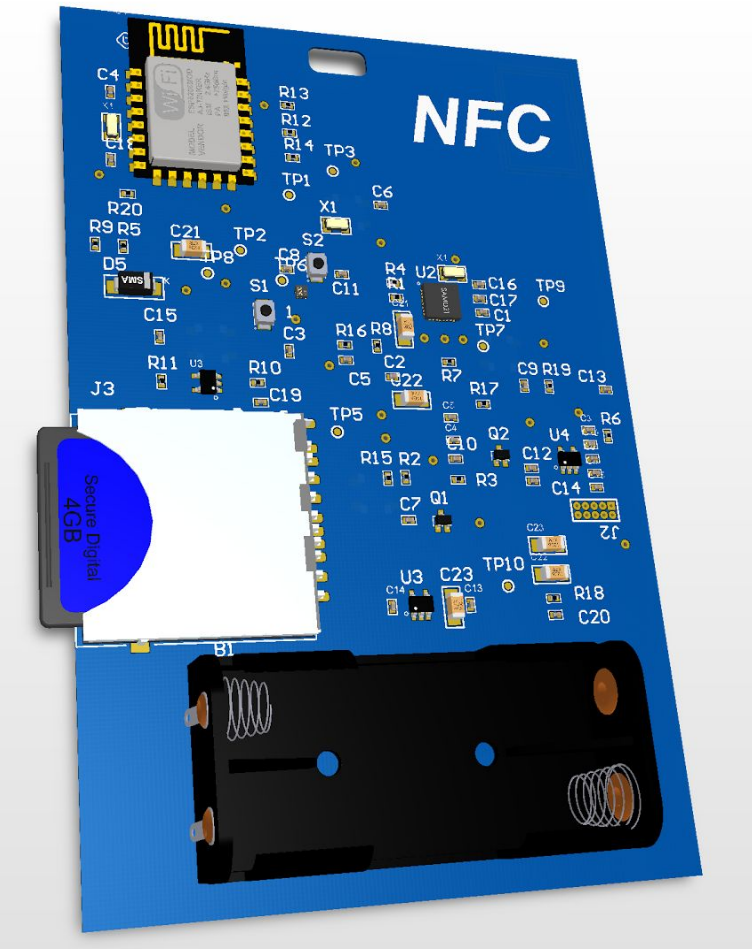
TouchBadge

An Internet Connected Lanyard for Conferences



Computer Science and Engineering Department, University of Nevada, Reno

Team 11: Evan Grill, Jon Weatherspoon, Cody Worsnop

Instructors: Sergiu Dascalu, Devrin Lee

External Advisors: Daniel deLaveaga and Elizabeth Hitchcock of Breadware, Inc.

November 15, 2017

Table Of Contents

Abstract

Conferences exist to cater to almost every industry, and some companies hold conferences entirely dedicated to their own customers. Theses events are prime locations for networking with other people in the industry as well as potential customers. For vendors at these events, conference badges are an important tool as they allow for the collection of attendee data for future contact. TouchBadge looks to bring that kind of experience to every conference attendee.

Working with hardware partner Breadware, Inc., we are developing TouchBadge, a smart conference badge. TouchBadge allows the transfer of contact information between attendees wirelessly and without an external reader. Once obtained, contacts are stored in the cloud for later retrieval and are exportable to the user's customer relationship management (CRM) software of their choice.

This document serves as a reference point to the development of the product. The prototype design was formulated and developed to provide a general understanding of what the product is and how it is unique.

Introduction

TouchBadge is a smart conference badge. Designed with hardware partner Breadware, Inc., it consists of an electronic badge equipped with near-field communication (NFC) and Bluetooth protocols to allow for interaction between TouchBadges as well as smartphones for syncing to the TouchBadge web service. The goal of TouchBadge is to reduce the time and effort it takes to share information in order to enhance the networking experience of conference attendees.

