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

FORMAT: What does the product offer. Main purpose of product. Methods used to analyze. Procedures applied during analysis. Findings after applying analysis. Calculate results here for review. Conclusion on profit of product. Include information on how product addressed public. Finally include any limitations regarding the report (example: lack of resources)

Team Member Introductions

The MEAN.machine team used an Agile approach to software development and consisted of five active members enrolled in CIS 440 Summer Session B. Benjamin Behrend acted as the SCRUM's Product Owner. His role included acting as the representative between the Career Center and the development team MEAN.machine. Assisting all team members Ben provided support to everyone with research assistance for the weekly sprints. He also participated in the setup of an Amazon Web Service account to host the team's produced application. Ben brings to the team over fifteen years of experience in IT. Ben is an Infrastructure Engineer, where he specializes in Networking and Security. He is a senior at the W.P. Carey School of Business and will be graduating in December with dual majors in Computer Information Systems and Business Data Analytics.

The SCRUM Master was Donna Tabique, a Computer Information Systems (CIS) bachelor of science student at ASU. She acted as the team leader and knowledge center for full application scope. Along with coordinating team efforts, she also participated in the code development. Her work especially utilized the tool Bootstrap and others.

Jaime Wallitschek is a senior double majoring in Computer Information Systems (CIS) 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 to minimize backtracking in later sprints requiring document deliverables.

Kawika Bader was a Team Member.

Zhengping Duan represented the team's third 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, pages, and calendar page code content.

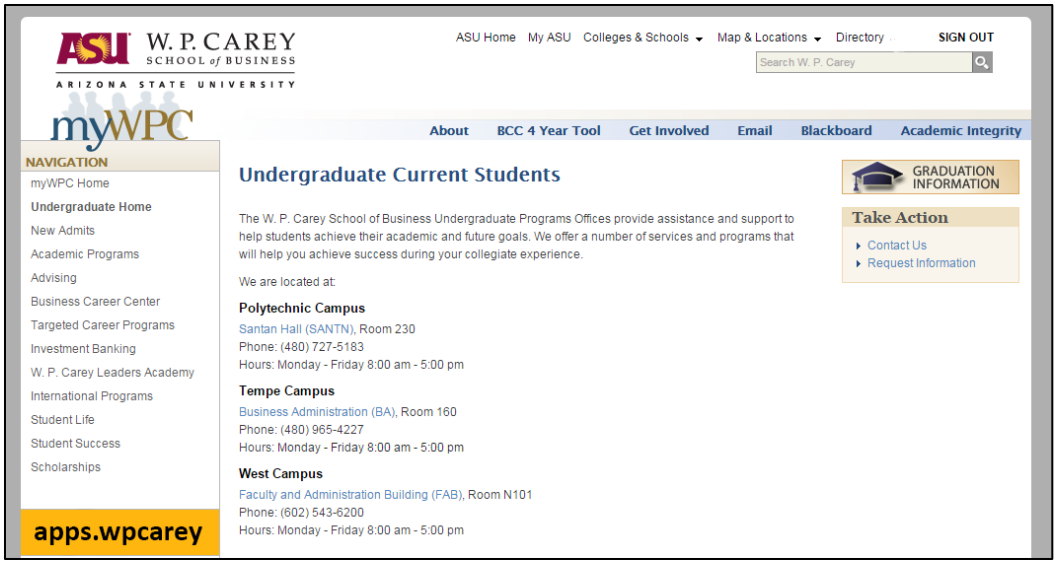
Client Introduction

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 number of major programs and number of possible certifications available to W. P. Carey students.

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. 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

