Photo Archive for the Bureau Of Engineering  
(PABOE)  
CS4961 & CS4962 Senior Design  
Project Document  
Prepared by:  

Team Members:  
Stephanie Daley, Nikko De Guzman, Phillip Gulegin, Alejandra Monton Sanches,  
Edwin Ochoa, Felipe Vega  

Faculty Advisors:  
Russell Abbott, Jungsoo (Sue) Lim, Laura Mann  
May 5, 2017  
CALIFORNIA STATE UNIVERSITY LOS ANGELES  
Los Angeles, California  

Photo Archive BOE  
(PABOE)
# Table of Contents

Section 1: Executive Summary 3

Section 2: Introduction 5

Section 3: User Guide 5

3.1 Requirements 5

3.2 User Instructions 6

3.2.1 iOS 6

3.2.1.1 Logging On 6

3.2.1.2 System Menu 7

3.2.1.2.1 Dashboard Tab 7

3.2.1.2.2 Tagging Tab 8

3.2.1.2.3 Camera Tab 9

3.2.1.2.4 History Tab 10

3.2.1.2.5 Settings Tab 11

3.2.1.3 Exit System 12

3.2.2 Android 13

3.2.2.1 Logging In 13

3.2.2.2 System Menu 14

3.2.2.2.1 Dashboard Tab 14

3.2.2.2.2 Tagging Tab 15

3.2.2.2.3 Camera Tab 19

3.2.2.2.4 History Tab 21

3.2.2.2.5 Settings Tab 22

3.2.2.3 Exit System 23

Section 4: Architecture and Design 24

4.1 SQL Server Database 24

4.2 Azure Storage Container 25

4.3 Distribution 25

Section 5: Conclusion 26

Section A: ACRONYMS 26
Section 1: Executive Summary

Welcome to CSULA’s 2016-2017 school year Senior Design project for the City of Los Angeles, Bureau of Public Works; from here on wards will be known as B.O.E (Bureau of Engineering). We are team P.A.B.O.E (Photo Archive for the Bureau of Engineering). As the acronym of our project states the purpose of our project was to create a photo archive for the bureau of engineering - using Azure SQL Server Database concurrently with Android and iOS mobile devices. In order to do this our team embarked upon a year long learning experience that none of us had ever had in our school careers. This summary will go over the requirements, how they were handled and the struggles involved, and the final product.

Firstly, the requirements set by the B.O.E. was the following: 1. Azure SQL Database services must be used, 2. Functioning mobile applications on both Android and iOS platforms, 3. Mobile applications must connect to the Azure SQL Database to both push and pull tagged photos, 4. Tagging is when a set of contexts and attributes are associated with a photo - i.e. Context: Project Number Attribute: 123456 - 5. The interface for both platforms must be easy to use and understand by B.O.E. employees - i.e. engineers working in the field.

In order to accommodate these requirements our team of six broke into three teams of two. An Android development team, an iOS development team, and a database/mobile connection to Azure team. The mobile application teams worked on both the front end design and the encompassing mobile capabilities; i.e. tagging; respectively using Java with Android Studio and Swift to program the applications. While the Azure team mainly focused on creating the SQL database in Azure and implementing the mobile to Azure connection on both mobile platforms.

There were many struggles from all teams in fulfilling the requirement expectations. Some blocks that came up for both the iOS and Android team was the fact that neither member had ever programmed in Swift, the iOS programming language, and Xcode, the compiler, as well as Android Studio respectively before. Both teams had many debugging issues to overcome in the course of this year. The Azure team, also, had to learn Azure from scratch and implement many provided documentation on the Microsoft website - that in hindsight was very poorly done on Microsofts part. There were many bottle neck issues that came up with Azure from basic connection to pushing/pulling from the database to sign on key strings for
mobile application use.