Since the design report, the design infrastructure has changed. The development of the webservice has shifted to use Azure as the primary form of communication to the backend. The Azure SDK has extensive tools that allow for a greater flexibility when implementing the project. The application shall still be interacted with through the web-based attendee application portal (AAP) and conference organizer application portal (COAP), or the companion mobile application, being developed for Android and iOS.

The web service will stand as a common point of communication for all of the various access interfaces available to users. Additionally, it will handle the database management keeping the interactions in and out as clean as possible. Security will be handled by requiring the use of Secure Sockets Layer (SSL) for the communications. Authentication is for the application is handled by an Azure B2C tenant in cooperation with the Microsoft Authentication Library (MSAL). MSAL uses token authentication and will be able to seamlessly re-authenticate the user in case of timeout.

The mobile application will be the primary point of interaction for the attendees. It will provide the ability for the user to activate or reactivate (in the case of failure) a TouchBadge device. Additionally, it will serve as the initiator for syncing the TouchBadge to the user's mobile device.

The AAP will allow the user easy access to all of their obtained contacts, separated by event, and provide the ability to export data to other platforms (e.g. CRMs, Outlook, etc.). The COAP will allow the conference organizer to create and manage conferences as well as manage the list of registered attendees.

Prototype Objectives

The prototype objectives consist of three subgoals that provide a broad overview of the end product objectives. The subgoals have been outlined below.

Plan

The overall plan of the prototype is to provide a interactive demonstration of the individual pieces of the project. To accomplish this, the team set out to develop pieces to a much more complete state than a traditional prototype, at the expense of having working implementations of all parts. The prototype includes several elements, as outlined below, in addition to UI testing, and unit testing. This provides a complete view from the consumer as well as developer side.

Mobile Application

As the mobile application is the main point of contact between the TouchBadge and the consumer, it shall have the greatest amount of development for the prototype. It shall provide a general structure of the application focusing on business-tier logic over front-end look. The prototype objectives are the following:

- The ability to log users in and out
- Add and remove contacts
- Push new data to the backend
- Demonstrate successful building on the build server in conjunction with unit testing

Website

As most of the development time will be spent on the mobile application, the website shall serve to simply show the backend data for the prototype. This shall be a proof of concept to demo the product to the instructors and external advisors. The objectives for the website are the following:

- The ability for the website to be displayed gracefully on multiple platforms
- Navigation to a contact page
- The ability to see all added contacts in the database

Backend

The backend shall play an integral role within the prototype. It shall be able to receive data from the mobile application and store it within a SQL database. It shall also be able to send data to the mobile application as well as the website when requested. Due to this, the backend objectives are the following:

- Provide a platform for which the mobile and web applications may authenticate against.
- Support the creation of multiple tables within the SQL database
- Allow for the sending and receiving of various data

Prototype Functionality

Mobile Application

The mobile application shall contain a broad overview of the total functionality that shall be available in the final product. The functionality available in the prototype shall be the following:

Signing In and Out

Logging in will use the Microsoft Authentication Library (MSAL) to authenticate users against a Business to Consumer (B2C) Azure Tenant. Users must log in to use the application as it provides a baseline of information of the user. For the prototype, users will be able to create their own account or authenticate against LinkedIn. Further development will bring the ability to log in via Google, Facebook, or other social media providers. The prototype shall also have the ability to reset their password if lost or forgotten, and should support two-factor authentication for greater security.

Adding and Removing Contacts

Contacts shall also be present in the prototype. When adding a contact, the user shall be prompted for full name of the contact, phone number, email address, and picture if wanted. Once added, the contact shall be automatically saved locally as well as pushed to the backend assuming a connection is available. While viewing contacts, users shall be able to remove users which deletes the contact locally and in the backend. Moreover, users shall also be able to tap on the contact and view additional info that is not able to be viewed at a glance.

Adding and Removing Events

Events shall also be added for the purposes of showing the event structure. During production, users will have the ability to add personal events that are stored locally and in the backend. Events that they are registered to shall automatically be pushed to their calendar.

