

Final Project Paper

Team Triple-P: Alex Smith, Herb Glaser, Kaitlyn Dominguez, Jill Oestreicher, Jake Burke

Computer Science / ITS Capstone Project

Table of Contents

Summary of Relevant Facts	2
Software Project Analysis	3
Specification of User Requirements	3
Use Case Diagram and Documentation	5
Activity Diagrams and Documentation	6
Sequence Diagrams and Documentation	7
Software Project Design	8
Database Design, Dodel, and Documentation	8
User Interface/Navigation/HCI/Design Principles	12
Infrastructure Specifications	18
Prototype Deployment Instructions	22
Cost Analysis	30
Project Plan	31
Ethics	38
Appendices	41
References	42

Summary of Relevant Facts

The capstone project for students in the School of Computer Science at Marist College is an immersive, culminating experience for students to combine and showcase the skills they have learned over the course of their undergraduate education. As a result, our team, consisting of Alex Smith (Computer Science), Herbert Glaser (Computer Science), Kaitlyn Dominguez (Computer Science), Jill Oestreicher (Information Systems), and Jake Burke (Information Technology), has spent all semester working under the advisement of Dr. Pablo Rivas to create and deploy our capstone project to the best of our abilities. This experience has not only allowed us to demonstrate what we have learned, but to continue developing our skills by participating in a real world project that models what most of us will be involved in as part of the workforce after graduation.

Our project was to create an e-commerce site for a small startup based in Arizona that sells used truck parts and lists used trucks and trailers to receive a commission. This company, owned by Jaime Higuera, is called Arizona Custom Services and Sales LLC. While a fairly new business, ACSAS already had established a few websites before our project. Mr. Higuera requested for us to create a new site that he can use as the sole, centralized site for his business. This website needed to be able to accept credit card payments from customers and place orders when the transactions are verified. The site also needed to provide sales reporting, user analytics, and inventory maintenance for administrative control. We believe we have properly fulfilled Mr. Higuera's requests in creating this site by working hard to gather requirements, break up tasks, and create the necessary components. This paper illustrates the process we took as a team in completing this project, along with all of our documentation and instructions in order to hand it over to our client.

Software Project Analysis

Specification of User Requirements

This documentation includes all of our client requirements based on client interviews, messages, and other communications.

Customers can:

- Manage account
 - Create or delete account
 - Save payment information
 - Can sign up for email promotions
 - View past / current orders and their status
- View Items
 - Look at item listing, view item information and price
- Search Items
 - Filter to find the items they want
- Add to cart
 - View Cart
 - Review which items they would like to purchase
 - Place Order
 - Securely pay with credit card information
 - Checkout with account or guest checkout (no account required)
- Find company contact information
 - Contact info page
 - Chat feature: will investigate the ability to incorporate into site
- Manage Their Own Items on Truck/Trailer Page
 - Post Items

Admin/Company can:

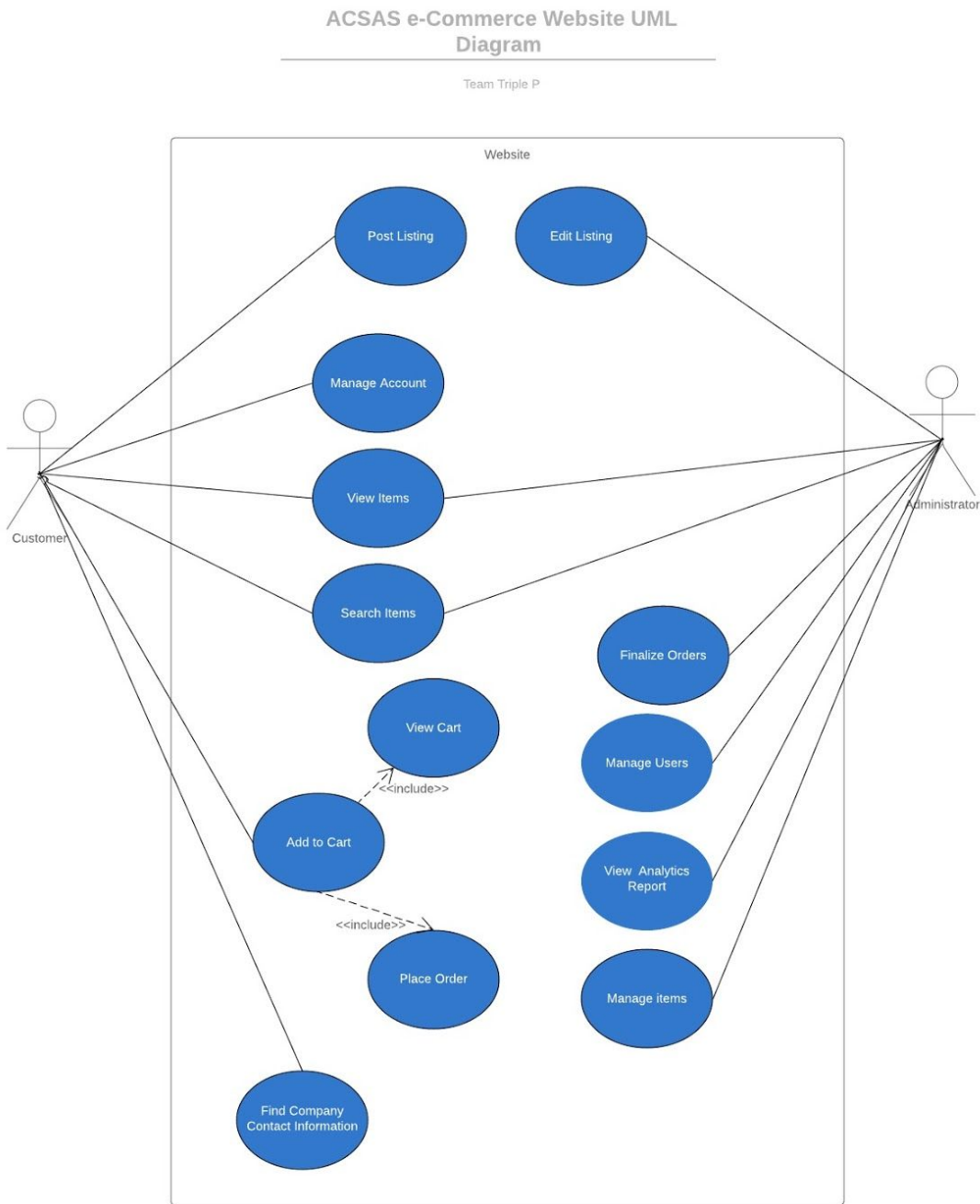
- Manage Items
 - Post Items
 - Remove Items

- Edit Item Description
 - Availability
 - Picture of item
 - Price of items
 - Brief description
- View Items
 - Look at item listing, view item information and price
- Search Items
 - Filter to find the items they want
- Finalize Orders
 - Review orders, confirm if the payment was valid, prepare item for shipping
- Manage Users
 - Review users orders, send them email promotions, etc.
- View Analytics Report
 - Visitor count, time, day, etc.
 - This will determine the best time to do a system update if necessary
 - Sales report
 - Weekly, monthly, orderly

Other general requirements:

- Parts, trucks, and trailers all have separate parts of the website
- Reuse logo