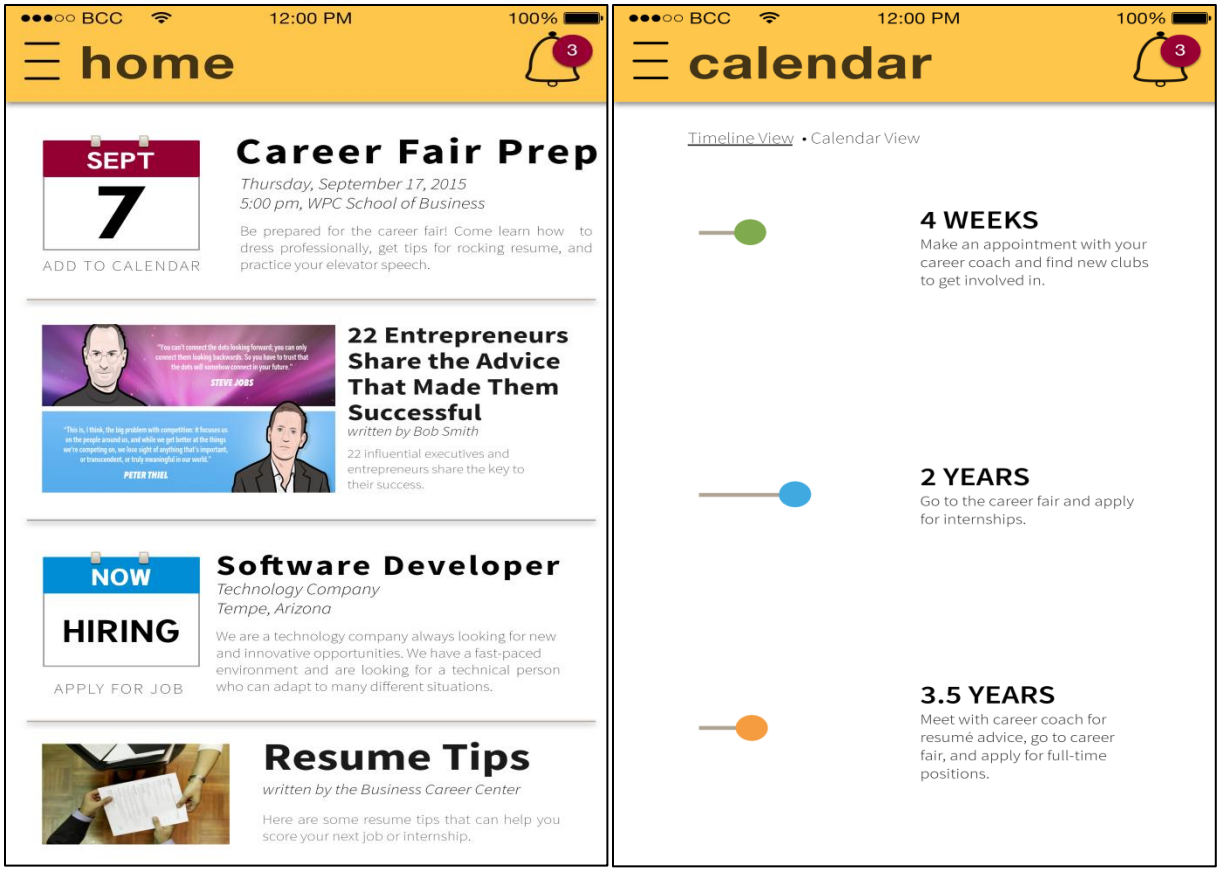
Using a Amazon Web Service the team has delivered an application that is working on a public address. The coding languages used to create the final product include something, something, and something. More details here on actual product. Include any missed items not incorporated in final version of product.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

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.

Digital Prototypes:



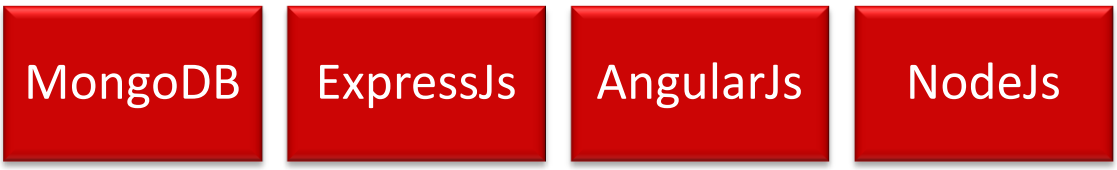
Technology Stack Information

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:



General Code Information

???

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.

