eZsell

by

Austin Scheidt, Matt Blume, Scott Cramer

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

_*Austin Scheidt*_____	_04/20/2020_____
Austin Scheidt	Date
_*Matthew Blume*_____	_04/20/2020_____
Matt Blume	Date
_*Scott Cramer*_____	_04/20/2020_____
Scott Cramer	Date
_*Abdou Fall*_____	_04/20/2020_____
Abdou Fall, Faculty Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2020

TABLE OF CONTENTS

List of Illustrations

Figures

Tables

ABSTRACT

eZsell is an online marketplace for buying and selling surplus technology. According to a study done by Statista, the average lifespan of IT equipment is five years, and the Department of Information Resources in Texas claims that maintaining equipment past this lifecycle can be up to 59% more expensive than maintaining new equipment. eZsell allows companies, universities, and individuals to recoup some of these maintenance costs by providing a quick and convenient way to market their excess technology. Users of this application are able to sell technology that would otherwise be more expensive to maintain, facilitating a market of usable bargain surplus for other interested parties.

1. INTRODUCTION

1.1 Problem Statement

It would be difficult to find an IT department that does not have a large inventory of surplus hardware. Excess hardware can be an expensive hassle to the department and can cost companies many hours of inventory and upkeep. Spiceworks tells us that two-thirds of large enterprises (5,000+ employees) plan to deploy 5G technology by 2021. This means these companies will be looking to upgrade their old technology by 2021 to meet new standards.

eZsell aims to combat this expense by providing an easy to use solution that acts as a medium to market equipment that has been moved to surplus. The intent of eZsell is to give people and companies a simple option to profit off hardware that would otherwise tax their time and space. The application must be cross-platform to ensure usability for all parties and will feature an interface that is intuitive and effective so that dealing with surplus via our medium is simpler than stockpiling and managing an ever-growing pile of surplus hardware.

1.2 Solution

eZsell is an online marketplace for buying and selling used computers/parts. This service will provide a website and corresponding mobile app that allows users to easily put any hardware up for sale and allow people who are nearby to be able to easily purchase the product. An alternative to eZsell would be large online marketplaces such as Amazon or eBay. eZsell aims to beat these sites by providing a solution tailored entirely to the realm of technology. Another alternative would be govdeals, a website that focuses on bulk surplus. eZsell will allow for

companies and universities to sell in bulk and accommodates individual sales. In this marketplace the buyers and sellers will be able to interact through a messaging system to get the final details before purchasing a product.

The target audience is companies who have policies for replacing computers after a couple years. This allows companies to make back some money on those older machines that they might normally just recycle. Another target audience for eZsell is anyone who is 18 years of age or older and who needs to buy or sell excess hardware. To make buying and selling as easy as possible we will have an online web version and a mobile version of the marketplace.

1.3 Project Goals

Develop a marketplace that can be accessed via online web application or as a mobile application. This will provide people in the computer community to be able to create, read, update, and delete posts for a wide variety of computers/parts. You will also be notified if you have a post and someone messages you about it.

eZsell is going to be an application that has a clean simple design that is hosted on an Azure server. The web application was developed using ASP.NET Core Framework. The mobile app was done using Xamarin Studios. Front-end development was accomplished using Bootstrap. Bootstrap provides simple implementation and clean visuals. Using C#, ASP.NET allows us to easily build sleek user interfaces (UI), write services to handle our business logic, and write repositories to interface with our database backend. Xamarin uses C# to simply and quickly build mobile apps for iOS, Android, and Windows Phone devices. Lastly, eZsell stores inventory data on Microsoft Standardized Query Language (SQL) Server and will protect in-app user information.

1.4 Overview

For the remainder of this final report it will outline in detail how our project was completed. The report includes the following sections project concept/solution, Design Objectives, Methodology, User Profile, Use Case Diagram, Gantt Chart, and Problems we encountered during the process.

2. DISCUSSION

2.1 Project Concept/Solution

2.1.1 "Big Picture"

Develop a sleek, user friendly web application (app) and corresponding mobile app

eZsell meets the project scope through the following items:

- All the features mentioned in Project Goals
- Sleek web application with a visually appealing graphical interface
- Front end written primarily in C#, along with Hyper Text Markup (HTML) and Bootstrap
- Back end written primarily in C#
- Mobile App developed in C# using Xamarin Studios
- Database written entirely in Microsoft SQL Server
- Site access secured with modern authentication tools

2.1.2 Origin

eZsell was inspired by Austin Scheidt, Matthew Blume, and Scott Cramer. The group had discussed the possibility of an online marketplace portal for used items. This idea was expanded upon by Scott with the idea to focus on surplus hardware. As the idea was discussed, Matt had the idea to implement an Application Program Interface (API) that would make it possible to design a web service, as well as a mobile app for the idea. As a group we discussed features that would make the service a marketable and attainable idea.

Before the semester began, the team conversed to narrow down the list of project ideas and eZsell was chosen for its necessity and marketability. The team figured that most IT departments struggle with surplus equipment as technology evolves and solutions are made. After determining that the service would be applicable to many departments the decision was finalized, and eZsell was the result.

