

Audit

A Senior Project
presented to
the Faculty of the Computer Science Department
California Polytechnic State University, San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science

Tammy Kong, Jordan Tang, Myron Zhao
June 2016

Senior Project Final Report

Tammy Kong, Jordan Tang, Myron Zhao

Audit, Winter 2016 - Spring 2016

Faculty Advisor: Lubomir Stanchev

Overview

Audit is an application that aims to create a center of user experiences of online shopping. Through observation and personal experience, we felt there was not an adequate way of online shoppers to verify the legitimacy of a website. Online shopping has become extremely prominent over the past decade, and one of the pitfalls of this is that it is relatively easy to create a fraudulent business. The Audit application serves as a review site that is similar to Yelp, but geared towards online retailers. The application allows users to create an account and leave reviews for companies that they have had experiences shopping at. If the company does not exist within the site's database, they are given the option to request a new company to be added.

Project Goals

Initial goals included...

- Working together through a development cycle of a large project
- Explore frameworks, tools, and languages to assist and aid in both web and android development
- Understand the process of producing software in an agile method
- Learn the differences of developing a web application and an android application

Deliverables

We created the following to complete this application

- Created initial requirements document and plans for each sprint cycle
- Designed horizontal prototype with balsamiq
- Deployed the website with stable backend
- Created Android version to mirror functionality of the website

Technologies Used

This section details the different technologies we used in order to create our application. We wanted to use a backend service that would be easy to setup and quick to learn. Parse was a service that we had experience with in the past, so it was our go to service for our backend. Since our application was mainly user-facing, we tried to search for a front-end framework that would make it easiest for us since all of the group members did not have any front-end

experience. We decided to use Bootstrap, which is an HTML, CSS, and JS framework that allows the website to also be mobile friendly.

Back-end

Parse and Javascript

A lot of group members did not have backend experience, so we searched for a service that we could set up quickly. Parse was a service that we had a little experience with in the past, which made setting it up seamless. Documentation provided for Parse was also very clear and easy to follow. It was relatively simple to query things to and from the Parse database, and did not require knowledge of any database languages. Parse also provided a lot of features that were required to create our applications. One of the most helpful was a User object already provided by the API. The service was already set up to accept user sign ups and sign ins. What also attracted us to using the service was its compatibility with both Javascript and Android. It was a service we could set up one database across both of our applications. We also wanted to take advantage of a Facebook sign in that Parse provided, but we did not have the time to complete that feature.

There were definitely some downsides to using Parse. There were some issues handling asynchronous calls to the database, which hindered our ability to put in some custom features. There were also a lot of small issues we had with Parse which were inflated because of our overall inexperience using their API. Familiarity with the platform would have definitely improved the speed at which we developed.

Using Javascript worked for the most part, but some issues we had with it arose for the same reasons as using Parse. As a group, we had a general inexperience with using the language, so it made more complex features we wanted to implement difficult to figure out.

There are also still some questions to be answered using Parse as our backend. We did not do any large scale testing, so we are unsure of how the product performs under a large number of users. We also believed we set up the database in a way that avoided any issues if multiple people were querying simultaneously, but again this was not stress tested and may have some unexpected behaviors under heavy stress.

Front-end

Bootstrap

Bootstrap is a front end framework that provides both HTML and CSS design templates, forms, buttons, and other parts for front end development. It is a framework purely made for front end

development, and the design templates made it easier for us to use. Since the group members did not have a lot of front end experience, using templates to speed up the development process was almost a must. For the most part, using bootstrap turned out to be the correct choice, as there were a lot of UI formatting issues we probably would not have been able to solve without it.

HTML / CSS / Javascript

The three “core” languages for web development. HTML is the markup language used to render web pages. CSS describes the presentation of the page. Javascript denotes any logic or backend calls that need to be made. Using these three languages were necessary for creating our web application.

Deployment

AWS

We deployed our web application on our AWS EC2 instance in order to learn about deploying a website and system configurations. Using AWS at first was tricky because it was difficult finding good documentation or a reliable tutorial. We were eventually able to figure it out and hosted our website by setting up the server. However, in terms of maintainability, using AWS would require some effort since we would need to consistently update the website with any new features implemented.

Workflow

In order to manage our code, we used a shared repository on GitHub. This allowed us to work in parallel and track any changes we made to our codebase. Our main lines of communication was through group chat messages. We also worked in an agile method, but we did not use any boards such as Jira to keep track of tasks and issues to be resolved. The work was managed in a very informal way, and our methods probably would not have scaled at all to a larger team. If we were to change one thing, we would emphasize the importance of using some software such as Jira to track issues and tasks that everyone can see. Making a list of tasks visible to all group members would have greatly improved our development.