In the end; however, all three teams buckled down and worked extremely hard to provide as quoted from our liaison Ray Uyemura from B.O.E., “[a] clean interface for our engineers to use” for both of our mobile platforms. The iOS and Android team have completed a fully function front end application for each respective platform - allowing a user to take a photo from the app and tag it from a provided list of tags; i.e. permit or project number; and move it to an upload cue where a User can also perform CRUD; Create, Read, Update, Delete; functions to any photo and tags associated with it. The Azure team has gotten a functioning backend connection the the Azure SQL Server on the Android platform. A User can push tagged photos from the upload cue and pull tagged photos to the app from the Azure Database to their Android device. Unfortunately. there was not enough time to do the same for iOS.

In conclusion, we completed all of the requirements accept getting the backend of iOS working. Nevertheless, when we spoke to our liaison, Ray Uyemura, about it he was happy at the work we accomplished saying, “you guys got more done in a year than many of our own employees would have with such a project”. Therefore, we are happy with what we accomplished and we are proud at how much we learned from struggling with new material this school year.
Section 2: Introduction

P.A.B.O.E is a cross platform mobile application that will allow employees, from here on will be referred to as User(s), of the Bureau of Engineering to take relevant photos while working on any project, tag said photo with relevant information, and store it in the cloud. The User(s) will also be able to tag and upload photos not taken within the app to the cloud; as well as, recall all photos stored in the cloud through the app.

Section 3: User Guide

This section provides a general walkthrough of the system from initiation through exit. The logical arrangement of the information shall enable the functional personnel to understand the sequence and flow of the system. Use screen prints to depict examples of text under each heading.

3.1 Requirements

<table>
<thead>
<tr>
<th>Activity</th>
<th>New Capability</th>
<th>Feature Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Deployment</td>
<td>1a - Any Bureau of Engineering employee can use the app to take a relevant photo for a specific project and upload to the cloud database over wifi</td>
<td>1b – allow both Bureau of Engineering employees; as well as, any one from the public domain to take a photo and send to the cloud database.</td>
</tr>
<tr>
<td>Cloud/Web Deployment</td>
<td>2a - Azure web services (Database and Containers) are used to house tagged pictures.</td>
<td>2b – may need a more detailed archive for old and/or completed project photos.</td>
</tr>
</tbody>
</table>

5
3.2 User Instructions

3.2.1 iOS

3.2.1.1 Logging On

In order to use P.A.B.O.E you must be a current employee of the B.O.E and have a valid Username and Password.

The Username will be the Employee’s City ID number

The Password will be the same one used when logging into a B.O.E desktop system.

Tap Login to verify the credentials. If everything is correct the employee will have complete access to the app and the Dashboard page (3.X) will appear.

<table>
<thead>
<tr>
<th>Activity</th>
<th>New Capability</th>
<th>Feature Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Application</td>
<td>3a- Can be downloaded from the App and Google Play Store.</td>
<td>3b – may need to update the apps based upon new software releases for both iOS and Android.</td>
</tr>
</tbody>
</table>
3.2.1.2 System Menu

P.A.B.O.E is a tabular app; which means, that the functions of each tab on the app are completely intertwined. The tabs are as follows:

---

### 3.2.1.2.1 Dashboard Tab

The Dashboard allows a User to get a general view of whether the User is able to use the app to the best of its ability.

Tags that are used to describe a photo taken by a User are up to date with the document in the Azure web portal.

Permission Configuration Status tells the User if he/she has granted the app the correct decisions in order for the app to work. I.e. Allow the app access to the Users phone camera.

Photos to be uploaded is a cue that a User can scroll through of tagged photos waiting to be uploaded via Wifi.

NOTE: Tagged Photos are only allowed to be uploaded via WIFI.
3.2.1.2.2 Tagging Tab

The Tagging Tab allows a User to select images; taken from within the app and/or outside the app; that have not yet been tagged and apply them to multiple photos.

Users can either scroll or get a sum total view of all non-tagged images taken within the app and select them here.