Notes on Hardware

As the hardware itself is still being designed and produced, the prototype shall not have any configuration elements available. Our partner Breadware is designing and making the hardware and has agreed to write the firmware for it as well. The app shall then allow advanced configuration once Breadware is finished.

Web Application

For the prototype demonstration, the web application will serve as a visual demonstration of our backend infrastructure. The prototype of the web application will provide the following functionality:

Signing In and Out

When the user first navigates to the web application, they will be prompted to authenticate against the same B2C tenant as the mobile application. The web application prototype, because it serves mostly as a visual demonstration of the backend infrastructure, will only support local account authentication. Further iterations of development will include the ability for users to authenticate using the same identity providers as the mobile application, such as LinkedIn, Google, and Facebook.

Viewing All Contacts

Once the user has logged in, they will be redirected to the main page, which will display to them the amount of contacts that are currently stored in the backend database. The main page will also provide a method for navigating to the prototype of the contacts page, where a user will be able to view their contacts. For the prototype, all users will be able to view all contacts in the database, but in further versions, both the web application and the mobile application will ensure that a user can only view the contacts they've been linked with.

Viewing Detailed Contact Information

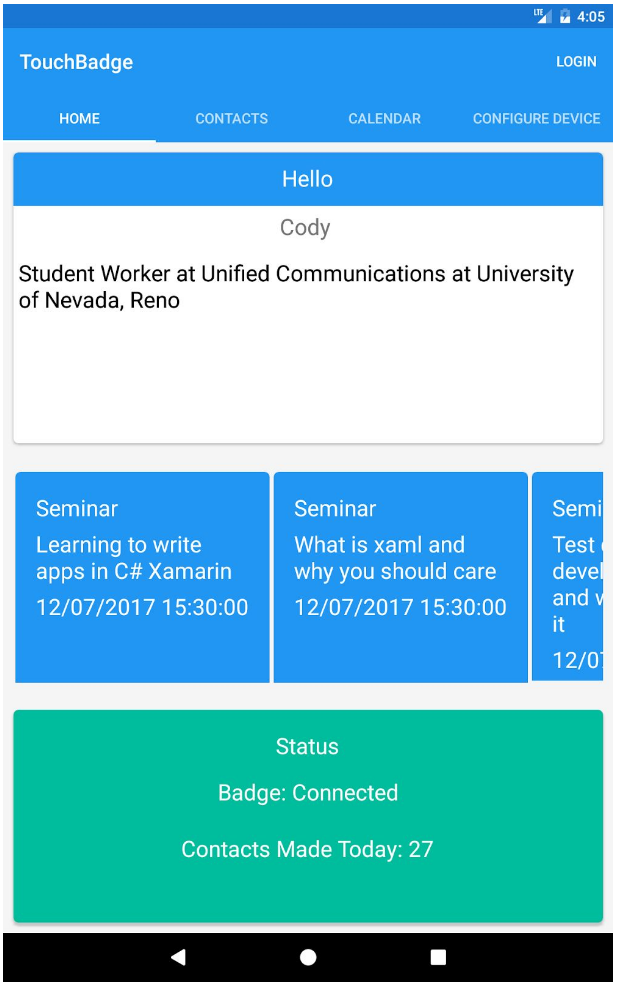
The contact page will display a list of all contacts as well as some basic information about them, such as their name, job title, and photo. Should the user want to view more detailed contact information, they simply need to click on one of the contacts in the list. Doing so will bring up a bootstrap modal with all the contact information for that person.

Backend

For the prototype demonstration, the mobile and web applications interact directly with the database using existing tools built into Microsoft Azure. This allows for unified user login across the platform. The team intends to replace the direct database interaction with a backend web service that will handle the interactions instead.