Development

Initial Documents and Horizontal Prototype

Before any technical development, we outlined the order of features developed throughout each sprint so we could better gauge the progress of our work. After this was approved by our advisor, we proceeded to create a horizontal prototype, which is a low-fidelity mockup of the different states and screens of our application, and how each screen flows to each other. Once

complete, we researched necessary technologies to develop the applications, which we used to create a vertical prototype. The vertical prototype simply uses each piece of technology, API, or tool to confirm that they all have the expected behavior and provide the necessary features for the application.

We also created the following use cases for our application:

Use Case 1: Signing Up for an Account

Use Case ID:	1
Use Case Name:	Signing Up for an Account
Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016
Actors:	Unregistered Audit User
Description:	A user is interested in using the Audit website to browse and rate online companies. The user will navigate to the Audit webpage and select the “Sign Up” button. Their page will be redirected to the sign up page, where the user will enter their information for their new account. After submitting their information, the user will receive an email confirming their account’s validity.
Preconditions:	<ol style="list-style-type: none"> 1. The user has a valid email address 2. The user has a stable internet connection
Postconditions:	<ol style="list-style-type: none"> 1. User receives a confirmation email to activate their account
Normal Flow:	<p>1.0 Successful Sign Up</p> <ol style="list-style-type: none"> 1. User navigates to the Audit homepage in their web browser 2. Webpage displays links to log in, sign up, search, or find more information about the website 3. User chooses the “Sign Up” option 4. Website redirects to the sign up page, which prompts the user to enter their first name, last name, email address, and password 5. User fills out all fields and submits their information 6. Webpage thanks the user for signing up and presents “Ok” button 7. User presses “Ok” and is redirected to the homepage

	8. User receives a confirmation email and validates their account
Alternative Flow:	<p>1.1 User decides not to enter personal information (branch at step 5)</p> <ol style="list-style-type: none"> 1. User selects the “Cancel” button 2. Return to step 1 <p>1.2 User attempts to log in without an account</p> <ol style="list-style-type: none"> 1. User attempts to log in via the log in page 2. User cannot log in without an account 3. User is prompted to try again, sign up, or reset account information via email 4. User selects “New? Sign up here!” button 5. Return to step 4
Exceptions:	<p>1.0.E.1 User does not receive confirmation email</p> <ol style="list-style-type: none"> 1. User selects “Contact” button on homepage 2. User is redirected to information page 3. Information page displays contact email and phone number 4. User contacts Audit staff through an email or phone call 5. User reports their issue 6. Audit staff manually resolves issue <p>1.0.E.2 User enters invalid/taken email</p> <ol style="list-style-type: none"> 1. System will notify the user that the email cannot be used and will prompt them to try again 2. Return to step 5
Includes:	None
Priority:	High
Frequency of Use:	Once per user
Business Rules:	None
Special Requirements:	None
Assumptions:	User does not already have an account
Notes and Issues:	None

Use Case 2: Logging In

Use Case ID:	2
Use Case Name:	Logging In
Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016
Actors:	Registered Audit User
Description:	An Audit member wishes to log in to their account.
Preconditions:	<ol style="list-style-type: none">1. User has an Audit account2. User has a stable internet connection
Postconditions:	<ol style="list-style-type: none">1. User is granted access to their account
Normal Flow:	<p>1.0 Successful Log In</p> <ol style="list-style-type: none">1. User navigates to the Audit homepage in their web browser2. Webpage displays links to log in, sign up, search, or find more information about the website3. User selects “Log In” button4. User is redirected to log in page5. User is prompted for email and password6. User submits their information7. User is redirected to the homepage logged into their account
Alternative Flow:	None
Exceptions:	<p>1.0.E.1 Incorrect Account Information</p> <ol style="list-style-type: none">1. System notifies the user that their combination of account name and password is incorrect2. User is prompted to try again, sign up, or reset account information via email3. Return to step 5
Includes:	None
Priority:	High
Frequency of Use:	Once per time user uses Audit
Business Rules:	None

Special Requirements:	None
Assumptions:	Registered users will log in when they use Audit
Notes and Issues:	None