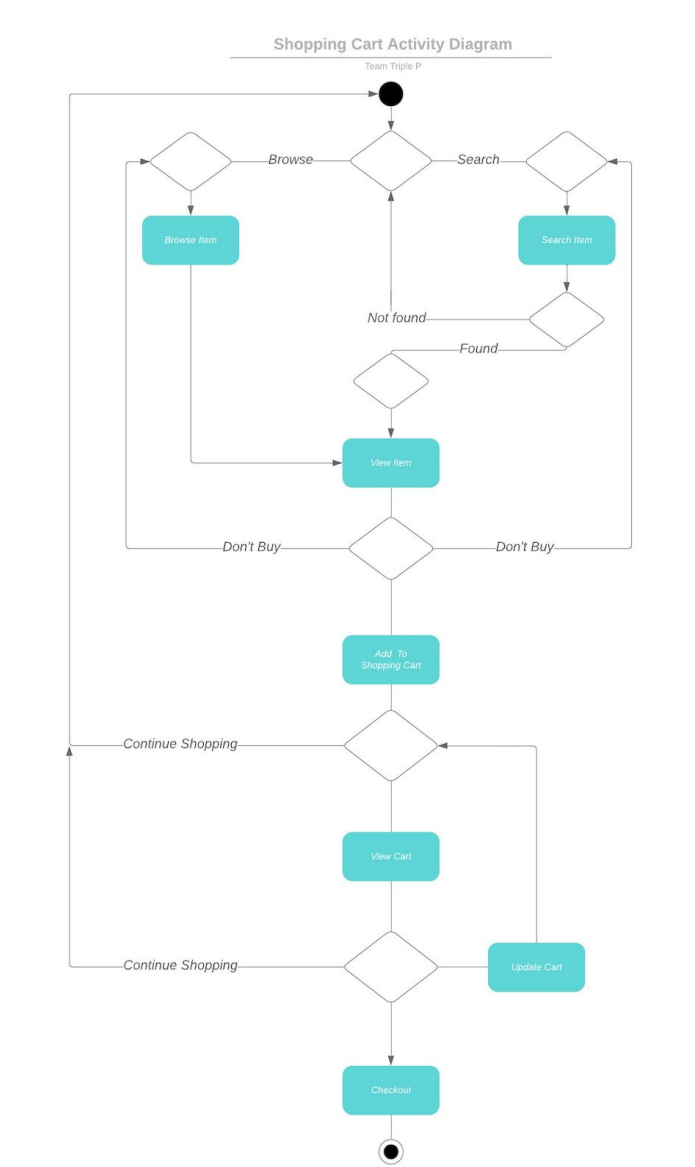
Use Case Diagram and Documentation



A customer can view and search for items, as well as add items to their cart, view their cart, and place an order as a guest without an account. They can also find company contact information. They can make an account and manage said account, as well as post and edit listings with that account.

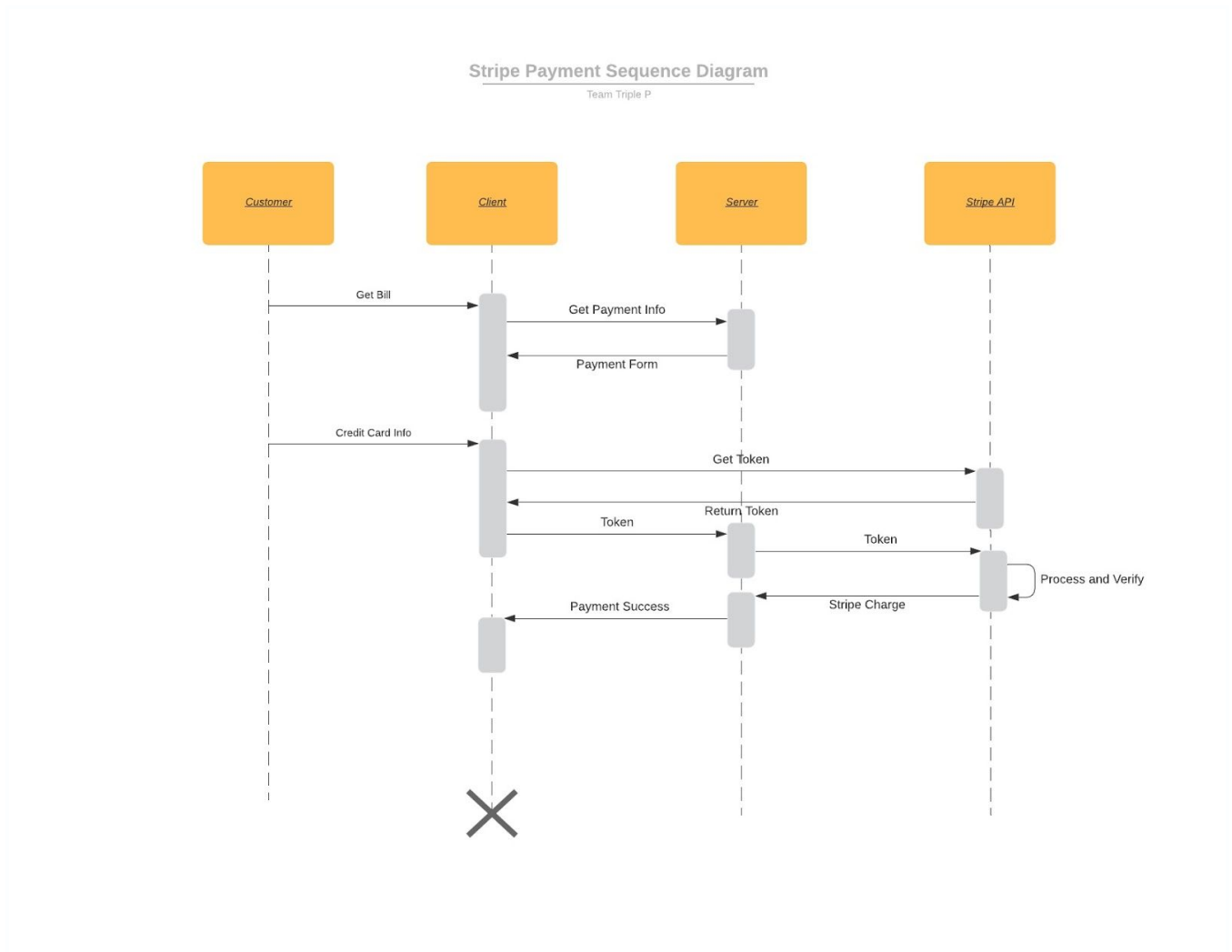
An admin can also view and search for items. They also have the special ability to finalize orders, manage users, view the Google analytics report, and manage product items.

