

# Employees Becoming Neighbors

by

James Allen, Micah Johnson, and Zach Goodwin

Submitted to  
the Faculty of the School of Information Technology  
in Partial Fulfillment of the Requirements for  
the Degree of Bachelor of Science  
in Information Technology

© Copyright 2019 James Allen, Micah Johnson, Zach Goodwin

The authors grant to the School of Information Technology permission  
to reproduce and distribute copies of this document in whole or in part.

  
\_\_\_\_\_  
James Allen  
Date 4-15-19

  
\_\_\_\_\_  
Micah Johnson  
Date 4-15-19

  
\_\_\_\_\_  
Zach Goodwin  
Date 4-15-19

  
\_\_\_\_\_  
Abdou Fall, Faculty Advisor  
Date 04-15-19

University of Cincinnati  
College of  
Education, Criminal Justice, and Human Services

April 2019



Prepared by  
**James Allen, Micah Johnson, Zach Goodwin**

Students of  
University of Cincinnati  
College of Education, Criminal Justice, and Human Services  
School of Information Technology

April 2019

# Table of Contents

Section 1. Abstract.....	1
Section 2. Introduction .....	2
2.1 Background .....	2
2.2 Problem .....	4
2.3 Solution .....	5
2.4 Project Goals.....	6
2.5 Overview .....	8
Section 3. Project Management .....	9
3.1 Project Concept.....	9
3.2 Project Gantt Chart.....	10
3.3 Project Timeline.....	11
3.4 Budget and Other Financials .....	12
3.5 Project Communication Plan.....	16
Section 4. Product Development.....	17
4.1 Design Objectives.....	17
4.2 Methodology and Technical Approach.....	19
4.3 User Profile.....	22
4.4 Use Case Diagram .....	25
4.5 Technical Architecture and Discussion .....	26
4.6 Testing Approach .....	28
4.7 Application Screenshots .....	36
Section 5. Discussion.....	46
5.1 Problems Encountered .....	46
5.2 Future Recommendations .....	47
Section 6. Conclusion .....	49
6.1 Fall Semester 2018 Lessons Learned .....	49
6.2 Spring Semester 2019 Goals.....	50
6.3 Spring Semester 2019 Lessons Learned.....	51
6.4 Final Thoughts.....	52

# List of Illustrations

## Figures

<u>ITEM</u>	<u>PAGE</u>
Figure 1. Project Gantt Chart .....	10
Figure 2. Project WBS.....	12
Figure 3. Operating Budget .....	13
Figure 4. Subscription Model.....	15
Figure 5. Event Workflow .....	21
Figure 6. Use Case Diagram.....	25
Figure 7. System Architecture .....	26
Figure 8. Application Architecture .....	27
Figure 9. Sample QA Test Case .....	29
Figure 10. Test Plan.....	31
Figure 11. Example Test Case Results .....	34
Figure 12. EBNportal.com UI- employee home page .....	36
Figure 13. EBNportal.com UI- Chapter Leader home page .....	37
Figure 14. EBNportal.com UI- new event form .....	38
Figure 15. EBNportal.com UI- Program Manager home page .....	39
Figure 16. EBNportal.com UI- dashboard & reporting .....	40
Figure 17. EBNportal.com UI- event listing.....	41
Figure 18. EBNportal.com UI- event detail employee view .....	42
Figure 19. EBNportal.com UI- BRG chapter management form .....	43
Figure 20. EBNportal.com UI- VTO entry form for employees .....	44
Figure 21. EBNportal.com UI- employee CSV import.....	45
Figure 22. Team Poster .....	54

## Tables

<u>ITEM</u>	<u>PAGE</u>
Table 1. Team Collaboration Tools .....	16
Table 2. User Profile .....	22
Table 3. Operating Budget Breakdown .....	55

## Acronyms and Abbreviations

<b>EBN</b>	Employees Becoming Neighbors
<b>BRG</b>	Business Resource Group
<b>PM</b>	Program Manager
<b>CL</b>	BRG Chapter Leader
<b>HR</b>	Human Resources
<b>PTO</b>	Paid Time Off
<b>VTO</b>	Volunteer Time Off (typically paid time off)
<b>OWASP</b>	Open Web Application Security Project
<b>HTTPS</b>	Hyper Text Transfer Protocol (secured)
<b>TFS</b>	Microsoft Team Foundation Server

## Section 1. Abstract

Fortune 500 companies need a simple, single solution to help them track their progress towards their Social Responsibility goals. To make the social responsibility value come to life, companies allow their employees to take paid time off to volunteer. Companies also sponsor community outreach events via employee-led Business Resource Groups (BRGs). Today, most companies don't have a single solution or single source of truth that tracks where the company's money or employee volunteer time is going. To solve that problem, our team created Employees Becoming Neighbors. Our product is a "one-stop shop" for companies to track their employees' paid volunteer time off and give their BRGs a tool to manage their community events. Our product also gives company Executives one place to run reports to see how their company is meeting their goals. We are doing this because we wanted to make a product that meets a real-world need and applies to the private sector.

## Section 2. Introduction

### 2.1 Background

Most Fortune 500 companies list “corporate social responsibility” as one of their core values and goals. To achieve that, these companies will create a company-wide program and goal to “give back” with their time and money. To make this happen, companies will often create the following options:

- One survey showed 38% of companies allow employees to take “Volunteer Time Off” (VTO)<sup>1</sup>, time when an employee serves in the community for a day or half a day, but still gets paid as if he/she showed up to work a normal day. The employee fills out some paperwork to register their VTO time.
- Start a BRG program where employees form groups around some affinity (e.g., “U.S. Bank African-Americans of Cincinnati” chapter, “Verizon Women of Denver” chapter). Over 70% of large companies have some form of a BRG program<sup>2</sup>. These chapters will host community events, and the company will fund the event. For example, an event might be to paint some walls and a mural in a local elementary school. Large companies might have dozens of chapters around the country. Program Managers of the BRG program will approve and track the

---

<sup>1</sup> "How Creating a Volunteer Culture Can Help Your Business." Accessed October 14, 2018. <https://www.prnewswire.com/news-releases/how-creating-a-volunteer-culture-can-help-your-business-300470679.html>.

<sup>2</sup> "Are Employee Resource Groups Good for Business? - SHRM." Accessed October 14, 2018. <https://www.shrm.org/hr-today/news/hr-magazine/0916/pages/are-employee-resource-groups-good-for-business.aspx>.

hundreds of events, event expenses, and event attendance with the help of Chapter Leaders.

- Allow small teams (i.e., a group from a bank branch) to hold smaller, non-BRG community events, where the company will pay for the employees' time and provide funding for the events. For example, the company will pay for gloves, shovels, and mulch for a "park landscaping event". The team's manager will fill out some paperwork to register how much money was spent and how many hours the team volunteered.

Companies want to be able to report back to their customers, investors, and the media how he/she have helped the community. To do this, the company needs to track how, when, and where it spends its money and track how much time employees give serving.

## 2.2 Problem

Fortune 500 companies may not have a single way to track how their company spends its money or total company time employees use to volunteer in their communities under the name of the company. For example:

- Some companies may not make it easy for employees to track their individual volunteer time or for team managers to track their team's events.
- Unless it's built in-house, BRG Program Managers don't have a tool on the market that helps them track BRG-related events, attendance, and expenses. So many BRG programs resort to excel spreadsheets and e-mail as their "system".
- A Company Executive or Leader may find it difficult to figure out how much money the company is spending on corporate social responsibility or calculate total company-wide volunteer hours.
- The data collected for these efforts are often stored in multiple locations and databases.

## 2.3 Solution

To solve these problems, we created a “one-stop shop” for Fortune 500 companies for all things related to corporate social responsibility. This solution is called **EBNportal.com**.

**Ebnportal.com** is a system that provides the following:

- Employees can easily submit their time and details of their volunteer day.
- Team Managers can submit their events for approval, and track attendance, and expenses.
- BRG Program Managers can control the events that get approved and make sure BRG Chapters are tracking attendance and expenses.
- Provide company Executives one place to get reports on volunteer hours, expenses, and event attendance.
- Serve as a single source of truth regarding data.

## 2.4 Project Goals

Our goal was to build a subscription-based SaaS offering that Fortune 500 companies can sign up for. **EBNportal.com** is an Internet-facing Web application. We built this under the umbrella of a fictitious company “Employees Becoming Neighbors”, where our company is a third-party vendor, and our customers are the Fortune 500 companies.

**EBNportal.com** serves as a central hub to allow companies to do the following:

- Provide employees a simple way to register their Volunteer Day details. The data will be available to company Leaders in reporting.
- Give Team Managers a place to submit their events data (e.g., who attended, expense data). The data will also be available to company Leaders in reporting.
- Give BRG Program Managers and Chapter Leaders a place to manage events. Program Managers will be able to approve or reject events. For approved events, Chapter Leaders will track who attended, and track expenses. Employees will be able to search for events, and RSVP to the event. The system will notify the employee of the upcoming event days before it happens. Employees can also “follow” a chapter that interests them.
- Provide company Executives a single source of truth for their reporting needs when tracking their social responsibility objectives.

**EBNportal.com** is a multi-tenant system, meaning each company will have its own dataset in the system not accessible by other companies. The system is secured with a login authentication mechanism to determine which company the person works for. The system also enforces page-level authorization to block non-approved users (e.g., certain pages can be accessed only by Program Managers).

## 2.5 Overview

This remainder of this report describes in detail how we completed this project. This report covers the following topics:

- project management topics like project timeline and budget.
- production development topics like design and architecture.
- a discussion of what we learned and future considerations.

## Section 3. Project Management

### 3.1 Project Concept

We wanted to do a project that addressed a real-world need and something that targets companies in the private sector.

This project idea came about after James Allen had discussions with people who work in Human Resources at his employer U.S. Bank. Employees who manage the Business Resource Group program faced difficulty managing their program. The idea of adding a separate component of allowing employees to enter regular volunteer hours came about from feedback from Professor Jim Scott, based on his time working in the private sector.

### 3.2 Project Gantt Chart

Figure 1. Project Gantt Chart below shows our estimated project timeline for fall and spring semesters.