2.2 Design Objectives

2.2.1 Break Down

- All users will be required to create an account when they want to buy or sell
- There will be a search function incorporated for people who are coming to the application to find something specific
- A wish list will be implemented so that users can save items they might want to buy for a later date
- The buyers and sellers will be able to communicate back and forth via a messaging system when a buyer is interested in a seller's item

2.2.2 Goals that were Abandoned

- The search function was abandoned because we ran into a time crunch and felt like it wasn't an essential part for our application to work
- The ability to communicate back and forth via a messaging system was also abandoned due to time concerns and making sure we had our main features working correctly

2.3 Methodology

Following the design requirements laid out in class allowed us to meet our goals by providing us with a solid foundation to work from. Having deadlines to meet kept us focused and on track to complete our project on time.

Following our procedures allowed us to reach our goals by giving us guidelines on how work should be done. This allowed us to keep up with deadlines and develop our application in an efficient, effective manner

2.4 User Profile

We have concluded that the End User is the only type of user that will be interacting with our application. The potential users for our eZsell application are users who either have surplus technology to sell or you are looking to buy some older technology. The users will be able to list new items in bulk or individually. They will also be able to buy these items in bulk or individually.

Buyer Table:

User Profile Form
PROJECT: Online Technology Surplus Marketplace (eZsell)

POTENTIAL USERS:

- Companies (Buy)
- Universities (Buy)
- Individuals (Buy)

SOFTWARE, INTERFACE, AND RELATED EXPERIENCE:

This application is primarily focused on providing a bulk surplus marketing solution for technology departments that have an abundance of non-essential hardware. This service is designed for individuals that will have experience in the technology field and those likely to have excess hardware.

Those without hardware to sell within the service will still find value in the application in that they will have the ability to purchase either in bulk or individual components.

EXPERIENCE WITH SIMILAR APPLICATIONS:

- Amazon
- eBay
- Govdeals

TASK EXPERIENCE: • Buying items for sale on Amazon • Using an application to buy technology • Navigating through an application on a web browser or mobile phone
FREQUENCY OF USE: Whenever an individual or a company has a technological item/s they want to buy. This can occur daily, weekly, monthly, or yearly.
KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS: • Quickly buy items that are up for sale • Easily navigate the site as a seller • Easy to search for specific items you need

Table 1

Seller Table:

User Profile Form

PROJECT: Online Technology Surplus Marketplace (eZsell)

POTENTIAL USERS:

- Companies (Sell)
- Universities (Sell)
- Individuals (Sell)

SOFTWARE, INTERFACE, AND RELATED EXPERIENCE:

This application is primarily focused on providing a bulk surplus marketing solution for technology departments that have an abundance of non-essential hardware. This service is designed for individuals that will have experience in the technology field and those likely to have excess hardware.

Those without much hardware experience within the service will still find value in the application in that they will have the ability to sell either in bulk or individual components.

EXPERIENCE WITH SIMILAR APPLICATIONS:

- Amazon
- eBay
- Govdeals

TASK EXPERIENCE:

- Posting items for sale on eBay
- Using an application to sell technology
- Navigating through an application on a web browser or mobile phone

FREQUENCY OF USE:

Whenever an individual or a company has old technology they don't need anymore. This can occur daily, weekly, monthly, or yearly.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Quickly put items up for sale
- Easily navigate the site as a buyer
- Easily create individual or bulk listings

Table 2

2.5 Use Case Diagram

Figure 1: Use Case Diagram introduces the project's use case diagram. eZsell has two parties in the use cases. Both groups, buyers and sellers have the same access to buying and selling. A buyer finds other listings available for purchase on the global listings page, and a seller would list items via the create listing page.