Use Case 3: Submitting a Review

Use Case ID:	3
Use Case Name:	Submitting a Review
Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016
Actors:	Registered Audit User
Description:	An Audit member wishes to leave feedback on a company. The user looks for the company using the search bar. The user will either find the company's review page or be prompted to create the company's review page if one does not already exist. The user can then leave a review on the company's review page.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged into their Audit account 2. User has a stable internet connection
Postconditions:	<ol style="list-style-type: none"> 1. A review is left on the company's review page
Normal Flow:	<p>1.0 Company Page Found</p> <ol style="list-style-type: none"> 1. User searches for the company via the search bar 2. Company page is retrieved from the database 3. User is redirected to company page, which displays the store name, average ratings, location, website, reviews, and a "Write A Review" button 4. User selects "Write A Review" and is redirected to the Write A Review page 5. User is prompted to leave a comment and rate the company on customer service, shipping, and quality out of 5 stars 6. User fills out at least one field 7. User submits the review and is redirected to the company's review page, where their review is added

Alternative Flow:	<p>1.1 Company Page Not Found (branch at step 2)</p> <ol style="list-style-type: none"> 1. User is redirected to the “Store Not Found” page, which displays a “Be The First To Write A Review” button 2. User selects “Be The First To Write A Review” 3. User is redirected to the first review page, which prompts the user to enter the company name and company website, along with the normal review options 4. User fills in the company name and website (if available) 5. Return to step 5 <p>1.2 User decides not to enter information (branch at step 6)</p> <ol style="list-style-type: none"> 1. User selects the “Cancel” button 2. User is redirected to the company review page
Exceptions:	None
Includes:	None
Priority:	High
Frequency of Use:	High usage, as it is the main purpose of the site
Business Rules:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case 4: General Browsing

Use Case ID:	4
Use Case Name:	General Browsing
Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016

Actors:	Any user registered or unregistered
Description:	A shopper is debating whether or not to purchase merchandise from a particular online store. The user wishes to find reviews to gauge the quality and reliability of the shop.
Preconditions:	1. User has a stable internet connection
Postconditions:	1. User finds reviews for the store
Normal Flow:	<p>1.0 Company Page Found</p> <ol style="list-style-type: none"> 1. User navigates to the Audit homepage 2. User searches for the company via the search bar 3. User is redirected to company page, which displays the store name, average ratings, location, website, reviews, and a “Write A Review” button 4. User scrolls down to see all reviews made by Audit users
Alternative Flow:	None
Exceptions:	<p>1.0.E.1 Company Page Not Found</p> <ol style="list-style-type: none"> 1. User is redirected to the “Store Not Found” page, which displays a “Be The First To Write A Review” button 2. User cannot find reviews for the shop on Audit
Includes:	None
Priority:	High
Frequency of Use:	High usage, as it is the main purpose of the site
Business Rules:	None
Special Requirements:	Company has a review page on Audit
Assumptions:	Reviews exist for the company in question
Notes and Issues:	None

Use Case 5: Finding Information About Audit

Use Case ID:	5
Use Case Name:	Finding Information About Audit
Created By:	Myron Zhao

Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016
Actors:	Any user registered or unregistered
Description:	A person on the website wishes to learn more about Audit.
Preconditions:	1. User has a stable internet connection
Postconditions:	1. User finds information about Audit
Normal Flow:	<p>1.0 Finding Information About Audit</p> <ol style="list-style-type: none"> 1. User navigates to the Audit homepage 2. Webpage displays links to log in, sign up, search, or find more information about the website 3. User selects “About”, “Contact”, “Help”, or “Terms” 4. User is redirected to information page, which contains information about Audit, contact information, help on using Audit, and the terms of service
Alternative Flow:	None
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Low usage, most users will assume they know how to use the site and be uninterested in extra information
Business Rules:	None
Special Requirements:	None
Assumptions:	User needs more information for help
Notes and Issues:	None

Use Case 6: Reporting Abuse

Use Case ID:	6
Use Case Name:	Reporting Abuse

Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016
Actors:	Registered Audit User
Description:	An Audit user finds a review that is against the Audit terms of service. The review may be excessively offensive, irrelevant, or spam. The user wishes to report it to Audit for removal.
Preconditions:	<ol style="list-style-type: none"> 1. User is on the review page for a company 2. User has a stable internet connection
Postconditions:	<ol style="list-style-type: none"> 1. Review is sent to Audit staff for inspection
Normal Flow:	<p>1.0 Reporting Abuse</p> <ol style="list-style-type: none"> 1. User is on company page, which displays the store name, average ratings, location, website, reviews, and a “Write A Review” button 2. User scrolls down to find a review against the terms of service 3. Review displays information about the reviewer, ratings for the company, their comment, votes on the review, and a “Report Abuse” button 4. User selects “Report Abuse”
Alternative Flow:	None
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Low usage, most users will abide by the terms of service
Business Rules:	None
Special Requirements:	A review that breaches the terms of service
Assumptions:	A minority of Audit users ignore the terms of service
Notes and Issues:	None