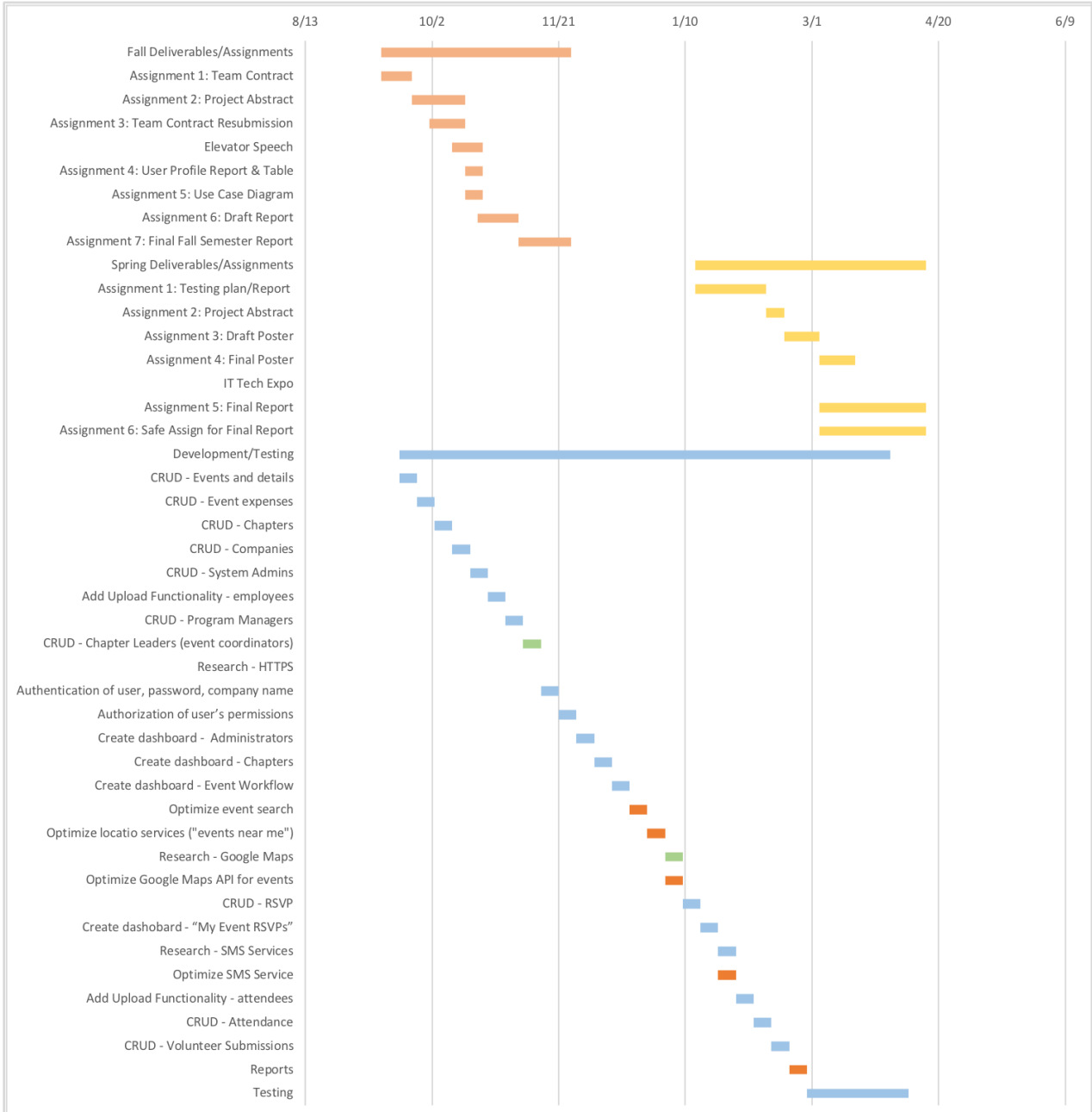


Figure 1. Project Gantt Chart

### 3.3 Project Timeline

**Figure 2. Project WBS** below shows the estimated work breakdown structure (WBS) and start and end dates we estimated in the fall for our iterative development.

Task Description	Start Date	End Date	Duration in days
<b>Fall Deliverables/Assignments</b>	<b>9/12</b>	<b>11/26</b>	<b>75</b>
Assignment 1: Team Contract	9/12	9/24	12
Assignment 2: Project Abstract	9/24	10/15	21
Assignment 3: Team Contract Resubmission	10/1	10/15	14
Elevator Speech	10/10	10/22	12
Assignment 4: User Profile Report & Table	10/15	10/22	7
Assignment 5: Use Case Diagram	10/15	10/22	7
Assignment 6: Draft Report	10/20	11/5	16
Assignment 7: Final Fall Semester Report	11/5	11/26	21
<b>Spring Deliverables/Assignments</b>	<b>1/14</b>	<b>4/15</b>	<b>91</b>
Assignment 1: Testing plan/Report	1/14	2/11	28
Assignment 2: Project Abstract	2/11	2/18	7
Assignment 3: Draft Poster	2/18	3/4	14
Assignment 4: Final Poster	3/4	3/18	14
IT Tech Expo	4/9	4/9	0
Assignment 5: Final Report	3/4	4/15	42
Assignment 6: Safe Assign for Final Report	3/4	4/15	42
<b>Development/Testing</b>	<b>9/19</b>	<b>4/1</b>	<b>194</b>
CRUD - Events and details	9/19	9/26	7
CRUD - Event expenses	9/26	10/3	7
CRUD - Chapters	10/3	10/10	7
CRUD - Companies	10/10	10/17	7
CRUD - System Admins	10/17	10/24	7
Add Upload Functionality - employees	10/24	10/31	7
CRUD - Program Managers	10/31	11/7	7
CRUD - Chapter Leaders (event coordinators)	11/7	11/14	7
Research - HTTPS	11/14	11/21	
Authentication of user, password, company name	11/14	11/21	7
Authorization of user's permissions	11/21	11/28	7
Create dashboard - Administrators	11/28	12/5	7
Create dashboard - Chapters	12/5	12/12	7
Create dashboard - Event Workflow	12/12	12/19	7
Optimize event search	12/19	12/26	7
Optimize locatio services ("events near me")	12/26	1/2	7
Research - Google Maps	1/2	1/9	7
Optimize Google Maps API for events	1/2	1/9	7
CRUD - RSVP	1/9	1/16	7
Create dashobard - "My Event RSVPs"	1/16	1/23	7
Research - SMS Services	1/23	1/30	7
Optimize SMS Service	1/23	1/30	7
Add Upload Functionality - attendees	1/30	2/6	7
CRUD - Attendance	2/6	2/13	7
CRUD - Volunteer Submissions	2/13	2/20	7
Reports	2/20	2/27	7
Testing	2/27	4/1	40

*Figure 2. Project WBS*

### 3.4 Budget and Other Financials

**Figure 3. Operating Budget** shown on the next page shows the estimated costs of building this system. It also includes expenses relating to staffing, hardware, training, and other company-related expenses.

Our total 1-year forecast is approximately \$1.5 million. Most of the budget goes toward paying full-time employees and contractors to develop the system as well as other personnel. Our plan uses contractor services for the first four (4) months of system development. We also accounted for cloud hosting costs, development tools, personnel training, and laptops people use to work on.

In our case however, as students of this Senior Design class, our actual costs were less than \$200. Most of the \$200 was used for our hosting service, HostGator who hosted our web application and database. The rest went towards minor expenses for the IT Expo like supplemental handouts and candy for visitors to our booth.

See the appendix for a detailed, monthly breakdown of this operating budget. The figure on the next page focuses just on the annual operating budget.

			Annual TOTAL	Annual no contractors TOTAL
<b>MSDN</b>				
<b>Human Resources Costs</b>	<b>License?</b>	<b>Cost Per Hour</b>		
<b>FTEs</b>				
Business Analyst	x	\$65	\$ 135,096	\$ 135,096
Senior Application Developer	x	\$80	\$ 166,272	\$ 166,272
Database Admin/Developer	x	\$80	\$ 166,272	\$ 166,272
Manager / System Administrator	x	\$95	\$ 197,448	\$ 197,448
Business Development / Account Manager		\$80	\$ 166,272	\$ 166,272
<b>Contractors</b>				
Developer	x	\$100	\$ 69,280	\$ -
Developer	x	\$100	\$ 69,280	\$ -
<b>Technical Costs</b>				
Azure PaaS Cloud Hosting		See note		
Azure Web Application Hosting			\$ 6,648	\$ 6,648
Azure Database Hosting			\$ 29,160	\$ 29,160
<b>Hardware</b>				
Workstations count		See note		
Workstations cost			\$ 9,333	\$ 9,333
<b>Development Tools</b>				
MSDN Licenses		See note		
MSDN Licensing			\$ 14,000	\$ 14,000
<b>Other Costs</b>				
Professional Services		See note	\$ 240,000	\$ 240,000
<b>Training</b>				
Pluralsight seats needed		See note		
Pluralsight subscription			\$ 4,500	\$ 4,500
			<b>\$ 1,273,561</b>	<b>\$ 1,135,001</b>

Human Resources Costs	<b>76%</b>	73%
Technical Costs	<b>5%</b>	5%
Other Costs	<b>19%</b>	22%

**Assumptions**

- \* We hire contractors for first 4 months during initial development.
- \* FTE hourly cost assumes loaded cost to cover taxes, benefits, etc.. FTE will not gross that hourly rate, instead approx. 60% of that rate.
- \* Contractor hourly cost assumes the contracting company rate, and that company will handle their own benefits, taxes, profit, etc..

**Notes**

- \* Azure costs based off of Azure calculator, 730 hours per month <https://azure.microsoft.com/en-us/pricing/calculator/>  
 Web App Hosting = Azure App Service S3, 4 cores, 7GB RAM, 2 instances, 1 SSL connection, West US, Linux-based, Standard Tier  
 Database Hosting = Azure Database for MySQL, Gen 4 16v core, 2 servers, 100GB storage, 100 GB Geo-Redundant storage
- \* Hardware = 1 business-grade laptop per FTE and Contractor, service contract, \$4,000 each, capitalizable across the 3 years
- \* Development Tools = MSDN Enterprise license, \$250 per developer per month <https://visualstudio.microsoft.com/vs/pricing/>
- \* Professional Services = Hiring external providers for Office 365 cloud (email), design, marketing, accounting, legal, HR, etc.
- \* Training = Pluralsight license, \$750/ user per year, \$62.50/month <https://www.pluralsight.com/pricing>

**Figure 3. Operating Budget**

**Figure 4. Subscription Model** shown on the next page shows how our company could offer **EBNportal.com** as Product as a Service offering and how we could structure our subscription pricing model.

The subscription model is broken up into two parts. First, each customer will pay for each Program Manager Application User assigned in the account. Second, each customer would pay for each employee record stored in the system.

This model also includes an estimated return-on-investment for customers by showing how much money they could potentially save when you consider this system will simplify and automate much of a Program Manager's work processes and daily routines.

<b>Subscription Rates Per Month</b>	
Subscription rate is made of up 2 parts.	
Cost per End User- Program Manager role	<b>\$695</b>
Cost per End User- Chapter Leader role	<b>included</b>
Cost per End User- Employee role	<b>included</b>
Cost per employee record stored in system	
1-10,000 employees	<b>\$0.149</b>
10,001+ employees	<b>\$0.119</b>

<b>Example small-sized company</b>	
1 Program Manager	\$695
100 employees	\$15
	<hr/>
	\$710 per month
	<b>\$8,519 per year</b>

<b>Example medium-sized company</b>	
2 Program Managers	\$1,390
1,000 employees	\$149
	<hr/>
	\$1,539 per month
	<b>\$18,468 per year</b>

<b>Example large enterprise</b>	
4 Program Managers	\$2,780
10,000 employees	\$1,190
	<hr/>
	\$3,970 per month
	<b>\$47,640 per year</b>

<b>How much does our system save a company?</b>	
Hours per week each PM uses doing manual work?	25
Hours per week, same work, using our system?	4
Hours per week saved using our product?	21
How much \$\$ is saved, per PM?	
(\$50 per hour, loaded hourly rate of a PM)	\$1,050
	<hr/>
	<b>\$54,600 per year</b>

Amount saved by small-sized example, 1 PM	<b>\$46,081 per year</b>
Amount saved by medium-sized example, 2 PMs	<b>\$90,732 per year</b>
Amount saved by large enterprise example, 4 PMs	<b>\$170,760 per year</b>

**Figure 4. Subscription Model**

### 3.5 Project Communication Plan

Our communication plan included meeting in person after class every Monday, meeting again online every Thursday evening, and using a mobile chat app for anything that can't wait for a following meeting. Occasionally we would hold impromptu online meetings if something important came up.