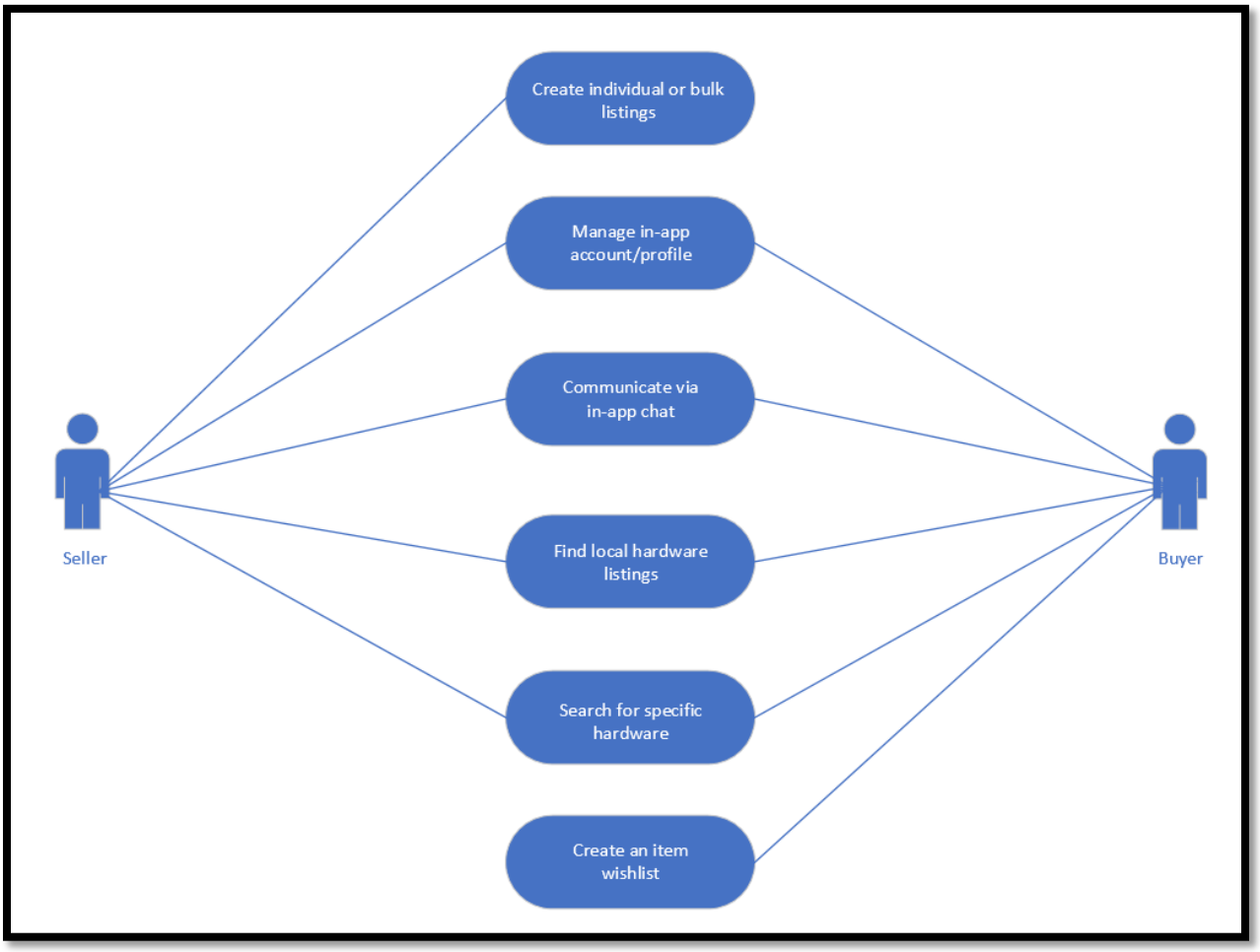

Figure 1

2.6 Gantt Chart

Figure 2: Fall Project Schedule and Gantt Chart is our Fall project schedule with the major deliverables listed.

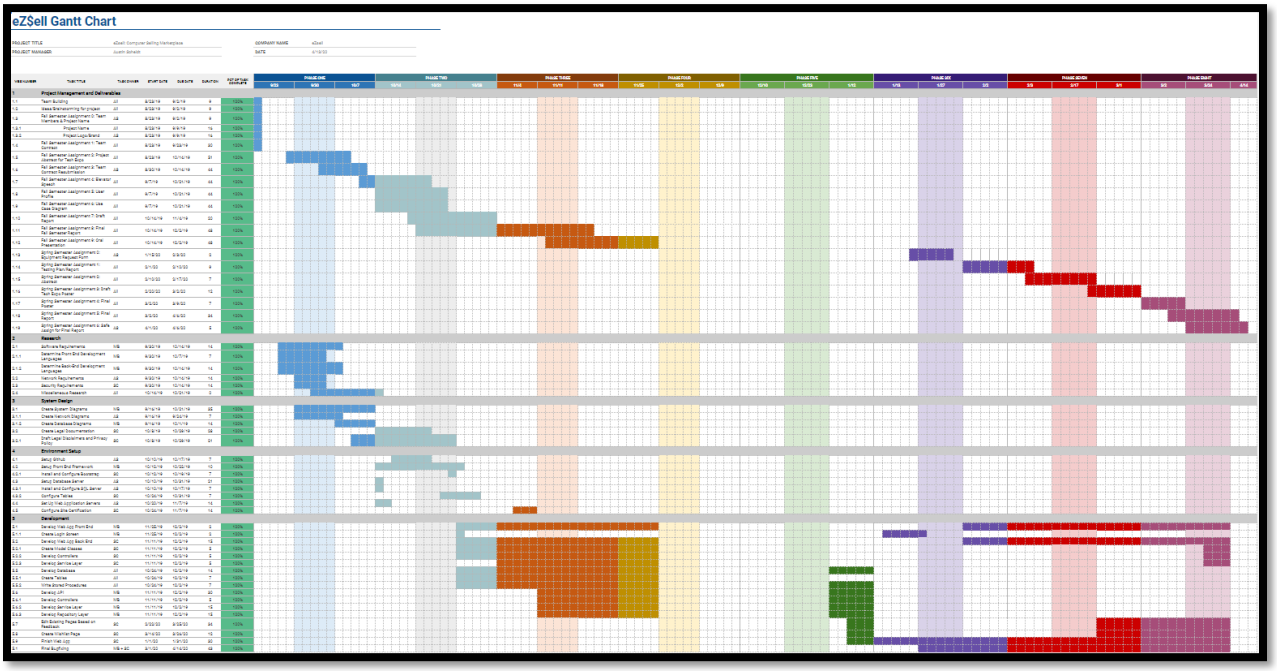

Figure 2

2.7 Technical Architecture

2.7.1 Network

We are using Azure web services to host our website. We decided on Azure because it offers a wide variety of services to help maintain, protect, and build websites. Although, we can't use many of those services with our project, we can use services like backup to allow us to back up our site however often we want. We also used Azure to create our database which we will cover in a future section.

2.7.2 Application

Our application uses C# for the methods that implement the interactivity of our service. The frontend was developed using the Jumpstart bootstrap framework and uses JavaScript and scss to animate and render specific functionalities of the web service. The backend is comprised of logic using the C# language.