Use Case 7: Navigating to Your Profile Page

Use Case ID:	7
Use Case Name:	Navigating to Your Profile Page
Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016
Actors:	Registered Audit User
Description:	An Audit user wishes to navigate to their profile page. The user may want to see or edit their information or see their reviews.
Preconditions:	<ol style="list-style-type: none">1. User has a registered Audit account2. User has a stable internet connection
Postconditions:	<ol style="list-style-type: none">1. User is redirected to their profile page
Normal Flow:	<p>1.0 Navigating to Your Profile Page</p> <ol style="list-style-type: none">1. User logs in to their Audit account (see Use Case 2)2. User is redirected to the home page, but is now logged in3. Homepage now displays their personal username, a gear symbol, and a “Sign Out” button in the upper right corner4. User selects “[Username]” button5. User is redirected to their profile page, which displays their profile picture, full name, location, and reviews
Alternative Flow:	None
Exceptions:	<p>1.0.E.1 User Cannot Log In</p> <ol style="list-style-type: none">1. User cannot log in due to incorrect username/password combinations or lack of account2. User cannot access the account through the upper right corner
Includes:	None
Priority:	Medium
Frequency of Use:	Low usage, most users will not need to view or change account information very often
Business Rules:	None

Special Requirements:	None
Assumptions:	User need to view or edit account information
Notes and Issues:	None

Use Case 8: Navigating to a Company's Website

Use Case ID:	8
Use Case Name:	Navigating to a Company's Website
Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016
Actors:	Any user registered or unregistered
Description:	An Audit user is interested in a company after reading reviews on Audit. The user wishes to go to their website in order to learn more or purchase their merchandise.
Preconditions:	1. User has a stable internet connection
Postconditions:	1. User is redirected to the company's webpage
Normal Flow:	<p>1.0 Navigating to a Company's Website</p> <ol style="list-style-type: none"> 1. User navigates to a company's review page via the search bar on the Audit homepage 2. Review page displays the store name, average ratings, location, website, reviews, and a "Write A Review" button 3. User selects the "[Website]" button 4. User is redirected to the company's webpage
Alternative Flow:	None
Exceptions:	<p>1.0.E.1 Website Link Does Not Exist</p> <ol style="list-style-type: none"> 1. Audit users have not yet added a link to the company's webpage 2. User is prompted to add a link to the company's webpage 3. User cannot navigate to company's webpage through Audit

Includes:	None
Priority:	Medium
Frequency of Use:	Medium usage, some users will want to purchase merchandise from companies if they deem their reviews adequate
Business Rules:	None
Special Requirements:	The company in question has a webpage
Assumptions:	The company's webpage has been linked to its Audit review page
Notes and Issues:	None

Use Case 9: Uploading a Profile Picture

Use Case ID:	9
Use Case Name:	Uploading a Profile Picture
Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016
Actors:	Registered Audit User
Description:	An Audit user wishes to add a profile picture to their account.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in to their Audit account 2. User has a stable internet connection
Postconditions:	<ol style="list-style-type: none"> 1. User's profile picture is updated
Normal Flow:	<p>1.0 Uploading a Profile Picture</p> <ol style="list-style-type: none"> 1. User navigates to their profile page (see Use Case 7) 2. User is redirected to their profile page, which displays their profile picture, full name, location, and reviews 3. User selects the "Edit Picture" button 4. A file manager for their machine appears in a new window 5. User navigates to the directory containing the desired picture 6. User selects the "Insert" button 7. File manager closes itself

	8. User's profile page is reloaded with the new profile picture
Alternative Flow:	None
Exceptions:	<p>1.0.E.1 Incorrect File Type</p> <ol style="list-style-type: none"> 1. System notifies user that the file selected is not a supported picture type and prompts the user to try again with a different picture file 2. Profile picture is not updated <p>1.0.E.2 Picture Is Too Large</p> <ol style="list-style-type: none"> 1. System notifies user that the picture is too large to fit the profile picture dimensions 2. System notifies user that picture must be cropped before it can be used and prompts them to try again 3. Profile picture is not updated
Includes:	None
Priority:	Low
Frequency of Use:	Low usage, most users will upload a photo once or not at all and find no need to update their picture
Business Rules:	None
Special Requirements:	None
Assumptions:	The user selects an appropriate picture for their profile
Notes and Issues:	None

Use Case 10: Voting on a Review

Use Case ID:	10
Use Case Name:	Voting on a Review
Created By:	Myron Zhao
Last Updated By:	Myron Zhao
Date Created:	January 10, 2016
Date Last Updated:	January 10, 2016