**Table 1. Team Collaboration Tools** below lists the tools we used across various communication channels.

Communication Method	Tool
Realtime chat	GroupMe mobile app
E-mail to each other	Personal Gmail account
E-mail to faculty	School e-mail account
Product Backlog, Sprint Board	Microsoft Azure Devops cloud service
Online meetings	Google Hangouts cloud service
Online meeting with faculty	Faculty's Webex meeting room
Documents shared drive	Google Drive cloud service
Source code repository	Github.com cloud service

***Table 1. Team Collaboration Tools***

## Section 4. Product Development

### 4.1 Design Objectives

EBNPortal.com is a Web-based application. Since customers will be accessing this app using a Web browser, we built this product with mobile responsiveness, security, and multi-tenancy.

The mobile-responsiveness objective deals with the allowing a user to use the application on a browser on different screen sizes you find on laptops, large monitors, phones, and tablets.

The security objective deals with designing the application to have strong defenses against various attacks. We analyzed the list of Open Web Application Security Project (OWASP) Top Ten security vulnerabilities for Web applications in 2017. We took our analysis of the OWASP Top Ten 2017 and designed our application using defense mechanisms recommended by OWASP.

For example, SQL Injection is a common vulnerability in Web applications that OWASP identifies as a top concern with major risk. A malicious user could use an input box on a Web page and inject SQL commands by constructing special lines of text, and if successful that user could cause damage to the database or even steal data.

We used recommended defense mechanisms to sanitize all inputs so that any specially built strings of text cannot be executed as code but treated only as text input.

The multi-tenancy objective deals with the fact that we want many customers to use our application but never allow one customer to see another customer's data. In our case, a customer is a Fortune 500 company. For example, if Procter & Gamble (P&G) and Fifth Third Bank are both customers of Employees Becoming Neighbors, we ensure that P&G employees will never see Fifth Third events and data, and vice versa.

There are some other considerations in our product. We wanted to make sure the user's experience is not painful or frustrating. We designed our product to, hopefully, be simple enough that most employees can figure out how to use the product with minimal training or documentation.

For general development, we used an iterative approach. Rather than designing the entire system all up front before we begin construction, we broke the project down into many smaller pieces. With each piece of functionality we developed, we only designed, built, tested, and deployed that entire piece before moving onto the next.

## 4.2 Methodology and Technical Approach

The main goals of this app are to allow employees to enter time he/she volunteered, managers of teams to enter hours their team volunteered, provide event management tools for BRGs, and reporting for executives.

To build this product, we chose PHP as our language for the server-side application. We chose PHP because we have a team of mixed experience levels in coding, and PHP provides an easy learning curve for new and less experienced developers. In addition, Micah Johnson has experience with PHP from a job co-op she had, and James Allen used PHP eight years ago. We went with PHP version 7 because PHP version 5 will soon stop getting support from its creators.

For the database, we chose MySQL. We chose MySQL for the same reason we chose PHP, because it has an easy learning curve for new and less experienced database developers.

For hosting we chose HostGator because of its low pricing for hosting, since we are not a real company with actual funding. We also used the LetsEncrypt program to get a free certificate for HTTPS protocol hosting, since we do not want to use HTTP. Finally, we also bought the domain name **EBNportal.com** from HostGator as well.

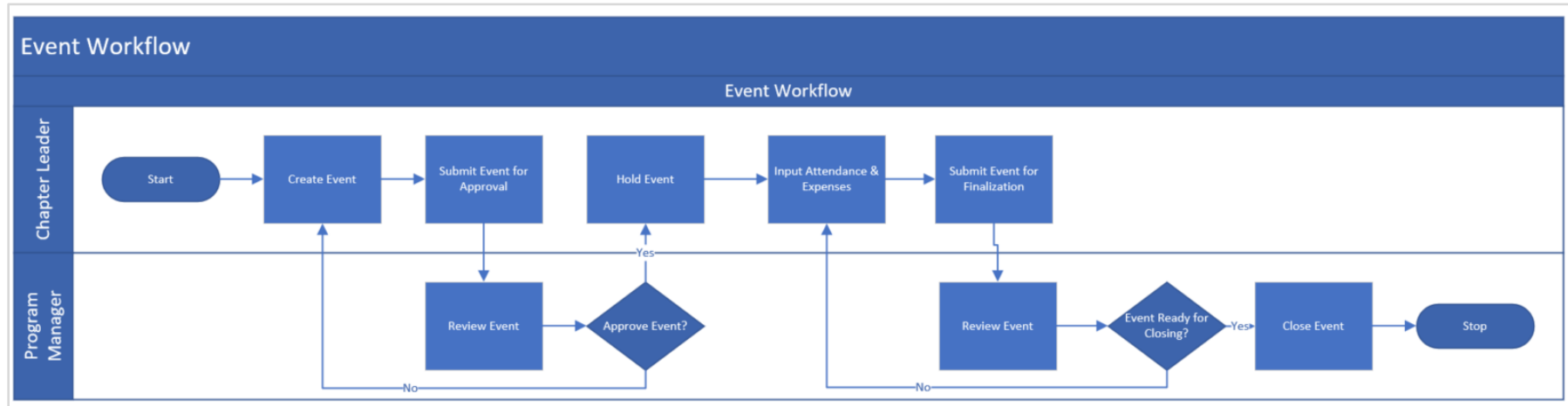
For authentication, we hashed our passwords with a salt using the BCrypt algorithm. We stored plain text passwords only during the early phases of development but went back

and rewrote the database and application code to always use hashing. We made sure all test accounts were deleted and recreated using the hashing approach.

For reporting, we chose to write PHP code that would generate Excel reports on-demand. We also created a dashboard that gives authorized users a quick glimpse of key metrics.

In terms of the product itself, while most of the application uses a “forms over data” paradigm, for event management, we had to build into our product the concept of a workflow for events. For example, a Chapter Leader may create an event, but the Program Manager in Human Resources (HR) must approve the event. There are several steps in the workflow for an event. **Figure 5. Event Workflow** on the next page shows the various phases and approvals an event must go through.

**Figure 5. Event Workflow** below shows the various phases of an event.



**Figure 5. Event Workflow**

The two main actors involved with the workflow of an event are the Chapter Leader (CL) and the Program Manager (PM). The CL creates the event and submits it for approval. The PM reviews the event and either approves or rejects it. After an event is approved, the event occurs. After the event is over, the CL updates the event record with attendance and expense details and submits it for finalization. The PM reviews the event details and determines if it can be closed.

## 4.3 User Profile

We determined that multiple types of users will interact with our application. The potential users for EBNportal.com are employees of a company looking to go to BRG events and volunteer their time on behalf of their employer. Another user would be a BRG Chapter Leader who plans the events and tracks who attends the events. Other users include Program Managers who approve the events and oversee the company's volunteer strategy. **Table 2. User Profile** below shows a detailed breakdown and analysis of our user profile.

<b>Application:</b>	
Employees Becoming Neighbors	<i>ebnportal.com</i>
<b>Potential Users:</b>	
<ul style="list-style-type: none"> <li>• Business resource group program Managers in Human Resources wanting to manage their business resource group program</li> <li>• Business resource group chapter Leader wanting to track their chapter events</li> <li>• An employee looking for an event to serve on, and looking to register their Volunteer Time Off</li> <li>• Team Manager who needs a place to register their team's outreach event</li> <li>• Company leaders looking for data and reports</li> </ul>	

***Table 2. User Profile***

**Software, Interface, and Related Experience:**

This project will be targeted at business and will serve as a central core to manage their events. Each company would have its own “zone” in the system not accessible by other companies. Key characteristics will include event tracking, “upcoming events” listings, event sign-ups, attendance tracking, reminder e-mail notifications, expense tracking, and reporting.

**Experience with Similar Applications:**

- CVent
- RegPack
- Gather

**Task Experience:**

- Using a Web Browser to access the ebnportal.com Web site
- Being an employee of a company and adding or withdrawing yourself from events
- Chapter leaders and project managers managing events

**The frequency of Use:**

Whenever companies want to add new events to correlate with other companies. This can be daily, weekly, monthly, or annually.

***Table 2. User Profile (continued)***

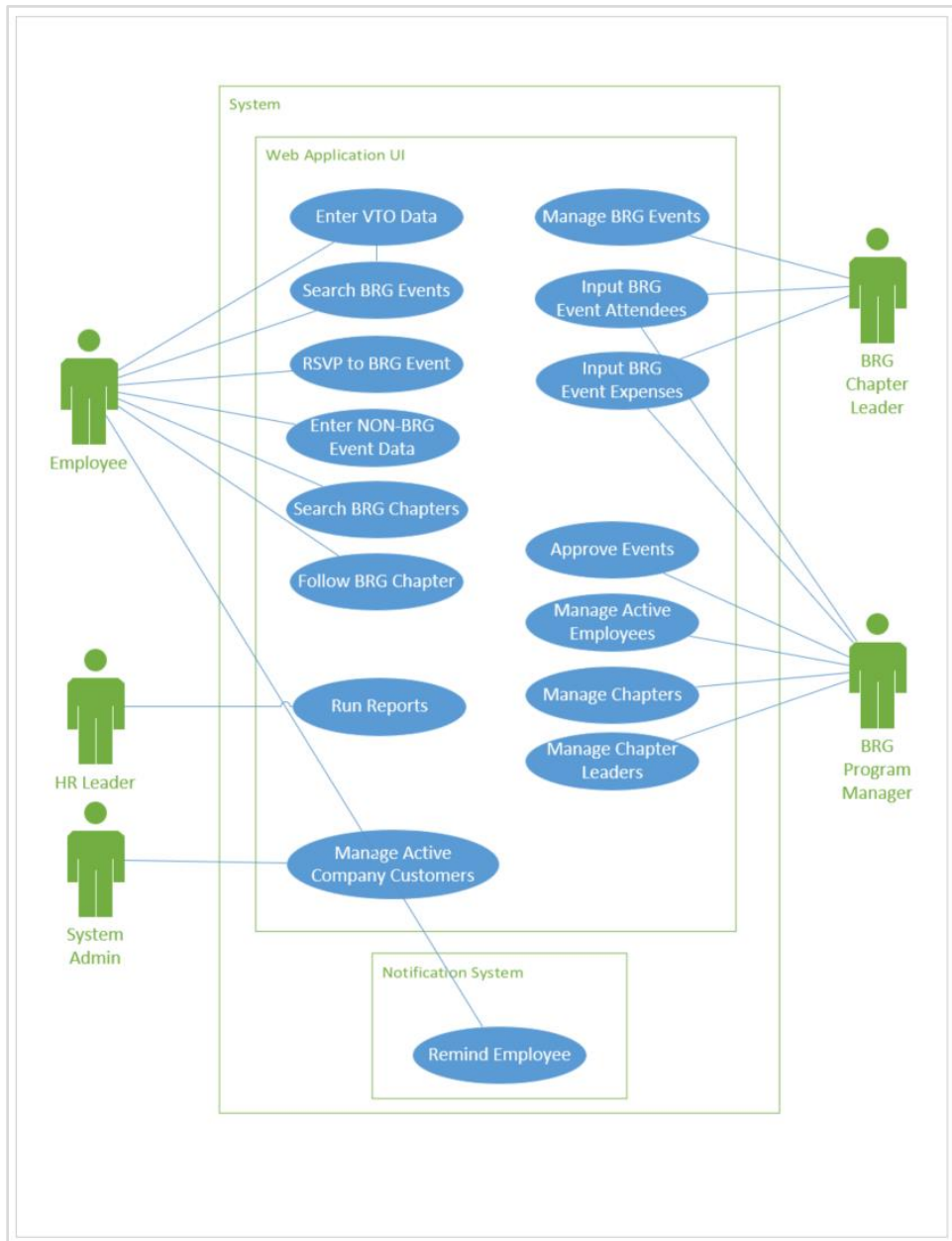
**Key Interface Design Requirements that the Profile Suggests:**