2.7.3 Database

Our database was developed using Azure MySQL. We developed a schema, as well as wrote multiple stored procedures used by our API. Our table structure includes a User table, where we store user information, a preferences table, where we store user preferences, as well as a Listing table, where we store relevant information about each product listing.

2.7.4 Security

For security in our project since we are using Azure, we get a highly secure cloud foundation managed by Microsoft. With this protection we also have access to other services within Azure, such as, application insights which allows us to monitor and setup triggers to monitor the health of our site.

2.8 Testing

2.8.1 Overview and Methodology

With our testing methodology, we had two main ideas for the overall testing strategy:

1. An agile approach by the team where we tested certain features in house every week
2. User testing to make sure we have all the functionality that we are expecting

The first way we went about testing was doing everything with an agile approach. This allowed us to make necessary changes before too many other features would be implemented, therefore, making the change would not impact any of the other features throughout our application.

Agile Workflow:

1. The developers work on creating a new feature
2. We would meet on Sunday each week and test new features as a team to make sure they work the way we expect
3. If it works as expected, we move on to adding more features.
4. If it does not work as expected, then that is the first feature we go back to and fix before we move onto something new
5. Create more new features and start back at the top of the workflow

Since this application was all our ideas and we are doing it on our own, we do not need to meet with anyone from some company that may have a stake in the project. This is a big reason why we feel like our weekly sprints are a good idea in our situation because in most cases you're not going to be able to get everyone involved to meet on a weekly basis and hash out problems.

For our second set of tests we decided to do user/functionality testing. This allowed us to get outside feedback in terms of how things work, how the users feel while using the application and if the user has any suggestions on how to make it better. Our users for this test case include friends, family, colleagues, and peers. User testing was vital to our project because everyone working on it already had a preconceived notion of how it works. Feedback from someone who was unfamiliar with how the system should work was very valuable.

User Testing Workflow:

1. Find a willing friend, family member, and just someone who would like to help out
2. Give them the application URL and have them navigate the site
3. They write down any issues they had while testing and if they have any recommendations on doing something differently
4. After the user does their part we ultimately decide if we see any changes necessary
5. If we do make a change based on feedback, we hand it back to the same person who tested it before and see if it works how they expected.
6. If we decide against a change then we just move on with what we already have

This testing provided us with a lot of useful feedback that may have never been brought to our attention without allowing the end users to test our application.

2.8.2 Scope of Testing

Our Test Cases covered all major features we are building in this system. Our test strategy included testing for both End Users in the application using a web browser. These end users include the buyers and the sellers. We knew that it would not be enough to conduct all the testing just amongst the team because there are a lot of parts to the application and something small could easily be overlooked by our team.

2.8.3 Test Plan

Table 3: Test plan shows all the test cases that we could come up with:

ID	Work Item Type	Title
		Functionality Testing
100	Test Case	**Able to login**
101	Test Case	Able to login with Google Account
102	Test Case	**Able to edit account settings**
103	Test Case	Able to edit/save account details
104	Test Case	Able to edit/save payment settings
105	Test Case	Able to edit/save notification settings
106	Test Case	**Able to create a new listing**
107	Test Case	Able to add seller information
108	Test Case	Able to add listing information
109	Test Case	Able to add pictures
110	Test Case	**Able to view all active listings as buyer/seller**
111	Test Case	Able to purchase an item
112	Test Case	Able to save payment information
113	Test Case	Able to search for an item
114	Test Case	Able to rate another user
115	Test Case	Able to view user's rating
116	Test Case	**Able to view my wishlist**
117	Test Case	Able to add an item to my wishlist
		User Testing
118	Test Case	Opinion on app experience
119	Test Case	UI/UX Testing & Feedback
120	Test Case	Was able to complete all funcitonality tests

Table 3

Objectives:

We wanted to accomplish the following with our test strategy:

1. All major features must have a Test Case that covers every step in which the end users may encounter.

2. Make sure that the Test Cases accounted for each of the end user roles which includes the buyer and the seller. Although the user can be a buyer and seller, we wanted to make sure all the features were hit.
3. Any Test Cases that failed during testing we are written down then so we could go back and address the issue.
4. Fix any items that we found as issues and re-test before the Tech Expo.

Logging Test and Procedures:

As stated above, we took two approaches, user testing for each and every feature, and weekly agile testing followed by a meeting to discuss results.

Both approaches presented us with bugs, however, most of the issues came with our agile testing approach and by time we got the user testing we had already flushed out a lot of the issues. Each time we did encounter a problem one of the developers would add it to their to do list and get it fixed.

Any problems that arose during our agile testing would be recorded in discord so that we could all easily see the problem and the appropriate party can claim it and fix the issue. Then for the user testing we made sure we had all our features listed in an Excel sheet so that we could get the most in-depth testing from the users.

After the individual testing throughout the week and logging any issues that we found we would meet on Sundays every weekend to go over those problems and divvy out the tasks to the appropriate team member.