Users select images to import into the app for tagging and/or select by scrolling through imported photos.

Here Users select tag(s) for select pictures above and tap the Upload button to send pictures to the Upload cue.
3.2.1.2.3 Camera Tab

The Camera Tab is where Users can take pictures with the App.

Users can preselect Tags for multiple pictures that might be taken at one time within the app.

Users can view selected Tags and/or add/remove them respectively.

Any pictures taken within the app that are not tagged will be sent to the Tagging page for the untagged cue; however, pictures taken with tags are automatically cued for upload over WIFI.

Tap the Open Camera button to access the Users phone camera.
3.2.1.2.4 History Tab

The History tab allows a User to view all photos uploaded to Azure Web Services.

Users can search for specific photos based upon tag names.

 Thumbnails are set up for easy, quick, and efficient viewing with scrollable capability. However, if a thumbnail is tapped a larger full resolution photo will load.
3.2.1.2.5 Settings Tab

The Settings Tab is where a User can confirm whether the app is being run to its fullest potential.

Users can choose to either allow the app to use mobile data to send multiple pictures to Azure web services or to wait for a solid WIFI connection to do so.

Users can choose to save old photos in case a photo is needed on another project and needs to be retagged. Or a User can choose to let all photos that have been uploaded to Azure are automatically deleted.

The delete after X_Days is a feature in which a fixed time frame of when already uploaded photos are preeminently deleted from the app/

Tap the Logout button if you wish to exit the application.
3.2.1.3 Exit System

In order for you to Exit the app you must:

1. Tap on the Settings tab below

2. Tap the Logout button.

NOTE: If you do Logout you will need to reenter your credentials when you start the app up again.
3.2.2 Android

3.2.2.1 Logging In

In order to use P.A.B.O.E you must be a current employee of the B.O.E and have a valid Username and Password.

The Username will be the Employee’s City ID number. However for the current version Username will be blank.

The Password will be the same one used when logging into a B.O.E desktop system. However, for the current version Password will be left blank.

Tap Login to verify the credentials. If everything is correct the employee will have complete access to the app and the Dashboard page (3.X) will appear. If you left Username and Password blank you will be able to Login.
3.2.2.2 System Menu

P.A.B.O.E is a tabular app; which means, that the functions of each tab on the app are completely intertwined. The tabs are as follows:

3.2.2.2.1 Dashboard Tab

The Dashboard allows a User to get a general view of whether the User is able to use the app to the best of its ability.

Tags that are used to describe a photo taken by a User are up to date with the document in the Azure web portal.

Permission Configuration Status tells the User if he/she has granted the app the correct decisions in order for the app to work. I.e. Allow the app access to the Users phone camera.

Photos to be uploaded is a cue that a User can scroll through of tagged photos waiting to be uploaded via Wifi.

NOTE: Tagged Photos are only allowed to be uploaded via WIFI.
3.2.2.2 Tagging Tab

The Tagging Tab allows a User to select images; taken from within the app and/or outside the app; that have not yet been tagged and apply them to multiple photos.

Users can either scroll or get a summary total view of all non-tagged images taken within the app and select them here.

Users select images to import into the app for tagging and/or select by scrolling through imported photos.

Here Users select tag(s) for select pictures above and tap the Upload button to send pictures to the Upload cue.
Tap Add Tags to select tags to apply to selected Pictures. Go to Tagging 3.1.2.4 to see how tags work.
Tap the category in which you want to Tag the image.
You can add more than one Tag by choosing a different context from the drop down to add Attributes.

Add an Attribute to the Context for the Tag.

Once done tap Upload to send picture with tag to the Upload Cue.

Tap Take Another if you wish to take a new picture with new tags.
3.2.2.2.3 Camera Tab

The Camera Tab is where Users can take pictures with the App.

Any pictures taken within the app that are not tagged will be sent to the Tagging page for the untagged cue; however, pictures taken with tags are automatically cued for upload over WIFI.