- CRUD for events, event expenses, chapters, companies, system admins, chapter leaders, and program managers.
- Authentication of user, password and company name
- Authorization of user permissions
- Admin dashboard showing events waiting for approval and finalization

***Table 2. User Profile (continued)***

## 4.4 Use Case Diagram

**Figure 6. Use Case Diagram** below shows the basic functionality each user type has with the application.

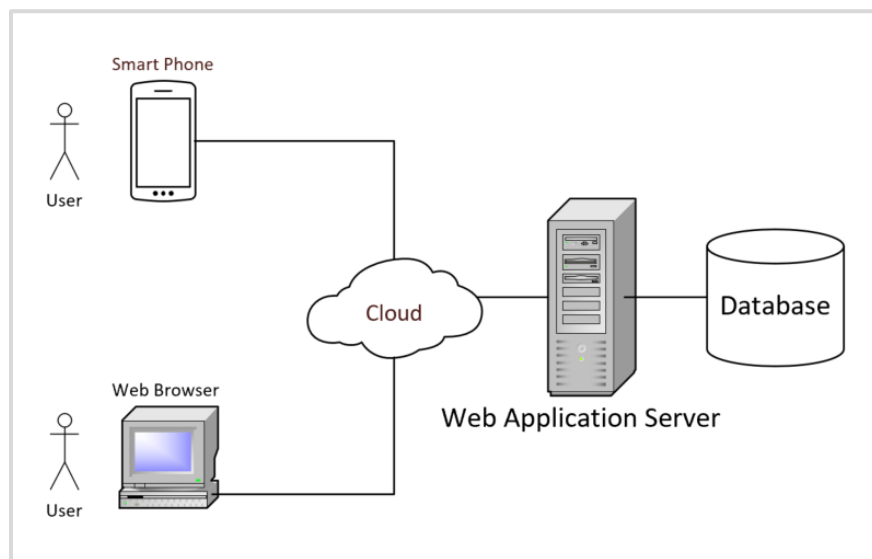


**Figure 6. Use Case Diagram**

## 4.5 Technical Architecture and Discussion

The architecture we chose was a standard 2-tier design where we have the PHP web application on the first tier, and the database on the second. Our system does not include any other dependencies like external web services. At this time, user authentication credentials and authorization are all handled within the system.

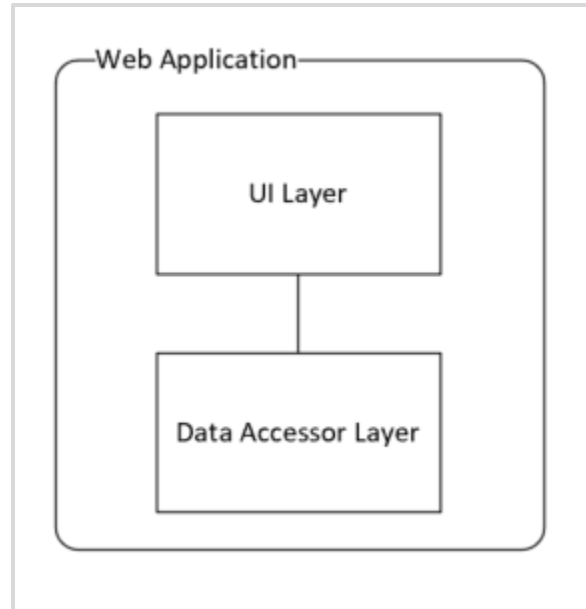
**Figure 7. System Architecture** below shows the 2-tier design of the whole system including web application and database. It also shows how it relates to the end user, their preferred tool to browse to our system, and the cloud.



**Figure 7. System Architecture**

The web application architecture has a 2-layer design where we have a UI component layer, and a Data Accessor layer.

**Figure 8. Application Architecture** below shows the 2-layer design of the application which includes the UI layer and the Data Accessor layer.



**Figure 8. Application Architecture**

## 4.6 Testing Approach

### Overview and Methodology

For our testing methodology, we used two approaches as our overall testing strategy:

1. Manual quality assurance testing with documented test cases
2. Team-based roundtable ad-hoc “exploratory” testing

For our first approach, we used a manual quality assurance (i.e., QA) testing approach. When we created a feature in our system, we also created one or more QA Test Cases in Microsoft Team Foundation Server (TFS).

QA Testing workflow:

1. Both the Developer and Test Lead share responsibility in creating QA Test Cases.
2. The Developer will run the QA Tests as he/she builds the feature, to help capture issues during design/coding time.
3. Once a Developer releases the code that pertains to a feature, that Developer marks the relevant QA Tests as “Ready”.
4. The Test Lead or another Developer will run the QA Tests.
5. If the QA Test fails, the tester will open a new Bug work item and inform the team.

It is important to have a “separation of concerns” with QA Testing. We made sure that someone *else besides the original Developer* ran the same QA Tests, to get a “second pair of eyes” on the system. For example, If Micah built a feature, we would also have Zach or James run the same tests. Software Developers tend to have a cognitive bias

where they accidentally overlook an issue with their own development as they “test their own code”.

When we created QA Test Cases, we first account for the “happy path”, that is the path an End User would take through a workflow without encountering any issues or errors. For example, to test each step of the workflow of creating an event, we created QA Test Cases for each step throughout the event creation workflow within validation rules.

We also included some QA Test Cases for error conditions. For example, we had test cases for making sure the system handles bad dates in the “create event” workflow.

**Figure 9. Sample QA Test Case** below is a screenshot showing an example of a QA Test Case with test steps:

TEST CASE 463\*

463 Employee VTO time entry

James Allen 0 comments Add tag

State: ● Ready Area: EventManager  
Reason: Completed Iteration: EventManager

**Steps**

Steps	Action	Expected result
1.	Load EBNPortal home page	
2.	Log in as Tester1	
3.	Load Employee VTO time entry form	You should see an approved time entry for 1/2/19, 2 hours
4.	Enter 2 for hours	
5.	Pick a date in the past	
6.	Enter some text in the	
7.	Click SUBMIT button	The form should indicate that your entry was submitted and pending approval

**Figure 9. Sample QA Test Case**

For our second approach, we also did a couple rounds of “round table” ad-hoc exploratory testing. We scheduled a couple of 1-hour team meetings via Google Hangouts. When we met, we each “played around” by exploring the system trying different things to see if we can “break” the system.

## Scope of Testing

Our Test Cases covered all major features we are building in this system. Our test strategy included testing for each of the End User Roles in the system using a web browser. It would be very ineffective to *only* conduct testing as a Chapter Leader user but *not* conduct testing as a Program Manager.

**Figure 10. Test Plan** below shows part of our overall suite of Test Cases in TFS to date:

Project: EventManager Server: jamesallen.visualstudio.com Query: Test Cases List type: Flat			
ID	Work Item Type	Title	State
463	Test Case	Employee VTO time entry (Employee)	Ready
464	Test Case	Manager VTO time entry approval (Employee)	Ready
465	Test Case	Manager VTO Time entry rejection (Employee)	Ready
466	Test Case	Create new event (Chapter Leader)	Ready
467	Test Case	New event form only allows future dates	Ready
468	Test Case	Submit event for approval (Chapter Leader)	Ready
469	Test Case	Approve event (Program Manager)	Ready
470	Test Case	Reject Event (Program Manager)	Ready
471	Test Case	Display list of all pending approvals (Program Manager)	Ready
472	Test Case	Show list of events pending approval (Chapter Leader)	Ready
473	Test Case	Show list of upcoming approved chapter events (Chapter Leader)	Ready
474	Test Case	Edit event (Chapter Leader)	Ready
475	Test Case	Delete an event (Chapter Leader)	Ready
476	Test Case	Delete an event (Program Manager)	Ready
477	Test Case	Upload manual attendance (Chapter Leader)	Ready
478	Test Case	Upload attendance using bulk CSV list	Ready
479	Test Case	System shows error for bad CSV imports	Design
480	Test Case	My chapters list (Chapter Leader)	Design
481	Test Case	Upload list of current employees (Program Manager)	Design
482	Test Case	Show list of new events not submitted yet (Chapter Leader)	Ready
483	Test Case	Search for events (Employee)	Design
484	Test Case	RSVP to event (Employee)	Design
485	Test Case	Receive email reminder of RSVP event (Employee)	Design
486	Test Case	See details of event (Employee)	Design
489	Test Case	Non Program Managers cannot approve or reject events	Design
490	Test Case	VTO Approvals- can only see Direct Report submissions	Design
491	Test Case	Report loads data	Design
492	Test Case	Update employee's role to Chapter Leader	Design
493	Test Case	Update employee's role to Program Managers	Design
494	Test Case	Remove employee from Chapter Leader role	Design
495	Test Case	Remove employee from Program Manager role	Design

**Figure 10. Test Plan**

## Objectives

Our Test Strategy had to accomplish the following:

1. All major features must have QA Test Cases covering every logical End User step of the “happy path” workflow, and some of the error conditions as well.
2. The entire collection of QA Test Cases must account for all End User roles (e.g., Program Manager, Chapter Leader, Employee, Report Reader).
  - a. Each *individual* QA Test Case does *not* have to account for all roles, but the overall *collection* of QA Test Cases must. For example, one Test Case can deal with the Program Manager role only.
3. All failed QA Test Cases and issues found during Exploratory Testing must result in a new Bug work item, or a plan in writing to address the defect.
4. All Bug work items are reviewed, fixed, and closed before IT Expo.
5. All QA Test Cases must eventually pass before the final release and IT Expo.