2.8.4 Test Results

Table 4: Test results shows everything from our previous test plan table and includes the results after we did all our testing:

ID	Work Item Type	Title	Pass/Fa
		Functionality Testing	
100	Test Case	**Able to login**	Pass
101	Test Case	Able to login with Google Account	Pass
102	Test Case	**Able to edit account settings**	Pass
103	Test Case	Able to edit/save account details	Pass
104	Test Case	Able to edit/save payment settings	Pass
105	Test Case	Able to edit/save notification settings	Pass
106	Test Case	**Able to create a new listing**	Pass
107	Test Case	Able to add seller information	Pass
108	Test Case	Able to add listing information	Pass
109	Test Case	Able to add pictures	Pass
110	Test Case	**Able to view all active listings as buyer/seller**	Pass
111	Test Case	Able to purchase an item	Pass
112	Test Case	Able to save payment information	Pass
113	Test Case	Able to search for an item	Pass
114	Test Case	Able to rate another user	Pass
115	Test Case	Able to view user's rating	Pass
116	Test Case	**Able to view my wishlist**	Pass
117	Test Case	Able to add an item to my wishlist	Pass
		User Testing	
118	Test Case	Opinion on app experience	Pass
119	Test Case	UI/UX Testing & Feedback	Pass
120	Test Case	Was able to complete all funcitonality tests	Pass

Table 4

2.9 Budget

Table 5: The budget table consists of an item, the original/actual cost, and a final cost of each item. All the items are features from Azure, and while we are using the free version of everything for our project when you look a big business side of what it would cost there is a big difference.

Item	**Original/Actual Cost**	**Final Cost**
App Service	$ 0 per month	$ 297.60 per month
SQL Database	$ ~5.35 per month	$ 138.47 per month
SQL Server	$ 0 per month	$ 0 per month
Application Insights	$ 0 per month	$ 0 per month
Total	$ 5.35 per month	$ 436.07 per month

Table 5

2.10 Problems Encountered

The development of our user interface was slower than anticipated, which led to slower development in the back end as well. This problem was resolved by spending more of the group's time focusing on the development of the user interface. Once the UI was developed, producing a working back-end was more straightforward. Another problem we encountered was setting up Identity Server 4 in our application. There is a ton of documentation out there on how to set up Identity Server, but it's very dense and requires some knowledge of how Identity Server

works. This issue was resolved by meeting with a friend who has experience setting up Identity Server before and was able to help guide us through the process.

The new year brought new problems. At the beginning of January, we updated the app to .NET Core 3.1. We then spent the entire month, plus some of early February, fixing all the problems that this update caused. These issues included Google login no longer working, Identity Server no longer working, and our CSS not being displayed. The final major roadblock was storing and displaying images using Azure Blob Storage. Displaying the images was somewhat simple but storing them turned out to be one of the most complicated parts of our project, and we ended up having to cut images entirely because we were running into a time crunch. Because of all the time we lost updating the app and trying to implement images, we ended up having to cut even more features

2.11 Future Recommendations

2.11.1 Redo

If we could redo the project from the beginning, there are a few things that could have been done better. Development of the front-end portion of the application could have been done earlier in the project lifecycle and could have been accomplished more expediently. This would have allowed more time to develop the core features during the fall semester.

2.11.2 More Time

If we had more time to work on the development of eZsell, we would have seen more of the promised core features come to life. Ideally eZsell would implement strong location-based listing searches and seamless communication in-app but developing the core functionality of the service took up most of our development time.

2.11.3 Suggestions

We have received suggestions regarding the uniqueness of our application. It is similar to other services in that it is an online marketplace, but it serves a specific purpose in providing a solution for used and out of lifecycle hardware. We have also heard feedback regarding the usability of our interface and plan to improve its responsiveness while making the interface intuitive.

2.11.4 Plans

While eZsell does solve an important problem, we do not plan on proceeding with any potential ventures after we have developed the application. It will mainly serve as a proof of concept.

3. APPLICATION PAGES

3.1 Login

Figure 3: The login page is the first page users will come to when visiting our application. Users are prompted to sign in using their google account or local login.