Actors:	Registered Audit User
Description:	An Audit user is invested in the Audit review process and wishes to improve the Audit user experience by voting on reviews to help sort them.
Preconditions:	<ol style="list-style-type: none"> 1. User is logged in to their Audit account 2. User has a stable internet connection
Postconditions:	<ol style="list-style-type: none"> 1. A review's vote count is updated and review is sorted
Normal Flow:	<p>1.0 Upvoting a Helpful Review</p> <ol style="list-style-type: none"> 1. User navigates to a company's review page (see Use Case 4) 2. User finds and reads a review 3. Review displays information about the reviewer, ratings for the company, their comment, votes on the review, and a "Report Abuse" button 4. User selects the "Yes" button next to "Was the review helpful?" 5. Review's vote count is updated and review is sorted
Alternative Flow:	<p>1.1 Downvoting an Unhelpful Review (branch at 4)</p> <ol style="list-style-type: none"> 1. User selects the "No" button next to "Was the review helpful?" 2. Return to step 5
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of Use:	Medium usage, some users will want to give their feedback on others' feedback
Business Rules:	None
Special Requirements:	None
Assumptions:	User cares enough to vote on reviews
Notes and Issues:	None

Website Development

After confirming that Parse was usable across both our web and Android platform, we decided to focus development on the website first. The initial timeframe for the website was only Winter quarter, but the process of learning web technologies slowed down development, eventually bleeding into the first half of Spring quarter. The website contains most of the functionality mentioned in our initial proposal with only small, non-vital features missing such as review sorting.

Android Development

Since development on the website took up a portion of Spring quarter, we were not able to begin development on the Android app until much later than expected. Because of this, the main goal for our android version was to get the main features of the application working. These features were company search, user profiles, and review submitting.

Main Issues

Asynchronous Querying

One of the biggest issues we faced during this process was how to handle asynchronous query calls to the Parse database. This was largely an issue because we would query the database for information, which would create another thread as the rest of the Javascript completed. The following Javascript code usually relied on the information being fetched from the database so it could be displayed on the web page. This would result in the Javascript completing without the necessary information to display on the page. We solved this issue by using Parse's asynchronous handler, which they call a "Promise". While it seemed easy to use at first, there were definitely some intricacies to them we had to figure out, mainly chaining multiple promises together. Getting comfortable with Parse's Promise scheme took a bit of learning, but allowed us to display the necessary information on our web page.

Element layout with Javascript on HTML DOM

Another issue we faced was using Javascript to generate elements and place them on the HTML Document Object Model, or DOM. The DOM is where all the elements of the HTML document exist, and the Javascript can access the DOM to change, add, or remove elements. While it was relatively easy to add and remove elements, it was difficult formatting them in the exact format that we wanted. There were times when we had to choose to use elements like a table in order to get the layout that we wanted.

Testing

Given the limited time we had for development, we did not employ an adequate amount of testing compared to a professional software team. Since we spent a lot of time learning new technologies and focusing on development, it was hard to split development time and testing time. It is also difficult to automate testing for web and mobile applications because so much of the application testing hinges on UI testing. However, we also determined that even in industry it is considered difficult to do automated UI testing, so we decided it was a fair tradeoff. The UI was modified a lot during development which also made testing difficult.

Reflection

Takeaways

The difficulty of the project was underestimated by our group. Learning the necessary technologies to create the website took much longer than expected, and there was also a smoothing period that involved us getting comfortable with the technologies as well. We could have taken advantage of time during the quarter before to learn the technologies and languages we were using so we could quickly begin development during the winter. Fortunately, we had a group member who was taking the Android development class, which made creation of the Android application much easier. This worked out in the end because it allowed us to spend more time on the website, and then quickly put together an Android application. We could have done more research on using some software tool that would have allowed us to develop the web and mobile platform simultaneously such as NodeJS or ReactJS.

We also wished we used some sort of issue and task tracking software such as Jira. Our main line of communication was through an online group chat, but this made it difficult to figure out tasks and issues to complete when people were not available or immediately responsive. We've had experience using Jira in our capstone classes, and having the issues on a public board would have increased productivity.

Because we spent so much time in development, we never found the time to promote our product. It would have been nice to get the experience of promotion and entrepreneurship, but the focus of our project was to get all functionality implemented. To improve development, we initially thought it would be beneficial to focus on the website first and then move on to the Android application. However, looking back, development of the Android application could have definitely been relegated to one group member, with the other two members focusing on the website.

Future Work

The current state of the project is probably not stable enough to be deployed for public use. However, we did learn a lot as a group and it was a very good experience of developing a piece

of software from scratch. The application would definitely need more real world testing from users, test groups, and stress tests.