### **Logging Test and Procedures**

As stated above, we took two approaches, manual QA Test Cases, and team-based roundtable Exploratory Testing.

Regardless of which of the two approaches we worked on, if we found an issue or a bug, we opened a Bug work item in TFS. Then one of the Developers assigned themselves the Bug work item, reviewed it, and fixed the bug.

We kept our QA Test Cases, Test Run data (e.g., pass or fail), and Bug work Items in TFS because TFS gave us a nice development experience to help us manage Test Cases and track test results.

Each team member was required to perform some QA Testing each week with the goal of getting every QA Test eventually executed. We did not assign required times for each person to perform QA Testing, as we wanted to leave it up to each person to decide what time worked best for them.

For Team Roundtable Exploratory Testing, we had the following meetings:

- 2/13/19, 9:00 pm - 10:00 pm
- 3/6/19, 9:00 pm - 10:00 pm

## Test Results

Testing in agile projects is different than in waterfall projects. We don't have "exit criteria" because there is no exiting from one stage in a project to another in a waterfall sense.

Instead we simply write the tests, code, run the tests, fix bugs, and then run the tests again until they pass. At the end of our project, 100% of our tests passed.

**Figure 11. Example Test Case Results** shows an example of our tool that shows results of our QA Test Plan.

Test suite: EBN-TestPlan (Suite ID: 488)

Tests Charts

+ New Add existing [X] [Print] [Refresh] [Refresh] [Run] [Play] [Refresh] [Check] [X] [Minus] [Pause] [Menu]

Outcome	Order ↑	ID	Title	State
✓ Passed	1	463	Employee VTO time entry (Employee)	⋮ Ready
✓ Passed	2	464	Manager VTO time entry approval (Employee)	Ready
✓ Passed	3	465	Manager VTO Time entry rejection (Employee)	Ready
✓ Passed	4	466	Create new event (Chapter Leader)	Ready
✗ Failed	5	467	New event form only allows future dates	Ready
✓ Passed	6	474	Edit event (Chapter Leader)	Ready
✓ Passed	7	482	Show list of new events not submitted yet (Chapter Leader)	Ready
✗ Failed	8	468	Submit event for approval (Chapter Leader)	Ready
✓ Passed	9	469	Approve event (Program Manager)	Ready
✓ Passed	10	470	Reject Event (Program Manager)	Ready
✓ Passed	11	471	Display list of all pending approvals (Program Manager)	Ready
✓ Passed	12	472	Show list of events pending approval (Chapter Leader)	Ready

**Figure 11. Example Test Case Results**

## What We Learned During Testing

We learned that our system had many bugs in it, requiring a lot of refactoring. As a result, we spent a lot more time this semester doing refactoring and less time building new features that we would have preferred.

That is especially true when you consider our team's small size, our various levels of experience in web and database development, and our limited capacity in lieu of other commitments outside of this class (e.g., full-time jobs, other courses, family).

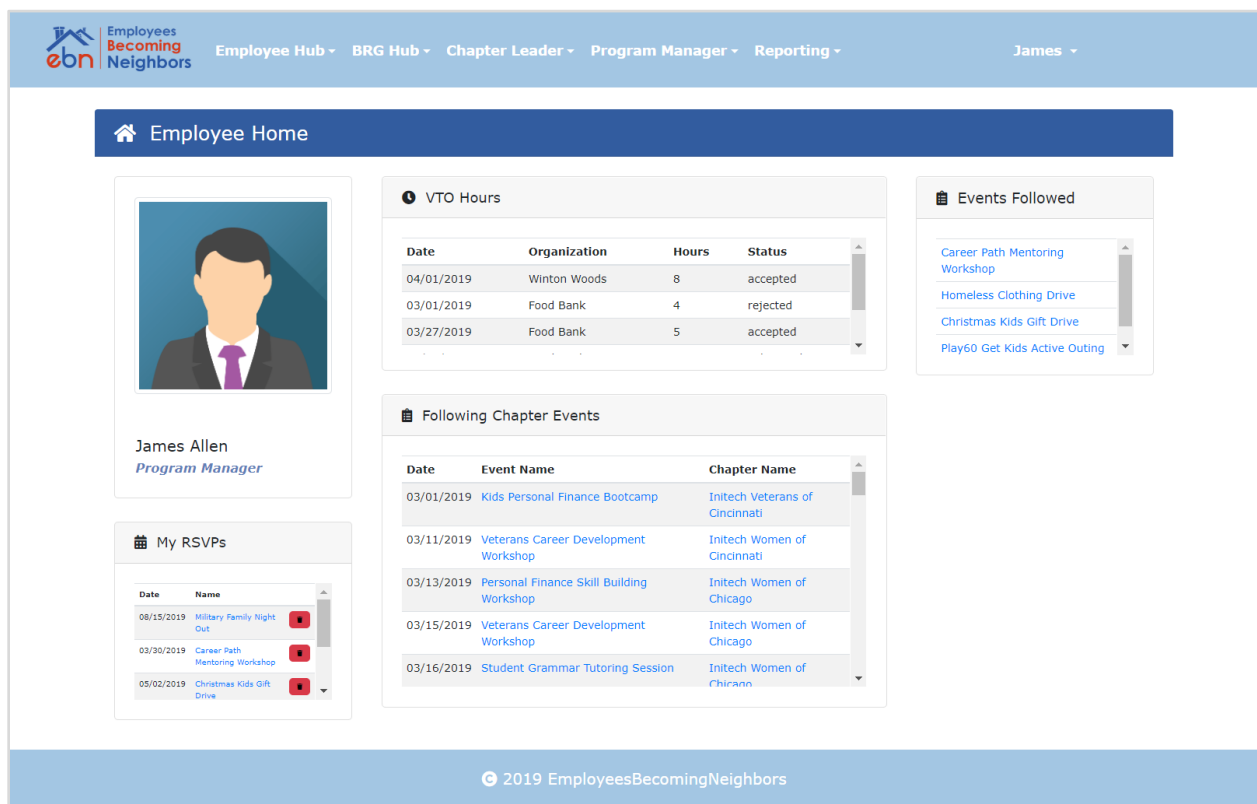
We also learned how much time it takes to do manual testing. It took a lot. If we could do this all over again, we would probably investigate a development ecosystem like .NET and an architecture like Model-View-Controller. These approaches would have given us a development environment that is more compatible with automated testing while still giving us an easy learning curve for team members with little programming experience.

Finally, on the bright side, because of testing we learned where to strengthen our system. As a result, when we presented at Tech Expo, we had more confidence that **Employees Becoming Neighbors** will stand up to the expectations of the Judging Panel and Faculty members. In other words, the more bugs we found during testing and fixed before Tech Expo, the better our system would be. Testing *while* developing a system is an absolute must.

## 4.7 Application Screenshots

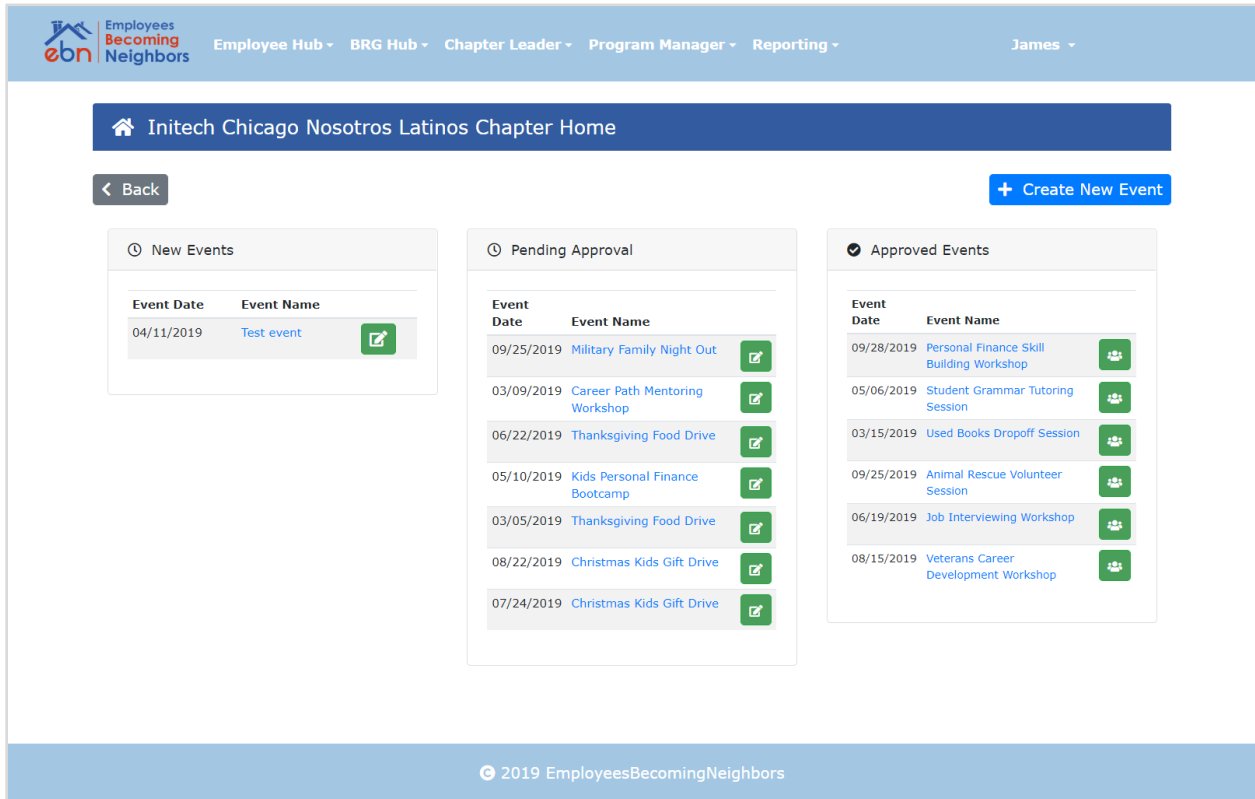
This section provides various screenshots of our product. This section will not cover every screen. Instead this section will cover major aspects of the product.

**Figure 12. EBNportal.com UI- employee home page** below shows the employee landing page a user sees after logging in.



**Figure 12. EBNportal.com UI- employee home page**

**Figure 13. EBNportal.com UI- Chapter Leader home page** below shows the Chapter Leader home page for a chapter they are assigned to.