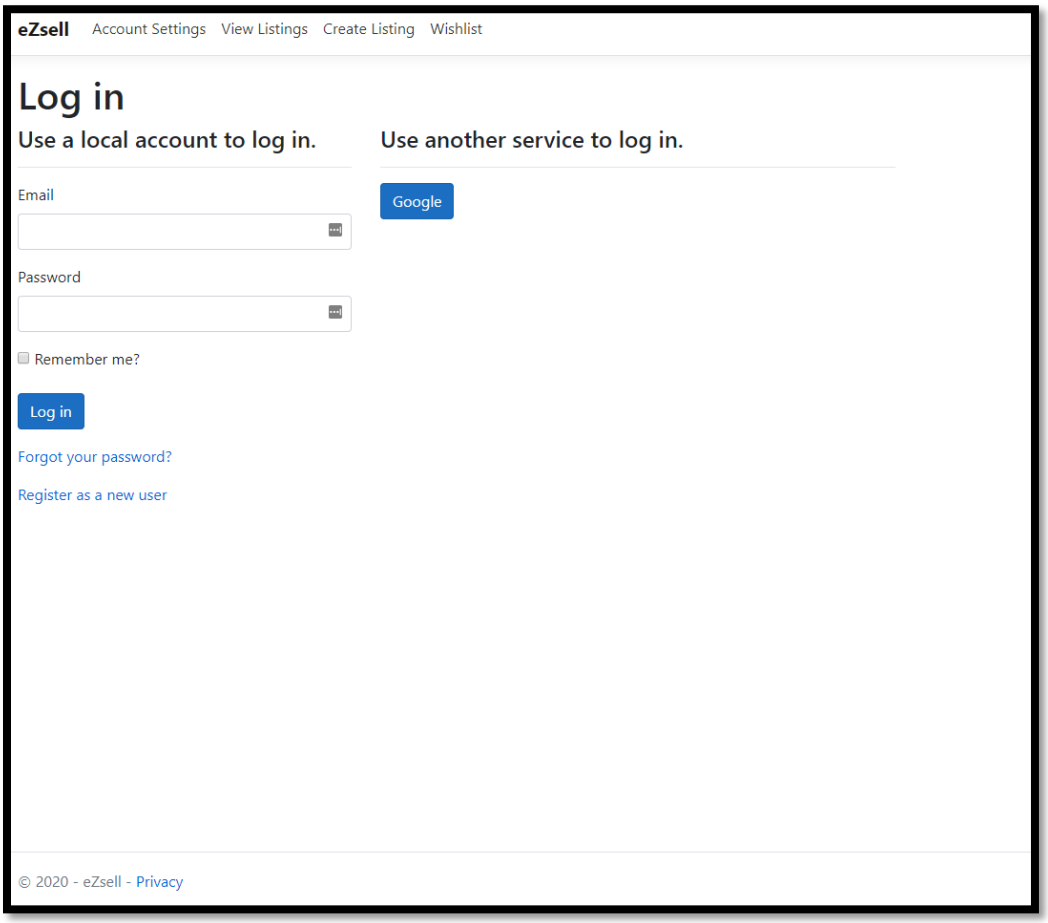

Figure 3

3.2 Account Settings

Figures 4: Account settings are where the users will go to edit any information to do with their account. These include functions such as wish list changes, location preferences, and username. Some of these are subject to change by Spring semester.

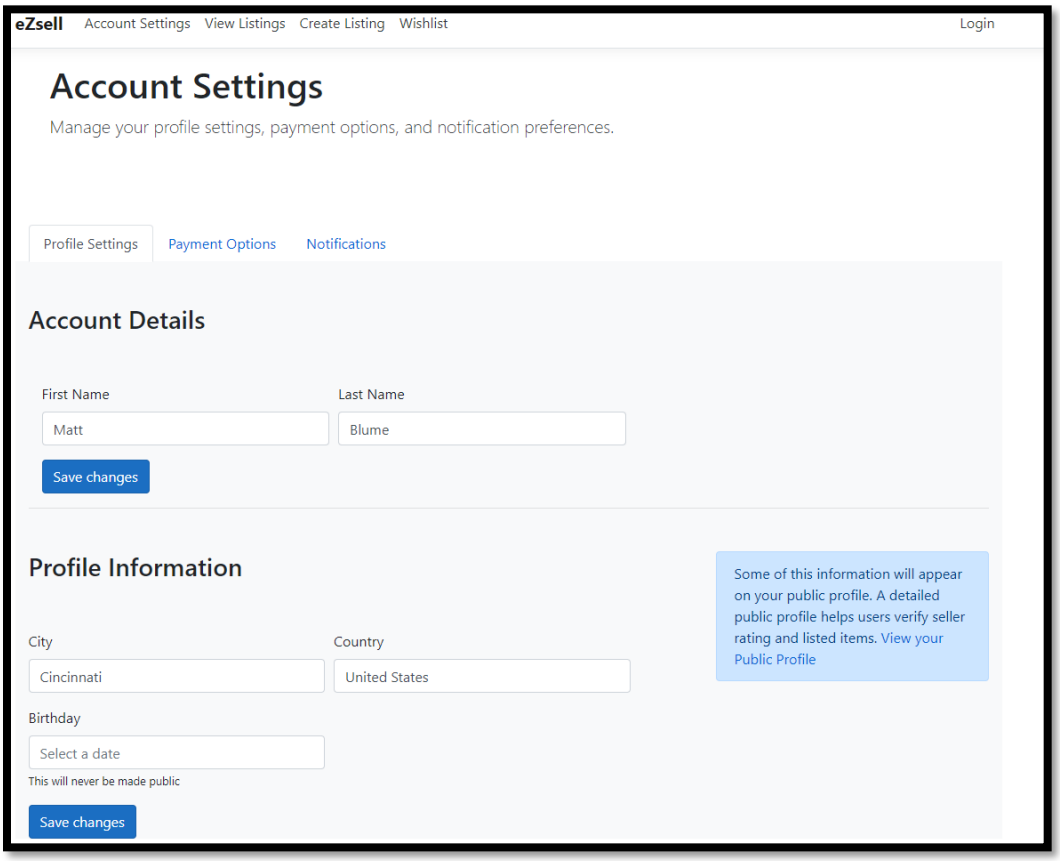

Figure 4

3.3 Listings

Figure 5: The listings page allows the users to view all listings that are currently up on our application. The users can see the item name, category, image, description, creation date, posted by, price and actions.