Developed Prototype

Mobile Application Prototype UI



Hello

Cody

Student Worker at Unified Communications at University of Nevada, Reno

Seminar

Learning to write apps in C# Xamarin

12/07/2017 15:30:00

Seminar

What is xaml and why you should care

12/07/2017 15:30:00

Semi

Test
devel
and v
it

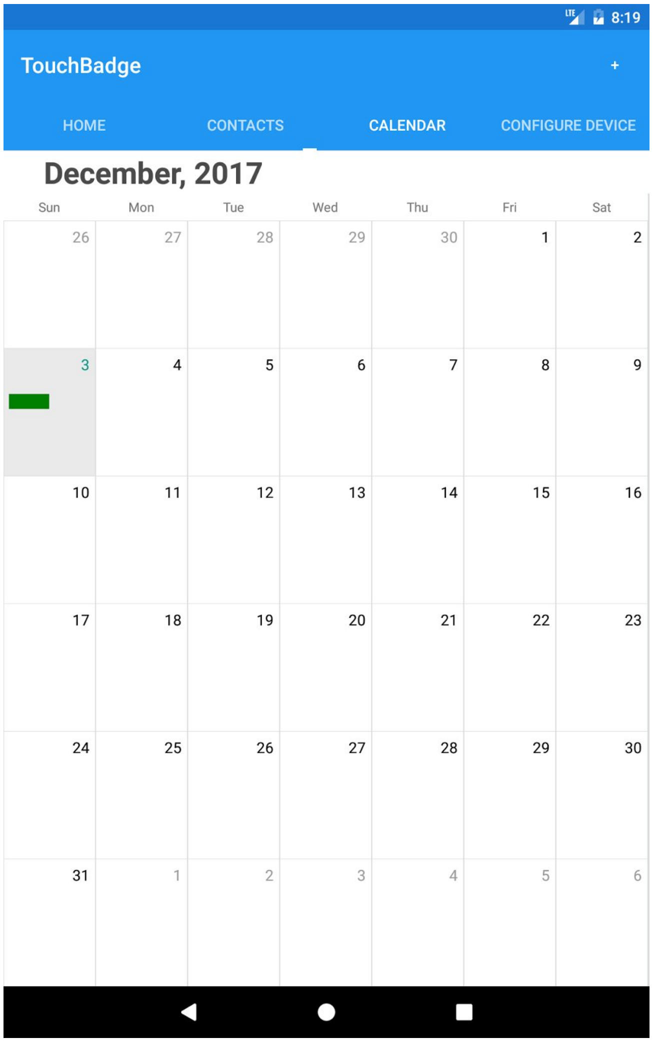
12/0

Status

Badge: Connected

Contacts Made Today: 27

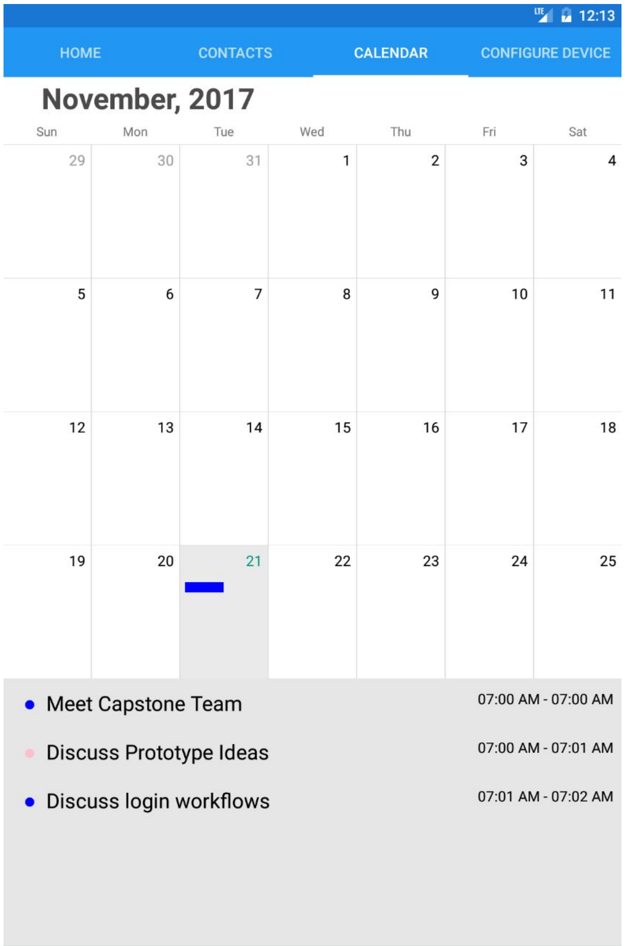
Figure 1: The TouchBadge view. The home screen is designed to give the user a complete status of the app including how many contacts they have made that day, the status of the badge hardware, how many seminars the user has left in the day and their personal information.