**Figure 13. EBNportal.com UI- Chapter Leader home page**

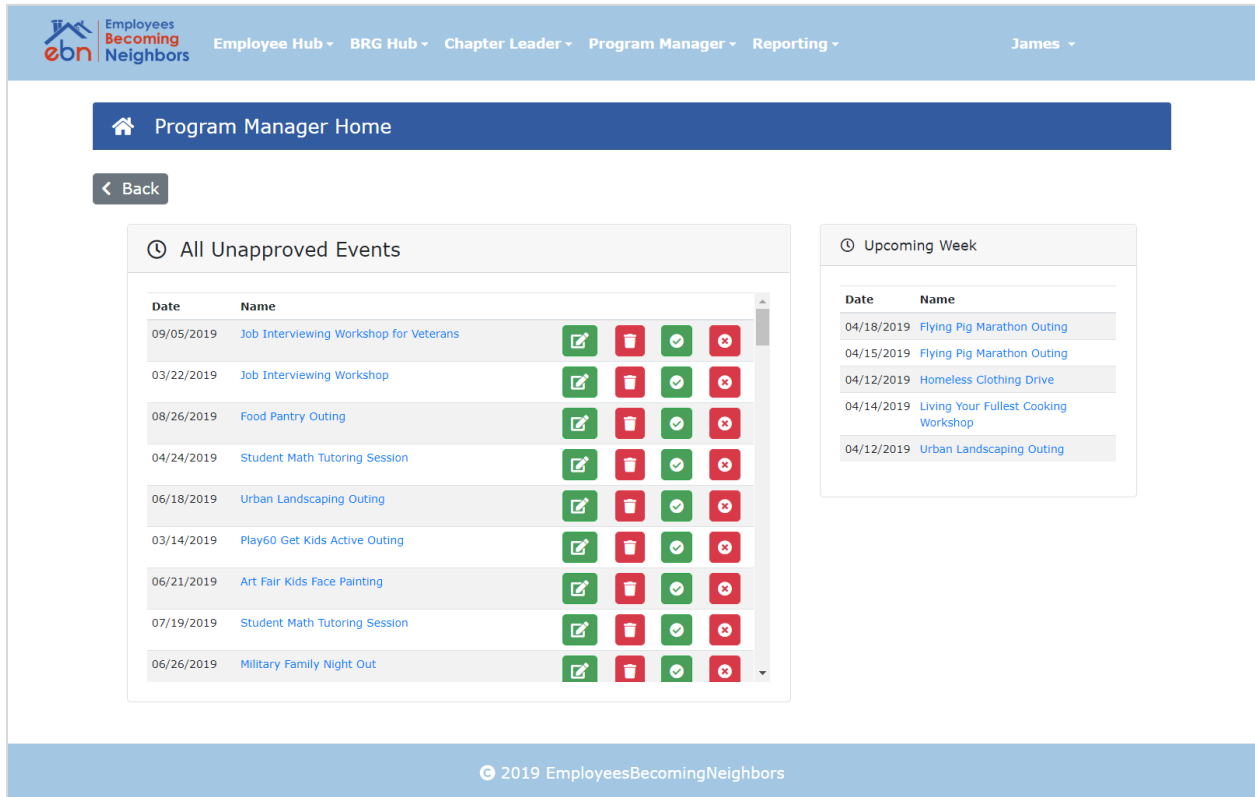
**Figure 14. EBNportal.com UI- new event form** below shows the “Add New Event” form Chapter Leaders use to create new events.

The screenshot shows a web application interface for adding a new event. At the top, there is a blue navigation bar with the logo for 'Employees Becoming Neighbors' (EBN) and several menu items: 'Employee Hub', 'BRG Hub', 'Chapter Leader', 'Program Manager', and 'Reporting'. The user's name 'James' is displayed on the right side of the navigation bar. Below the navigation bar, there is a 'Back' button. The main content area is titled 'Enter event details' and contains the following form fields:

- Select Chapter \***: A dropdown menu with 'Initech Chicago Nosotros Latinos' selected.
- Event Title \***: A text input field.
- Event Date \***: A date input field with the placeholder 'mm/dd/yyyy'.
- Event Start Time \***: A time input field with the placeholder '--:-- --'.
- Event End Time \***: A time input field with the placeholder '--:-- --'.
- Description**: A large text area for entering the event details.
- Address 1**: A text input field for the primary address.
- Address 2**: A text input field for a secondary address.
- City**, **State**, and **Zip**: Three separate text input fields for location information.

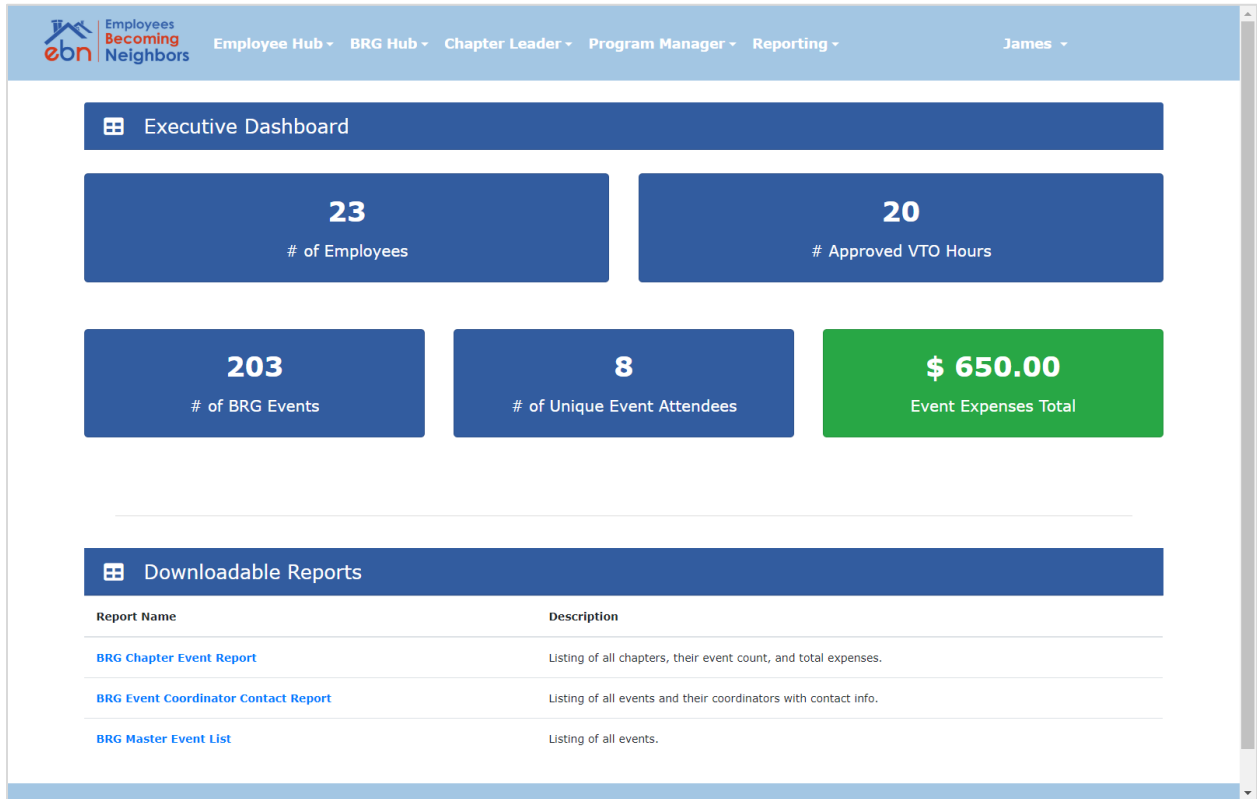
**Figure 14. EBNportal.com UI- new event form**

**Figure 15. EBNportal.com UI- Program Manager home page** below shows the Program Manager’s homepage. It gives them a view into what events need to be reviewed for approval, and a list of upcoming events.



**Figure 15. EBNportal.com UI- Program Manager home page**

**Figure 16. EBNportal.com UI- dashboard & reporting** below shows the Executive Dashboard and Reports page. A user with the permission to view this page gets a snapshot of important stats and has access to download reports with latest data.



**Figure 16. EBNportal.com UI- dashboard & reporting**

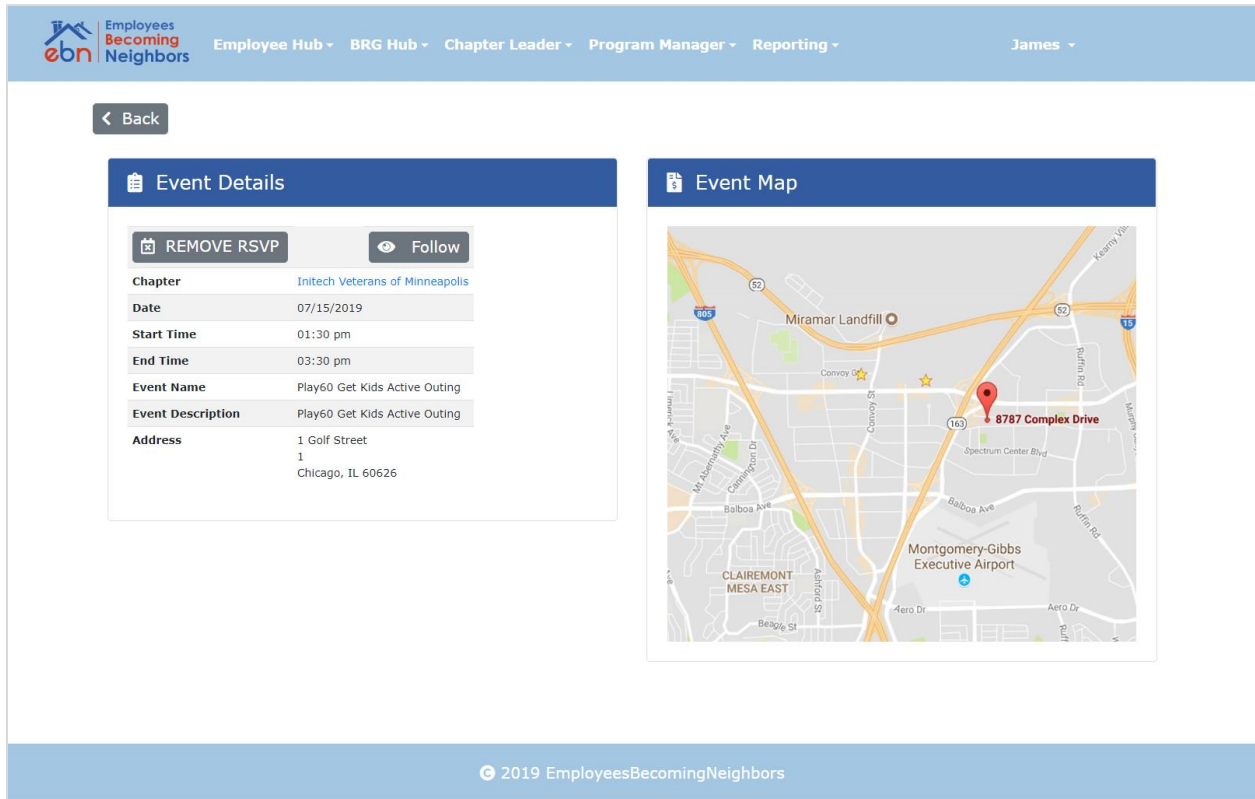
Figure 17. EBNportal.com UI- event listing below shows how we list out events to Program Managers.