Activity Diagram and Documentation



A customer wants to shop for a product. They can either browse through all products, or search for a specific product. If they search for an item and can't find it, they can try to search again or browse through all of them. If they find a product through searching or browsing, they can decide to buy it and add it to their shopping cart, or they can browse or search for another item again. If added to their shopping cart, they can continue shopping or decide to view their cart. After viewing, they can either continue shopping, update their cart, or checkout.

Sequence Diagram and Documentation

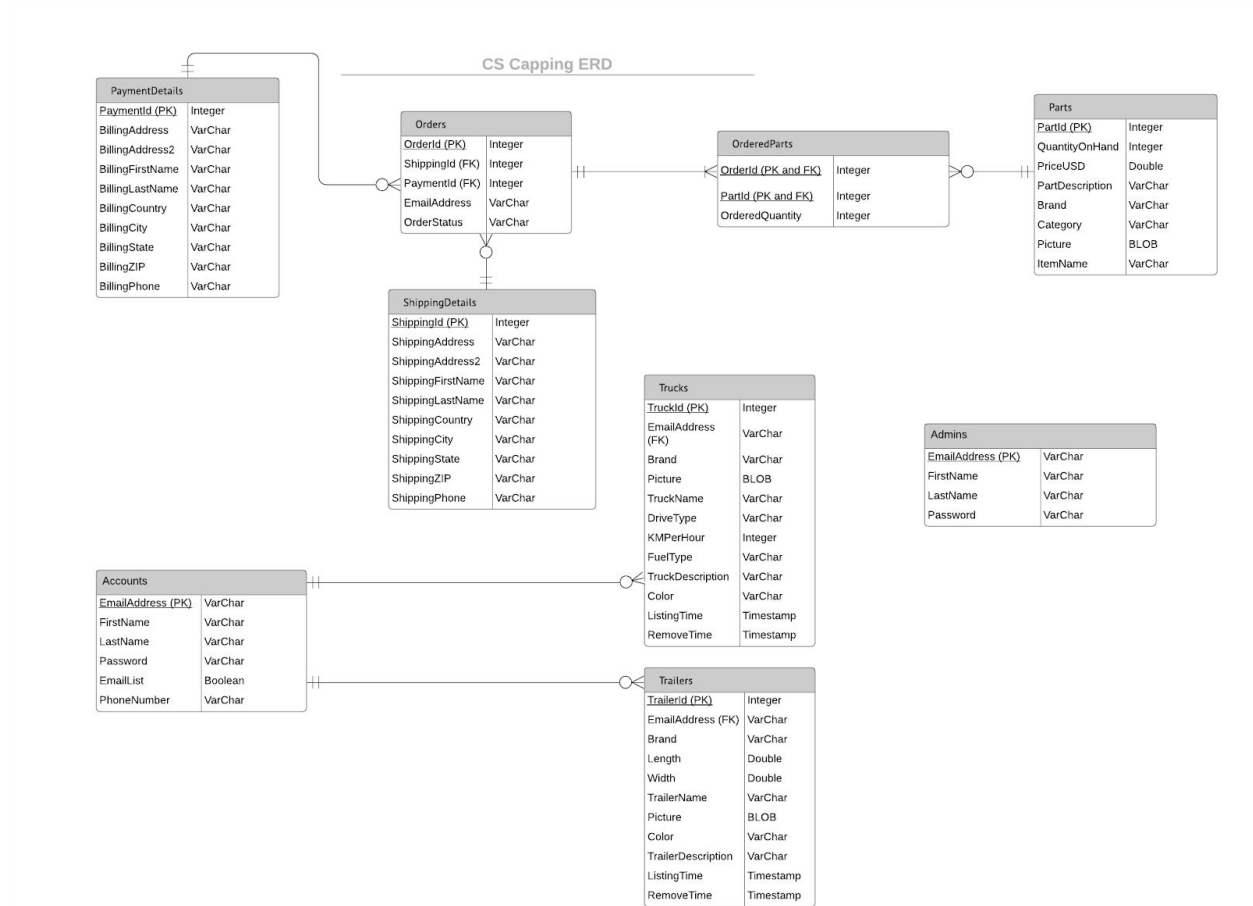


The user generates a bill after purchasing a product or listing. This bill is sent to the client who requests payment information from the server which sends a form back. The customer sends over their credit card information through the client's system which goes straight to the Stripe API as a token and is returned. This is reviewed by the client and is returned to the

server, and then back to the Stripe API to process and verify for accuracy. Then Stripe sends the final charge to the server who confirms payment success to the client.

Software Project Design

Database Design, Model, and Documentation



Accounts

Primary Key: Email Address

Columns:

- FirstName
- LastName
- Password
- EmailList: This value indicates if the account is signed up to receive promotional emails.

- PhoneNumber

Admins

Primary Key: Email Address

Columns:

- FirstName
- LastName
- Password

PaymentDetails

Primary Key: PaymentId

Columns:

- BillingAddress
- BillingAddress2: Optional second address line, if needed.
- BillingFirstName
- BillingLastName
- BillingCountry
- BillingCity
- BillingState
- BillingZIP
- BillingPhone

ShippingDetails

Primary Key: ShippingId

Columns:

- ShippingAddress
- ShippingAddress2
- ShippingFirstName
- ShippingLastName