December, 2017

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	1	2
3 🟩	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Figure 2: The calendar view. The calendar gives the user an overview of all the events they have planned throughout the day and month. As seen here, there is an event on the 3rd. The user is able to then touch the day and the calendar will drop down and display details about the event.

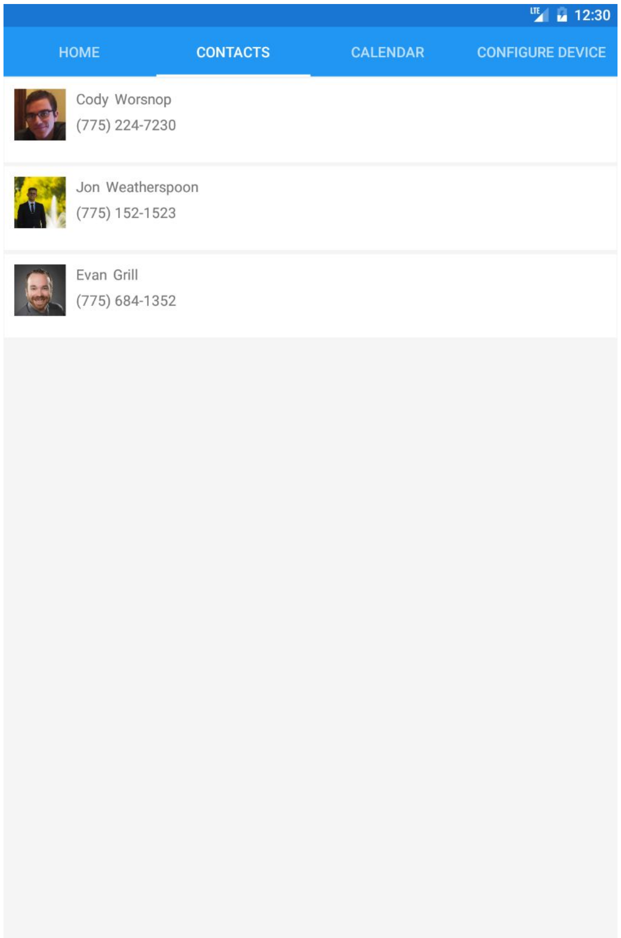


November, 2017

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25

● Meet Capstone Team	07:00 AM - 07:00 AM
● Discuss Prototype Ideas	07:00 AM - 07:01 AM
● Discuss login workflows	07:01 AM - 07:02 AM

Figure 3: The detailed calendar view. This view is seen when the user selects a certain day to look at. The events that day will be visible to the user in detail.