The screenshot shows the 'Master Event List' interface. At the top, there is a navigation bar with the EBN logo and the text 'Employees Becoming Neighbors'. The navigation menu includes 'Employee Hub', 'BRG Hub', 'Chapter Leader', 'Program Manager', and 'Reporting'. The user's name 'James' is displayed on the right. Below the navigation bar is a blue header for the 'Master Event List' with a '+ Add New Event' button. The main content is a table with the following columns: Id, Event Date, Event Name, Street Address, City, State, and ZIP. Each row represents an event and includes a trash icon and a plus icon.

Id	Event Date	Event Name	Street Address	City	State	ZIP
1	06/07/2019	Animal Rescue Volunteer Session	60469 Schurz Place	Chicago	IL	60613
2	03/30/2019	Career Path Mentoring Workshop	40 Warbler Pass	Chicago	IL	60640
3	09/23/2019	Food Pantry Outing	7 Spaight Crossing	Chicago	IL	60607
4	07/15/2019	Play60 Get Kids Active Outing	1 Golf Street	Chicago	IL	60626
5	08/15/2019	Military Family Night Out	617 Howde Terrace	Chicago	IL	60647
6	05/30/2019	Homeless Clothing Drive	32 Cascade Street	Chicago	IL	60603
7	04/06/2019	Kids Learn Tech Workshop	415 Wayridge Drive	Chicago	IL	60604
8	06/25/2019	Student Grammar Tutoring Session	1063 Ilene Center	Chicago	IL	60604
9	08/19/2019	Kids Learn Tech Workshop	107 Pine View Junction	Chicago	IL	60618
10	09/21/2019	Blood Donation Drive	1 Ramsey Alley	Chicago	IL	60605
11	04/22/2019	Urban Landscaping Outing	3106 Doe Crossing Trail	Chicago	IL	60639
12	09/06/2019	Christmas Kids Gift Drive	3662 Farwell Drive	Chicago	IL	60602
13	09/05/2019	Job Interviewing Workshop for Veterans	35481 8th Place	Chicago	IL	60649

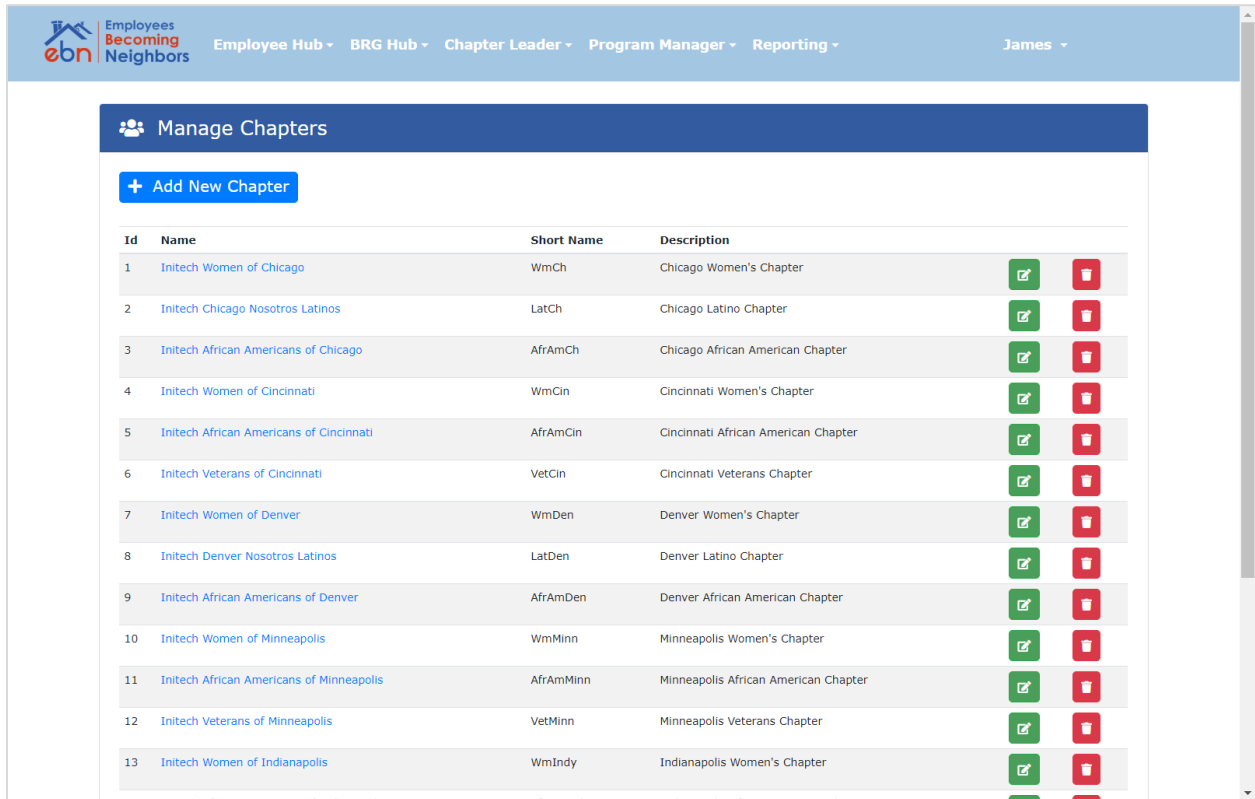
Figure 17. EBNportal.com UI- event listing

**Figure 18. EBNportal.com UI- event detail employee view** below shows how employees view details for an event they were interested in.



**Figure 18. EBNportal.com UI- event detail employee view**

**Figure 19. EBNportal.com UI- BRG chapter management form** below shows how a Program Manager will manage BRG chapters.



**Figure 19. EBNportal.com UI- BRG chapter management form**

**Figure 20. EBNportal.com UI- VTO entry form for employees** below shows a sample VTO entry page employees will use.

The screenshot displays the EBNportal.com interface. At the top, the logo 'Employees Becoming Neighbors' is visible alongside navigation links: Employee Hub, BRG Hub, Chapter Leader, Program Manager, and Reporting. The user's name 'James' is shown in the top right corner.

The main content area is divided into two sections:

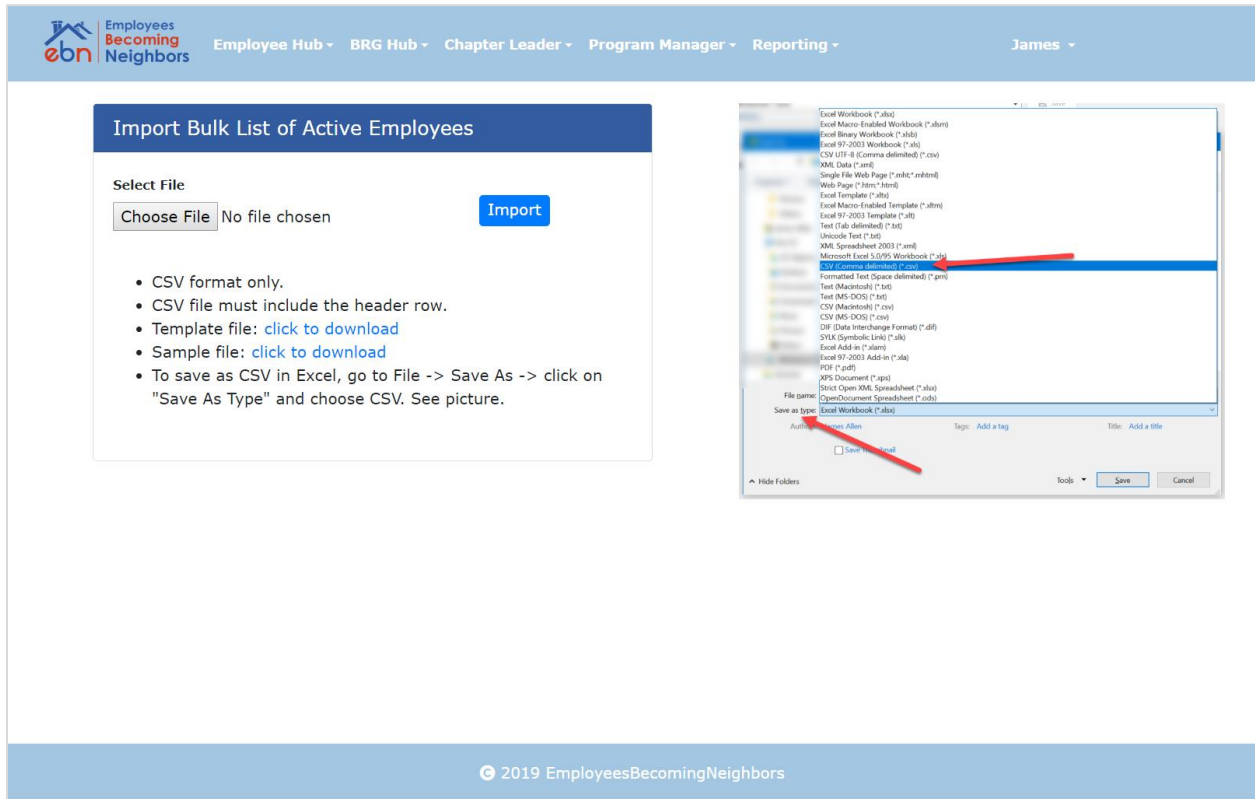
- My 2019 Volunteer Summary:** A table with the following data:
 

Date	Volunteer Organization	Supervisor Name	Hours	Status	Manager
04/01/2019	Winton Woods	John Doe	8	accepted	moorwood8
03/01/2019	Food Bank	Jane Doe	4	rejected	moorwood8
03/27/2019	Food Bank	Eugene Rogers	5	accepted	moorwood8
- Volunteer Time-Off Details:** A form with the following fields:
  - Hours \* (text input)
  - Date \* (text input with placeholder 'mm/dd/yyyy')
  - Volunteer Organization \* (text input)
  - Supervisor Name \* (text input)
  - Phone Number \* (text input)
  - Details \* (text area with 'max 150 characters' placeholder)
  - Submit (blue button)

The footer of the page contains the copyright notice: © 2019 EmployeesBecomingNeighbors.

**Figure 20. EBNportal.com UI- VTO entry form for employees**

**Figure 21. EBNportal.com UI- employee CSV import** below shows one of the CSV bulk upload pages for Program Managers.



**Figure 21. EBNportal.com UI- employee CSV import**

## Section 5. Discussion

### 5.1 Problems Encountered

For the most part our project has gone well. However, we did run into two issues that we had to deal with.

First, since we as students are not working at a real company with real infrastructure, we had issues getting our individual local development environments to mirror each other enough so that when we sync code or database changes, we wouldn't run into any issues.

Second, while we are making good progress, our team had a wide array of skillsets and level of experience. There was a misunderstanding of the various levels of experience each person brought to the team. So, we had to make some adjustments in the form of shifting workloads to make it more equal.

## 5.2 Future Recommendations

### **What would do we if we had it all over to do again?**

If we had to do this all over again, we would have chosen a different technology stack and brought on another programmer.

PHP and MySQL, while easy to learn and easy to write code in, is hard to manage when considering scaling and maintaining a large application. If we wanted to stay with PHP, we would explore a PHP framework like Laravel. Another option would be to use a different programming language and framework like Microsoft's C# and their ASP.NET Core Framework. Either option would help developers write scalable applications while still maintaining quality.