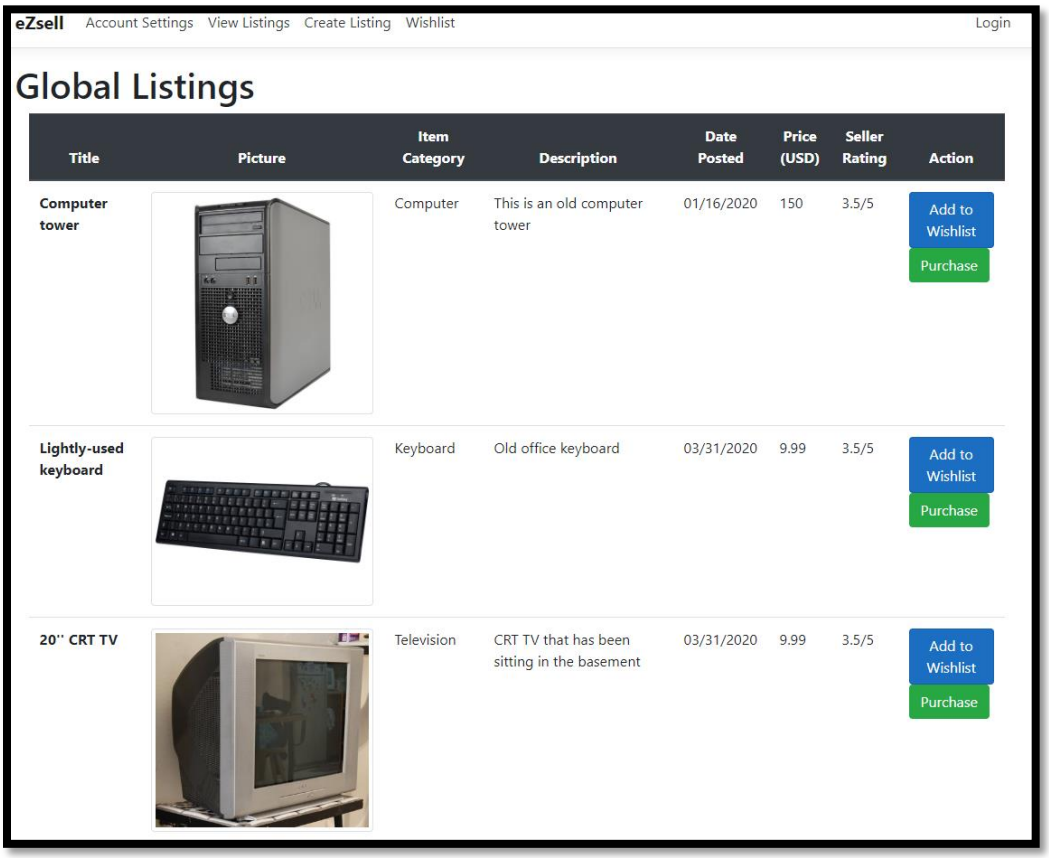

Figure 5

3.4 Create Listing

Figure 6: Create listings allows the sellers to put up their items that they don't need any more up for sale. On this page the seller can specify a title, category, price, picture and item description.

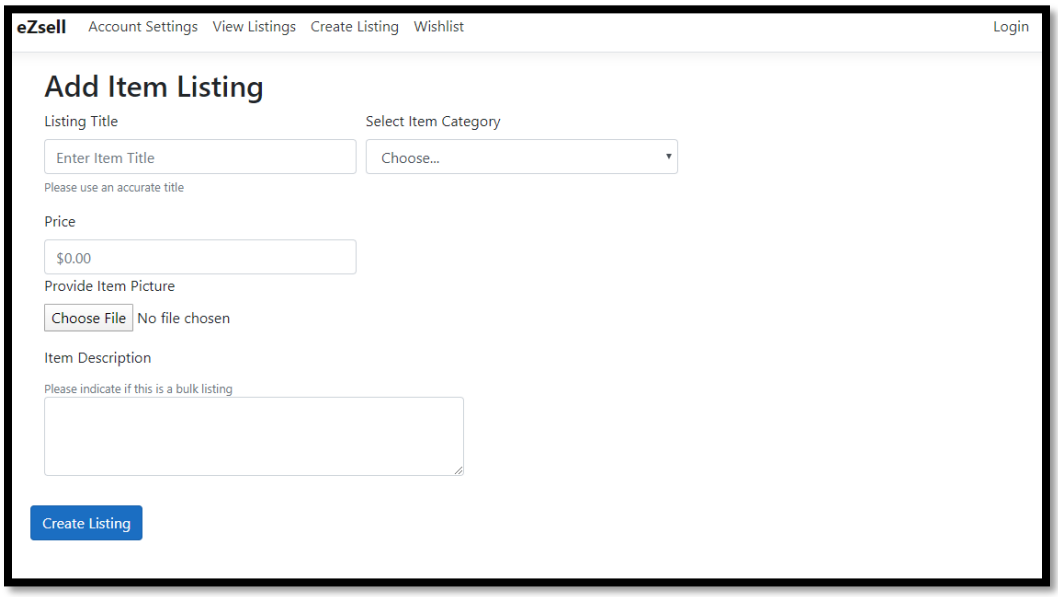

Figure 6

3.5 Wishlist

Figure 7: The Wishlist allows users to add an item from the view listings page to this wish list so that they can come back later and purchase an item.