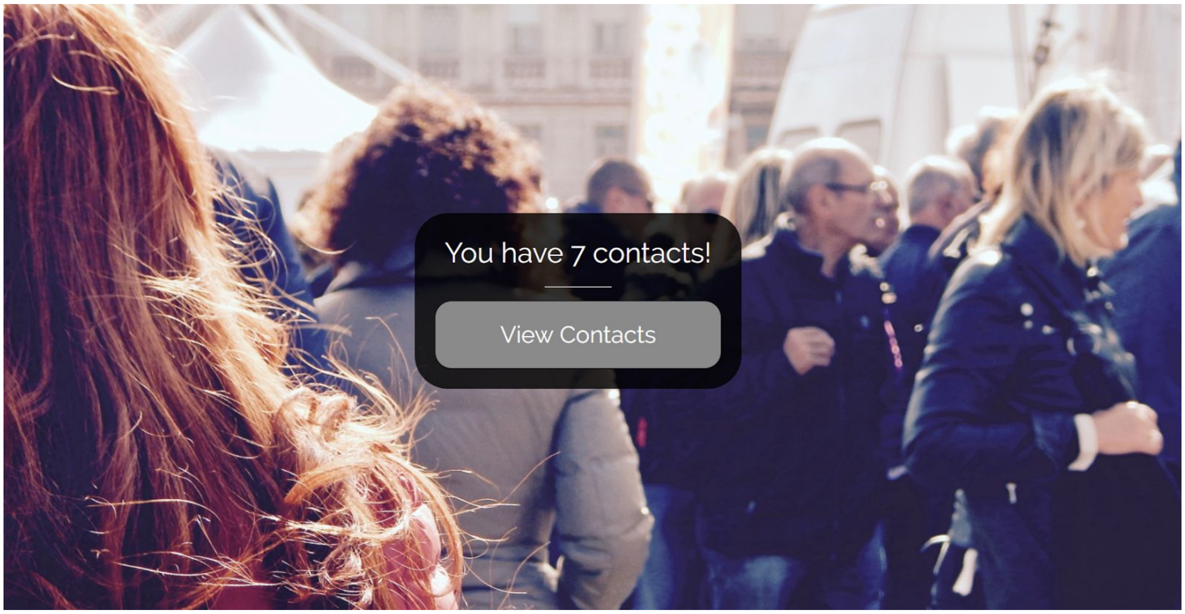
Cody Worsnop
(775) 224-7230

Jon Weatherspoon
(775) 152-1523

Evan Grill
(775) 684-1352

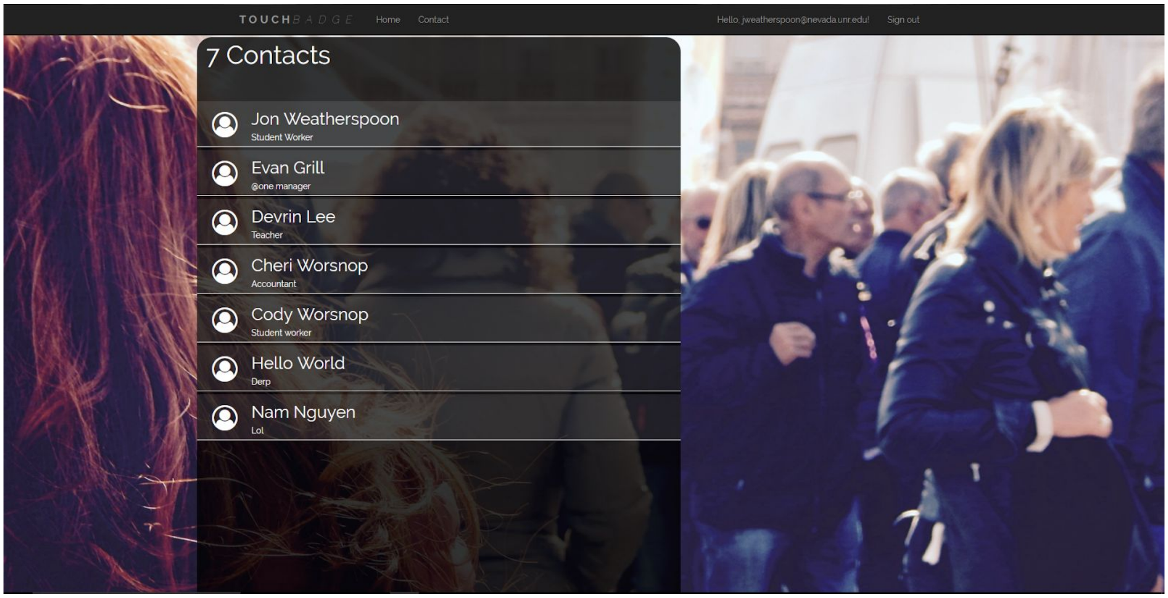
Figure 4: The contacts view. This view displays all the contacts the user has made over the course of using the mobile application. Upon tapping a contact a detailed view modal will display showing more information about the user.