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

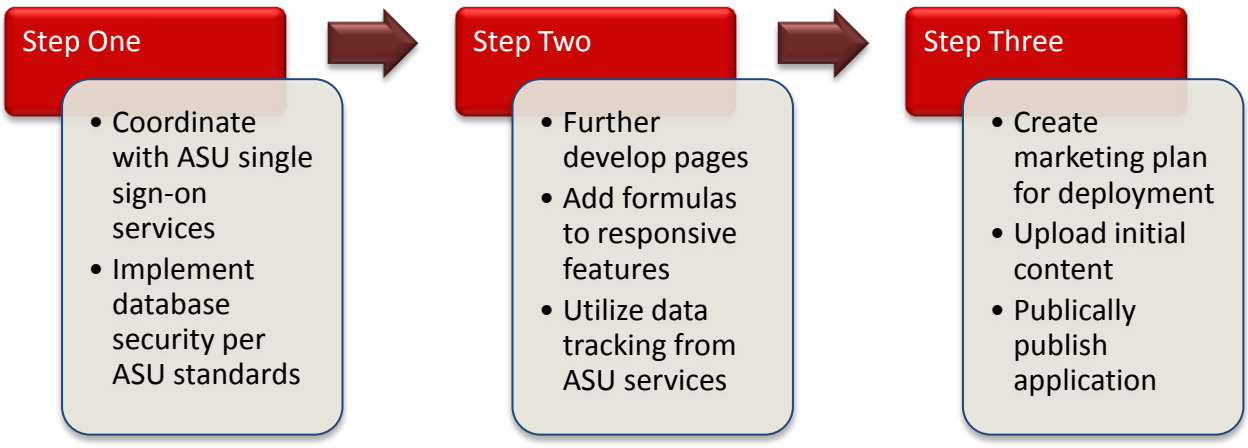
Wave two results

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:



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

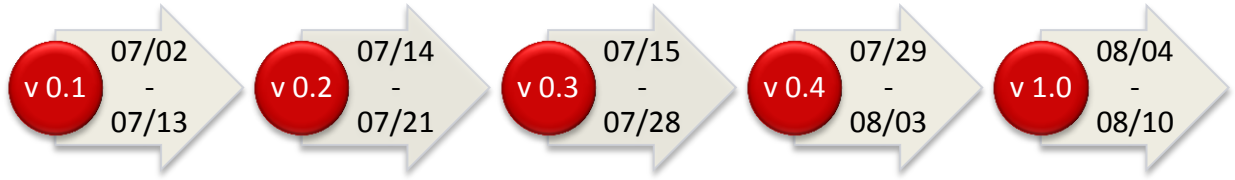
(http://xamarin.com/). This tool allows for applications to be built, tested, and monitored all across platforms that include Target iOS, Android, Windows, and MAC. The Xamarin application tool allows for a development team to implement the solution using native user interface controls unlike the current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

The first week consisted of team sprint planning for the first release of version 0.1. The focus began by attempting to 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 for the 5 remaining weeks. Conversations were had on technology stack preferences and how to utilize specific skills of each team member. A decision was made that the technology stack would be broken up by team members for research. Multiple documents were created to allow the team to narrow the to-be application's scope. Paper and digital prototypes were created for the proposed user interface that mirrored the requested template of "BabyCenter" provided by the Career Center representatives. Also, the prototype for the database was created with basic understanding of the necessary fields and collections to be created. No functioning code was created for the database with MongoDB but instead understanding of MongoDB CRUD (Create R U Delete) functionality. Kawika also brought in a tool called Yeomen to create a basic file structure for the MEAN technology stack.

Version 0.2

Research and training by team members was still being conducted into the second week of development. Ben studied all of the MEAN stack elements to assist the team in coordinating the four technologies. Zhengping focused on testing tools and MongoDB. Jaime also studied MongoDB along with Nodejs and Expressjs.

To focus the team, Ben prioritized team deliverables for the week's sprint. This was done by continuing communication with the Career Center of weekly deliverables. This sprint created the initial database collections evaluated in previous sprint. Two specific collection types were evaluated to store user data and page related content. During the creation process, basic steps / instructions were being written as a portion of documentation needed for a later sprint. This included information on basic MongoDB installation and how to create elements of the collection based database.

Also, a visual prototype of the database was created using Visio.

Version 0.3

Version 0.03 was the third sprint completed by the team. As a reoccurring theme, Ben lead the team's prioritization of deliverables for the sprint week. Like the other sprints, communication was constant between Ben and the Career Center representatives. To host the application, Ben started building the Amazon Web Services (AWS) environment.

Relating to the database, this sprint accomplished exporting the basic structure into CSV (Comma Separated Values) files. Along with the two exported collections, the MongoDB reference document was completed. It now also included information on how to import mongo value files.

Other documentation created during this sprint includes a user testing script for the in progress application.

Version 0.4

text

Version 1.0

text

Resource Links

GitHub

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

Amazon Web Services (AWS)

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

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

Documentation

Windows Installation

MAC Installation