Tap the Take Picture button to access the Users phone camera.
If you wish to retake the picture tap Take Another.

If you like the picture you have taken tap Next. Which will take you to 3.1.2.4. for Tagging.
3.2.2.2.4 History Tab

The History tab allows a User to view all photos uploaded to Azure Web Services.

Thumbnails are set up for easy, quick, and efficient viewing with scrollable capability. However, if a thumbnail is tapped a larger full resolution photo will load.
3.2.2.2.5 Settings Tab

The Settings Tab is where a User can confirm whether the app is being run to its fullest potential.

Users can choose to either allow the app to use mobile data to send multiple pictures to Azure web services or to wait for a solid WIFI connection to do so.

Tap the Logout button if you wish to exit the application.
3.2.2.3 Exit System

1. Tap on the Settings tab below

2. Tap the Logout button.

NOTE: If you do Logout you will need to reenter your credentials when you start the app up again.
Section 4: Architecture and Design

The main components of this Photo Archive, shown in the DFD above, each belong to one of three major categories: database, storage, or distribution.

4.1 SQL Server Database

This is where all the photos taken and uploaded to Azure will reside. Both Full resolution and thumbnail will be stored here along with latitude, longitude, context, attribute, and UserID.
4.2 Azure Storage Container

This is where the Blob resides; condensed files from the database. The History module 2.4 calls this in order to view and edit photos within the database.

4.3 Distribution

*Dashboard - Module 2.1*

The Dashboard module will allow a user to check if their permissions are configured correctly; as well as, see real time up-to-date uploading cue. The User will also be allowed to edit/remove a photo’s context and attributes from the upload cue.

*Camera - Module 2.2*

This module will allow a User to take a photo with or without contexts and attributes associated with it. If the User takes a photo with contexts and attributes it will automatically go to the Upload cue. If not it will be redirected to the Tag Module for tagging of contexts and attributes.

*Tag - Module 2.3*

This module allows a User to add contexts and attributes to multiple photos at once. Photos can be taken from the App or from the Users local photo library.

*Mobile Device Image Gallery*

Access to the Users phones photo library.

*History - Module 2.4*

A query from the SQL server database of all photos ever uploaded by the User.

*Settings - Module 2.5*

This module will allow a User to upload photo’s over Wifi only; as well as, allow a user to delete photos from the app after a specified number of days. Settings will also allow a User to log out of the App. However, once the User is logged out a User must reenter his/her credentials at the Dashboard module.
Section 5: Conclusion

In conclusion, we were able to fully realize the front end portion of both iOS and Android applications; with basic push and pull to the Azure Database on the Android side. We failed to provide bug free backend for Android and any backend at all for iOS; however, our liaisons, after informing them, were still quite pleased and happy with the result. They, the Bureau of Engineering, will continue to work on the backend for iOS and debug Android; as well as; add some more functionality to both apps for general public use. Finally, the most important lesson that we learned, in this year long project, was that researching and/or familiarizing oneself with a completely new language is extremely difficult and time consuming but well worth the rewards.

Section A: ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASC</td>
<td>Azure Storage Container</td>
</tr>
<tr>
<td>B.O.E</td>
<td>Bureau of Engineering</td>
</tr>
<tr>
<td>CM</td>
<td>Camera Module</td>
</tr>
<tr>
<td>DM</td>
<td>Dashboard Module</td>
</tr>
<tr>
<td>HM</td>
<td>History Module</td>
</tr>
<tr>
<td>MDIG</td>
<td>Mobile Device Image Gallery</td>
</tr>
<tr>
<td>PABOE</td>
<td>Photo Archive for the Bureau of Engineering</td>
</tr>
<tr>
<td>SM</td>
<td>Settings Module</td>
</tr>
<tr>
<td>SSD</td>
<td>SQL Server Database</td>
</tr>
<tr>
<td>TM</td>
<td>Tag Module</td>
</tr>
</tbody>
</table>