Web Application Prototype UI



You have 7 contacts!

Figure 5: The splash page for the TouchBadge web application provides a quick overview of the number of contacts available for viewing in the database. It also provides a means for navigating to the contact page, where a user will be able to view information about the contacts in the database.



Hello, jweatherspoon@nevada.unr.edu!

7 Contacts

Jon Weatherspoon
Student Worker

Evan Grill
@one manager

Devrin Lee
Teacher

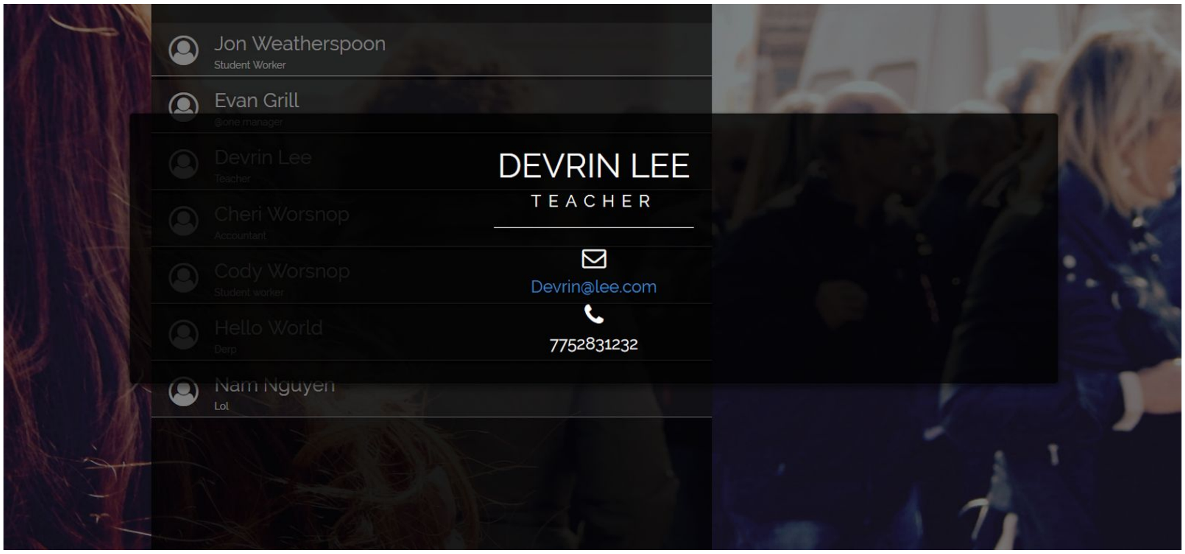
Cheri Worsnop
Accountant

Cody Worsnop
Student worker

Hello World
Derp

Nam Nguyen
Lol

Figure 6: The contact page incorporates a navigation bar at the top of the screen which displays information about the currently authenticated user and allows for navigation through the pages offered by the application. The main pane of the application shows the number of contacts a user can view followed by a list of the stored contacts.



Jon Weatherspoon
Student Worker

Evan Grill
@one manager

Devrin Lee
Teacher

Cheri Worsnop
Accountant

Cody Worsnop
Student worker

Hello World
Derp

Nam Nguyen
Lol

DEVRIN LEE
TEACHER

✉
Devrin@lee.com

☎
7752831232

Figure 7: The detail view of a contact appears as a bootstrap modal, so the user will not have to navigate away from the current page to view information about a contact. This will help increase the usability of the application while providing a modern, sleek appearance to the detail view.

Prototype Evaluation

Demo

We met with our sponsor at Breadware, Daniel deLaveaga on December 15th, 2017 at 3:00PM to demonstrate the current prototype for the TouchBadge system. The demo itself consisted of a broad overview of the application. We presented the Xamarin mobile application but also indicated and presented an Ionic application with AWS. We included creating a pseudo contact to show the functionality of the backend and also showed the capability to sync with the website.

