prepopulated fields in the user Login page. More specific results of the testing can be seen on the GitHub "Documents" repository.

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 focused on creating a product that can easily be integrated into ASU website as well as an application container for mobile device compatibility. However, it 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 integrate the application with ASU technology services. The second step will require completing the customization of the application and improvement of the user interface. Step Three will focus on including additional content and releasing the application out to students. Please see the below section for details on the requirements of future prior to the application usage.

Visual Summary of Next Steps:

Step One		Step Two		Step Three
•Coordinate with ASU single sign-on services •Implement database security per ASU standards	→	•Further develop pages •Add formulas to responsive features •Utilize data tracking from ASU services	→	•Create marketing plan for deployment •Upload initial content •Publically publish application
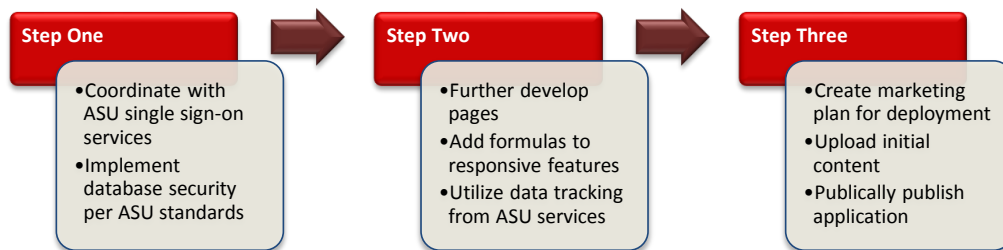
Technical Overview

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 considered and implemented in the solution. 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 with 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 on a public repository that is hosted by the public site, GitHub @ https://github.com/asu-cis440-summer/career-center-MEAN.machine. 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 students that log in. This would require adding an algorithm to the Timeline page so that the display is generated by the user's input graduation date.

The data for the application is currently being stored in a MongoDB database. This is not in sync 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 conjunction with step one but is less crucial than ASU sign-on updating. If sprint timelines allow for more work commitments, this could be performed earlier but is expected to take time due to ASU policies.

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) Allows students to get notifications on their mobile device for events they add to their calendar . The notifications would have customizable settings for increasing efficiency of updates viewed.

11

4) Upgrade Calendar page for Mobile options
    a) Create an option to push application data onto personal phone device. This would allow students to sync their phone calendar with the application calendar or 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.
6) Search feature
    a) Create search bar and algorithm to search through site content for faster access. Allowing information to be searched for by tag works and phrases could also be investigated.

Appendix

Change Log

Version 0.1

Focus began by clarifying future sprints and user stories for the ensuing four-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.

Product Back Log Visual:
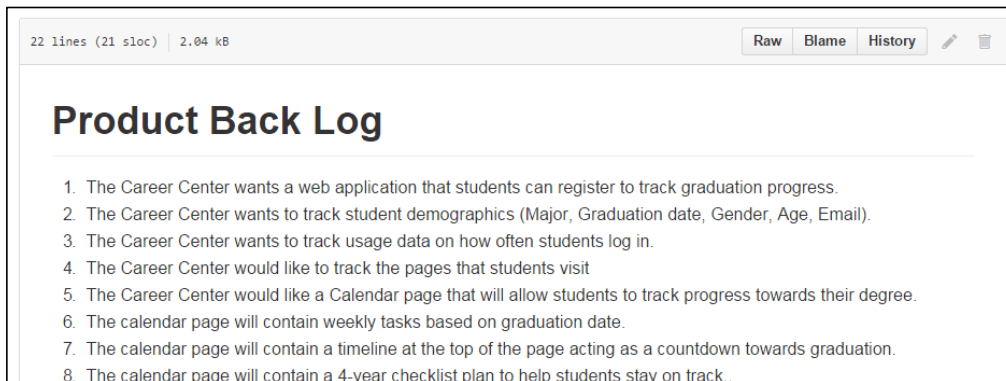
Also, the team coordinated to evaluate all possible technological and structural information needed. Conversations were had on technology stack preferences and how to utilize specific skills of each team member. Multiple documents were created to allow the team to narrow the to-be application's scope. First, Donna created paper and digital prototypes (examples under "Functionality – Planned") to show the user interface of the site. Then, Jaime created the prototype for the database 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. In this sprint, Donna published the GitHub Version 0.2 release along with the documentation created to date.

Release Two Documentation Visual:

1. Yeoman
1.1 Yeoman Installation

1. Before installing Yeoman, download Nodejs and Mongodb onto your system. For Mongodb instructions, see Section 2 for Mongodb and Section 3 for Nodejs.
2. Put the Mongodb "/data/db" directory into your application's directory.
3. Open Terminal or Command Line.
4. Change the current directory to your application's directory by typing

   ```
   cd appDirectoryName
   ```

   appDirectoryName is the name of your application directory folder.