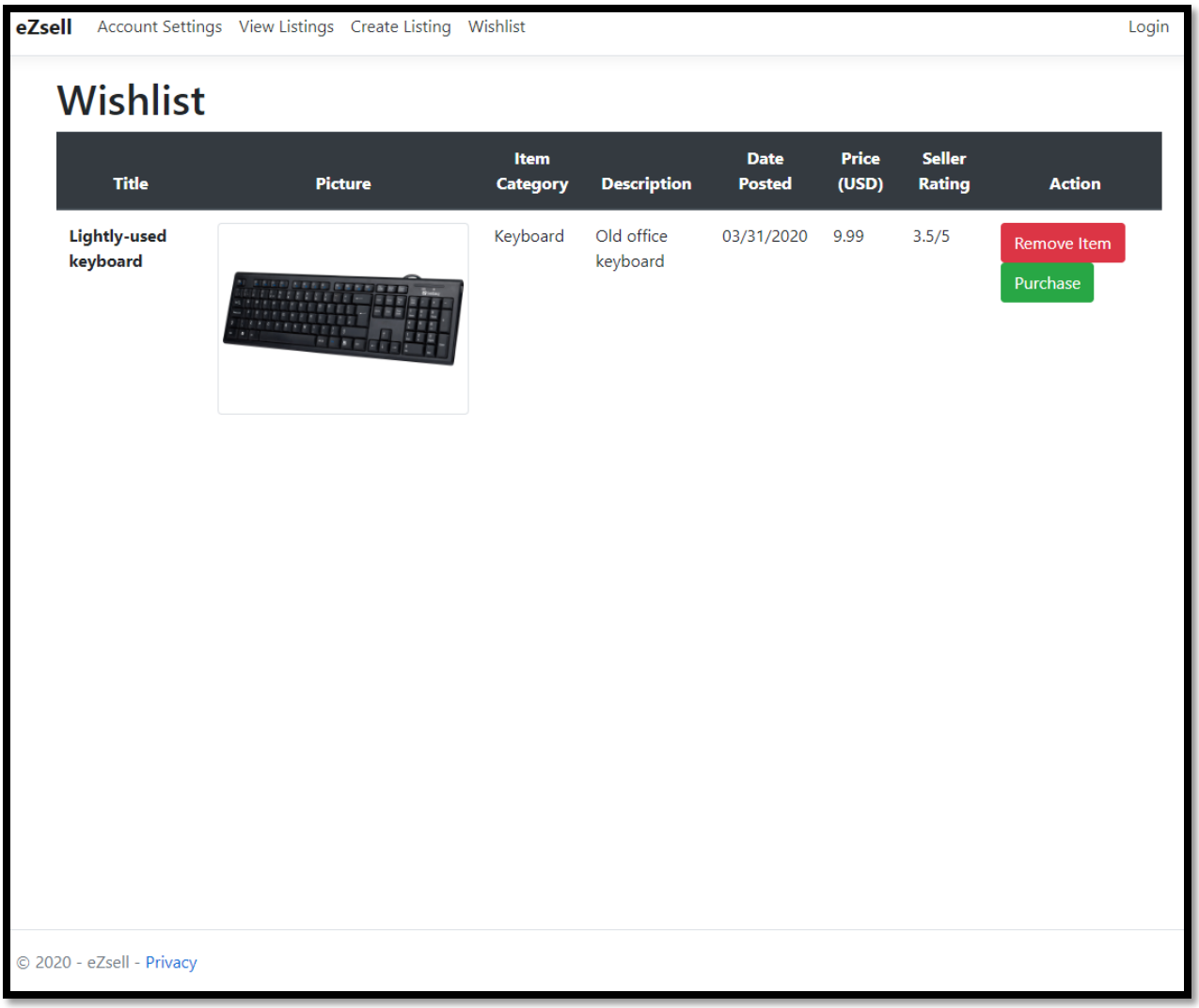

Figure 7

4. CONCLUSION

Fall Semester 2019

Our focus during the Fall semester was on establishing a solid foundation for our project. The biggest issue we have faced in doing this has been getting Identity Server 4 set up and getting Google login working. By tackling this problem, we have enhanced our knowledge of how Identity Server functions and now have a solid foundation to build the rest of our project from. In the Spring, we planned on being able to focus on fleshing out the core features of our app beyond the proof of concept stage they are in now.

Spring Semester 2020

Spring semester 2020 was very interesting. In December of 2019 Microsoft released .NET Core 3.1. We spent the beginning of 2020 updating our app to Core 3.1 and fixing all the build errors this caused. When we thought those were fixed, we launched the application, only to discover that some major features of our app no longer worked. We spent the next month trying to fix various problems, such as issues with Google login and our CSS not being displayed. When we finally had everything working, we moved onto our next two major tasks, overhauling the UI and uploading/displaying images from Azure Blob Storage. The UI overhaul went smoothly, but getting images working proved to be very complicated, and ultimately had to be abandoned for times-sake. We now found ourselves with a lot of work and not very much time left to do it, which led to more features being cut. We experienced some crunch leading up to our final

presentation, but ultimately, we were able to deliver a production ready product that had most of the features we had originally promised.

4. BACK MATTER

4.1 References:

[1] “Average lifespan (replacement cycle length) of consumer desktop PCs in the United States from 2018 to 2023 (in years)” 2019 [Online]. Available: https://www.statista.com/statistics/267465/average-desktop-pc-lifespan/

[2] “Review of Replacement Schedules for Information Technology Equipment” 2013 [Online]. Available: https://www.lbb.texas.gov/Documents/Publications/Issue_Briefs/257_IT%20Replacement%20Schedule.pdf

[3] “The Hidden Costs Of Your Aging IT Infrastructure” 2017 [Online]. Available: https://blog.juriba.com/the-hidden-costs-of-an-aging-it-infrastructure

[4] “Introduction to Identity on ASP.NET CORE” 2019 [Online]. Available: https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-3.0&tabs=visual-studio

[5] “Overview-IdentityServer4 Documentation” 2019 [Online]. Available: http://docs.identityserver.io/en/latest/quickstarts/0_overview.html