Feedback

Overall, our sponsor was happy with the prototype demonstration. He liked the structure of the UI and was satisfied with the progress we have made. The majority of conversation for improvement revolved around the TouchBadge hardware itself. Particularly, how is the user going to initially interact with it, and how is new information going to be synced across devices. For example, should the mobile application specify what information is currently loaded in the badge for the user to see?

From this point, Daniel laid out a roadmap for us to continue working on the UI. He will begin research on prototyping hardware we will be able to use for future development of the TouchBadge system.

Prototype Demonstration

Team 11 demonstrated their prototype to Devrin Lee and Connor Scully-Allison on December 7, 2017 at 8:00AM PST. Both Devrin and Connor offered some helpful points team 11 will consider when furthering the development of the TouchBadge system, which follows:

- Microsoft Azure vs. Amazon AWS
    - Team 11 may want to consider using Amazon AWS as its cloud hosting service, as it is free for students and may offer a more user friendly hosting service.
- Xamarin vs. Ionic
    - Team 11 may wish to consider switching the mobile development platform from Xamarin forms to Ionic. Xamarin seems to be the most optimized as it compiles to native code for both Android and iOS. Ionic utilizes HTML and CSS to create the user interface and as a result, presents a unified interface across both platforms as opposed to the native controls used by Xamarin.
- Design Critiques
    - Devrin mentioned some design elements that should be replaced or altered in future updates. These included adding a larger margin in UI boxes, and separating some boxes that contained two different types of information.

Design Updates

While developing the TouchBadge prototype, Team 11 ran into a few issues with Microsoft Azure and noticed a few shortcomings of the prototype functionality.

Web Application

The prototype for the web application does introduce some functionality, and certainly provides proof that Team 11 will be able to use the ASP.NET framework to develop an elegant and robust web application, but currently fails to address all the required aspects of the project.

The prototype's authentication system fails to include the same sign-in and sign-up flow that the mobile application does, and Team 11 wishes to ensure that both platforms follow the same authentication flow to ensure ease-of-use when switching between platforms. The issues with authentication will likely be simple fixes, as the issue exists because the web application does not currently use the same authentication policies as the mobile application.

The web application prototype also does not include any pages other than a placeholder index file as well as the contact page. For the final product, it must include a user dashboard, an events page, a calendar, and perhaps more should the design specification change during development.

Microsoft Azure

Microsoft Azure is a very powerful platform for creating applications, and the team felt that it was a logical choice when building the web and mobile applications on Microsoft Frameworks. However, with that power came a very steep learning curve. The team was able to understand how to create individual elements of the application stack but struggled when connecting them together.

As a result, the team is currently researching Amazon AWS and their cloud services, and may choose to migrate the backend infrastructure from Microsoft Azure to Amazon AWS if they discover AWS is more suitable for the TouchBadge system.

The team also plans to customize the UI for all pages regarding authentication. Currently, the TouchBadge prototype uses the default Microsoft login flow, but the team will customize the authentication flow in further development.

Team Contributions

Evan Grill

Over the course of the prototype development, Evan worked on the following items:

- Abstract - **0.5 hours**
- Introduction - **1 hours**
- Editing and Revision - **3 hours**
- Backend configuration - **3 hours**

Jon Weatherspoon

Over the course of the prototype development, Jon worked on the following items:

- Prototype Objectives - **0.25 hours**
- Prototype Functionality - **0.25 hours**
- Developed Prototype - **40 hours**
- Prototype Demonstration - **0.25 hours**

Cody Worsnop

Over the course of the prototype development, Cody worked on the following items:

- Coverpage **- 0.5 hours**
- Table of Contents **- 0.5 hours**
- Prototype Objectives **- 1 hours**
- Prototype Functionality **- 1 hours**
- Developed Prototype **- 40 hours**