We wish we had pursued getting a fourth team member with strong development skills to allow us to get more development done.

### **What would we do if we had more time?**

If we had more time, we would improve functionality around searching, email notifications, management of the EBNPortal.com customers (i.e., the Fortune 500 companies who sign up), and improve overall system performance.

### **What suggestions did we get from others?**

Feedback we got from others mainly regarded suggesting that we get feedback from market experts to get more validation if there is a market for this product. A lot of people suggested we should continue with this.

We also had a suggestion from someone at IT Expo. He suggested we not limit this product to just companies. He suggested we consider schools too because teachers, staff, and students all volunteer as well and they need to track their hours. So, we could open this up to non-profit organizations as well.

### **What are our plans for this project?**

At this time, Micah and James are looking into finding someone with business expertise, an “MBA-type”, who can help us with our missing business competencies. For example, business planning, startup financing, angel investors, financial forecasting, marketing, and sales.

Zach indicated he is no longer interested in this project as he is pursuing a master’s degree and career in the field of nursing. He agreed to give his share of the ownership over to Micah and James.

## Section 6. Conclusion

### 6.1 Fall Semester 2018 Lessons Learned

*This section is from November 2018.*

We feel we learned two main lessons this semester.

First, it is difficult to make sure a team stays focused, especially when the team works in different locations, and not in a shared workspace like an office. Teams can succeed in this situation, and we feel we are succeeding. However, teams must be diligent and intentional about making sure the team works well together and towards a shared goal.

Second, making a product of this magnitude is not a small task. A team must take so many things into consideration when designing, building, testing, and releasing a product. We must think about security, scalability, the user's experience, finding a hosting company, and many more things.

It gets even more complex when you consider that if we were a real business, we would have to worry about finance, accounting, taxes, business plans, having enough of a runway of funding, hiring employees, and so on.

## 6.2 Spring Semester 2019 Goals

*This section is from November 2018.*

Once the fall semester is over, we plan to continue to meet weekly except for the very end of Christmas. We plan on taking 2 weeks off in late December to give us a chance to keep a work-life balance and take a “mental health” break. That break might help us step away and process everything we worked through and maybe come up with new ideas.

As for our project goals, we plan on continuing with building out the product and testing. We hope to finish out the employee-based features (non-BRG event related features) and reporting. We also plan on improving the code base and database regarding the fact that many companies (i.e., customers) will use this application. We also plan on continuing to fortify the application around the area of security and OWASP Top Ten recommendations. Finally, we still need to figure out how we will run automated jobs that will notify employees via e-mail of an event he/she RSVP'd for.

## 6.3 Spring Semester 2019 Lessons Learned

*This section is from April 2019.*

We struggled with the imbalance of work load, where some team members were having to deal with most of the work load of programming. We learned that the process of forming a team should have been better handled. We would have asked tougher questions while interviewing team members to get a better sense of the bench strength of the team.

When building an app, it's just not feasible to think one person can go two entire semesters without writing code, only contribute to other tasks, and have that be equal to the coding effort.

We also learned how much work goes into making the project come to completion, with all the paperwork and technical work. We underestimated how much was involved.

## 6.4 Final Thoughts

*This section is from April 2019.*

We made it. We learned a lot through our struggles, and it feels great to cross the finish line. We hope the faculty continues to improve the Senior Design process.

We were impressed with the IT Expo event. We got a lot of great feedback. While we had an issue with the judging panel for our category, we think it is a great event for Seniors and others to show off their work and network with others.

We hope to see the CECH School of Information Technology continue to grow and become a powerhouse in the academic and business worlds.

## References

1. DeLoitte. "How Creating a Volunteer Culture Can Help Your Business" [prnewswire.com](https://www.prnewswire.com/news-releases/how-creating-a-volunteer-culture-can-help-your-business-300470679.html).  
<https://www.prnewswire.com/news-releases/how-creating-a-volunteer-culture-can-help-your-business-300470679.html> (accessed October 10, 2018).
2. Shelton Goode, Ph.D., Isaac Dixon, Ph.D., SHRM-SCP. "Are Employee Resource Groups Good for Business?" [bibme.org](https://www.shrm.org/hr-today/news/hr-magazine/0916/pages/are-employee-resource-groups-good-for-business.aspx). <https://www.shrm.org/hr-today/news/hr-magazine/0916/pages/are-employee-resource-groups-good-for-business.aspx> (accessed October 10, 2018).

# Appendix

Figure 22. Team Poster- below shows our team’s booth poster for the 2019 IT Expo 2019.

**ebn** | Employees Becoming Neighbors

University of CINCINNATI  
College of Education, Criminal Justice, & Human Services | School of Information Technology

**Problem**  
Many Fortune 500 companies don't have a single solution or single source of truth that tracks where the company's money or employee volunteer time is going.

**Solution**  
Employees Becoming Neighbors is a "one-stop shop" for companies to track their employee paid volunteer time off (VTO), and give their Business Resource Groups (BRG) a tool to manage their community events. Our product also gives company Executives one place to run reports to see how their company is meeting their goals.

**What is Employees Becoming Neighbors?**  
Employees Becoming Neighbors is a cloud-based SaaS solution for Fortune 500 companies that helps them track their social responsibility objectives.

**Technology**  
php, MySQL, HTML5, JS

Team 13: James Allen, Micah Johnson, & Zach Goodwin  
Faculty Advisor: Professor Abdou Fall

Figure 22. Team Poster

**Table 3. Operating Budget Breakdown** shows the monthly breakdown of the operating budget.

			Month 1	Month 2	Month 3	Month 4	Month 5	Month 6
	MSDN							
<b>Human Resources Costs</b>	<b>License?</b>	<b>Cost Per Hour</b>						
<b>FTEs</b>								
Business Analyst	x	\$65	\$ 11,258	\$ 11,258	\$ 11,258	\$ 11,258	\$ 11,258	\$ 11,258
Senior Application Developer	x	\$80	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856
Database Admin/Developer	x	\$80	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856
Manager / System Administrator	x	\$95	\$ 16,454	\$ 16,454	\$ 16,454	\$ 16,454	\$ 16,454	\$ 16,454
Business Development / Account Manager		\$80	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856
<b>Contractors</b>								
Developer	x	\$100	\$ 17,320	\$ 17,320	\$ 17,320	\$ 17,320	\$ -	\$ -
Developer	x	\$100	\$ 17,320	\$ 17,320	\$ 17,320	\$ 17,320	\$ -	\$ -
<b>Technical Costs</b>								
Azure PaaS Cloud Hosting		See note						
Azure Web Application Hosting			\$ 554	\$ 554	\$ 554	\$ 554	\$ 554	\$ 554
Azure Database Hosting			\$ 2,430	\$ 2,430	\$ 2,430	\$ 2,430	\$ 2,430	\$ 2,430
Hardware		See note						
Workstations count			7	7	7	7	7	7
Workstations cost			\$ 778	\$ 778	\$ 778	\$ 778	\$ 778	\$ 778
Development Tools		See note						
MSDN Licenses			6	6	6	6	4	4
MSDN Licensing			\$ 1,500	\$ 1,500	\$ 1,500	\$ 1,500	\$ 1,000	\$ 1,000
<b>Other Costs</b>								
Professional Services		See note	\$ 20,000	\$ 20,000	\$ 20,000	\$ 20,000	\$ 20,000	\$ 20,000
Training		See note						
Pluralsight seats needed			6	6	6	6	6	6
Pluralsight subscription			\$ 375	\$ 375	\$ 375	\$ 375	\$ 375	\$ 375
			\$ 129,557	\$ 129,557	\$ 129,557	\$ 129,557	\$ 94,417	\$ 94,417

Human Resources Costs	<b>76%</b>	73%
Technical Costs	<b>5%</b>	5%
Other Costs	<b>19%</b>	22%

**Assumptions**

- \* We hire contractors for first 4 months during initial development.
- \* FTE hourly cost assumes loaded cost to cover taxes, benefits, etc.. FTE will not gross that hourly rate, instead approx. 60% of that rate.
- \* Contractor hourly cost assumes the contracting company rate, and that company will handle their own benefits, taxes, profit, etc..

**Notes**

- \* Azure costs based off of Azure calculator, 730 hours per month <https://azure.microsoft.com/en-us/pricing/calculator/>  
 Web App Hosting = Azure App Service S3, 4 cores, 7GB RAM, 2 instances, 1 SSL connection, West US, Linux-based, Standard Tier  
 Database Hosting = Azure Database for MySQL, Gen 4 16v core, 2 servers, 100GB storage, 100 GB Geo-Redundant storage
- \* Hardware = 1 business-grade laptop per FTE and Contractor, service contact, \$4,000 each, capitalizable across the 3 years
- \* Development Tools = MSDN Enterprise license, \$250 per developer per month <https://visualstudio.microsoft.com/vs/pricing/>
- \* Professional Services = Hiring external providers for Office 365 cloud (email), design, marketing, accounting, legal, HR, etc.
- \* Training = Pluralsight license, \$750/ user per year, \$62.50/month <https://www.pluralsight.com/pricing>

**Table 3. Operating Budget Breakdown**

Month 7	Month 8	Month 9	Month 10	Month 11	Month 12	Annual TOTAL	Annual no contractors TOTAL
\$ 11,258	\$ 11,258	\$ 11,258	\$ 11,258	\$ 11,258	\$ 11,258	\$ 135,096	\$ 135,096
\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 166,272	\$ 166,272
\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 166,272	\$ 166,272
\$ 16,454	\$ 16,454	\$ 16,454	\$ 16,454	\$ 16,454	\$ 16,454	\$ 197,448	\$ 197,448
\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 13,856	\$ 166,272	\$ 166,272
\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 69,280	\$ -
\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 69,280	\$ -
\$ 554	\$ 554	\$ 554	\$ 554	\$ 554	\$ 554	\$ 6,648	\$ 6,648
\$ 2,430	\$ 2,430	\$ 2,430	\$ 2,430	\$ 2,430	\$ 2,430	\$ 29,160	\$ 29,160
7	7	7	7	7	7		
\$ 778	\$ 778	\$ 778	\$ 778	\$ 778	\$ 778	\$ 9,333	\$ 9,333
4	4	4	4	4	4		
\$ 1,000	\$ 1,000	\$ 1,000	\$ 1,000	\$ 1,000	\$ 1,000	\$ 14,000	\$ 14,000
\$ 20,000	\$ 20,000	\$ 20,000	\$ 20,000	\$ 20,000	\$ 20,000	\$ 240,000	\$ 240,000
6	6	6	6	6	6		
\$ 375	\$ 375	\$ 375	\$ 375	\$ 375	\$ 375	\$ 4,500	\$ 4,500
\$ 94,417	\$ 94,417	\$ 94,417	\$ 94,417	\$ 94,417	\$ 94,417	\$ 1,273,561	\$ 1,135,001

**Table 3. Operating Budget Breakdown (continued)**