5. To install Yeoman on your system, type:

   ```
   npm install -g yo
   ```

6. Install the MEANJs generator by typing:

   ```
   npm install -g generator-meanjs
   ```

7. Type: yo

8. Yeoman will then prompt the user to answer the following questions:

   ```
   ? 'Allo <user>! What would you like to do? Meanjs
   ```

Version 0.3

Version 0.03 was the third week-long sprint completed by the team. As a reoccurring theme, Ben led 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.

14

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.
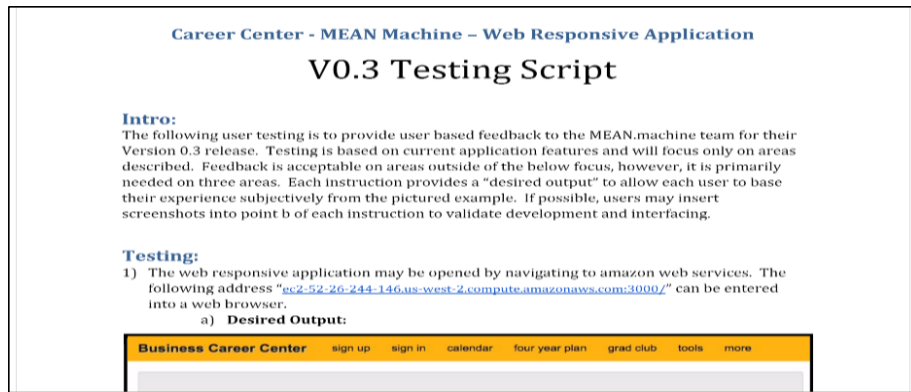
For the database, the basic structure was exported into CSV (Comma Separated Values) files from Jaime's local machine. Zhengping and Jaime worked to integrate the Mongo code, comprising of the database, with the application. Along with the two exported collections, the MongoDB reference document was 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.

Collection One CSV:

First Name	Last Name	Username	Password	Email	Major	Grad Date	Gender	Age
Jaime	Wallitschek	jwallits	jwallits	jwallits@asu.edu	CIS	08002015	Female	21
Donna	Tabique	donnatabique	donnatabique	dtabique@asu.edu	CIS	00000000	Female	00
Benjamin	Behrend	bbehr	bbehr	bbehrend@asu.edu	CIS	00000000	Male	00
Kawika	Bader	kawikabader	kawikabader	emil.bader@asu.edu	CIS	00000000	Male	00
Zhengping	Duan	zduan2	zduan2	zhengping.duan@asu.edu	CIS	00000000	Male	00

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.

User Testing Script Visual:

Career Center - MEAN Machine – Web Responsive Application

V0.3 Testing Script

Intro:
The following user testing is to provide user based feedback to the MEAN.machine team for their Version 0.3 release. Testing is based on current application features and will focus only on areas described. Feedback is acceptable on areas outside of the below focus, however, it is primarily needed on three areas. Each instruction provides a "desired output" to allow each user to base their experience subjectively from the pictured example. If possible, users may insert screenshots into point b of each instruction to validate development and interfacing.

Testing:
1) The web responsive application may be opened by navigating to amazon web services. The following address "ec2-52-26-244-146.us-west-2.compute.amazonaws.com:3000/" can be entered into a web browser.
    a) **Desired Output:**

Business Career Center	sign up	sign in	calendar	four year plan	grad club	tools	more

Version 0.4

Continuing into week four, Ben prioritized the team's sprint tasks and was communicating deliverables to the Career Center. He also completed the AWS by resolving deployment and installation 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 navigation bar. Troubleshooting jQuery was also completed by Donna related to routing problems. During the week's coding, Donna completed comments for her code and completed additional support documentation. Zhengping also completed his portion of support documentation for handing off the code. This was while Ben was completing the application's environment set up documentation. The AWS environment was completed with all code from the previous week's deliverable presentation.

Kawika finalized code enhancements and polished features for the front-end interfacing. He also provided technical based business value to the solution's presentation plans. Wrapping up, all members of the team contributed to the project report's release updates and technical selling points. Jaime worked to coordinate all team member's updates into a chronological and representative order for the past six week implementation schedule. A presentation was also developed showcasing the solution due August 10, 2015. The final release Version 1.0 was published on August 9th, prior to the showcase presentation.

Resource Links

GitHub Repository (Repo)

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

Amazon Web Services (AWS)

- http://ec2-52-11-5-180.us-west-2.compute.amazonaws.com:3000/

MongoDB

    https://www.mongodb.org/

- http://mongoosejs.com/
- https://github.com/mongodb/mongo/

ExpressJs

- http://expressjs.com/
- https://github.com/strongloop/express/

AngularJs

    https://angularjs.org/
    https://github.com/angular/angular.js

Node.Js

- https://nodejs.org

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

- End of Report -