- ShippingCountry
- ShippingCity
- ShippingState

- ShippingZIP
- ShippingPhone

Orders

Primary Key: OrderId

Columns:

- EmailAddress: If the user is signed into an account, it will use their account email address. If the user is using guest checkout, it will ask for an email address to fill this field.
- OrderStatus: Will tell if the order is being processed, confirmed, in transit, or delivered.
- ShippingId (FK): Orders have 1 and only 1 set of shipping information. If the customer is using an account and the account has shipping details linked, those shipping details will be used.
- PaymentId (FK): Orders have 1 and only 1 set of payment information. If the customer is using an account and the account has payment details linked, those payment details will be used.

Parts

Primary Key: PartId

Columns:

- QuantityOnHand: How many of this part is currently in stock.
- PriceUSD
- PartDescription
- Brand: What brand the product belongs to.
- Category: What area of the truck does this part corresponds to.
- Picture: Picture of the part.
- ItemName

Trucks

Primary Key: TruckId

Columns:

- Brand

- Picture
- TruckName
- DriveType: Four wheel drive, rear wheel drive, etc.
- KMperHour
- FuelType
- TruckDescription
- Color
- ListingTime: Time customer lists item
- RemoveTime: Time customer removes item
- EmailAddress (FK to Accounts): A truck has 1 and only 1 contact associated with the listing.

Trailers

Primary Key: TrailerId

Columns:

- Brand
- Length
- Width
- TrailerName
- Picture
- Color
- TrailerDescription
- ListingTime
- RemoveTime
- EmailAddress (FK to Accounts): A trailer has 1 and only 1 contact associated with the listing.

OrderedParts

Primary Key: OrderId and PartId (both are foreign key to their respective tables)

Columns:

- OrderedQuantity

- OrderId (FK to Orders): The order that given amount of parts is associated with
- PartId (FK to Parts): The part being ordered.

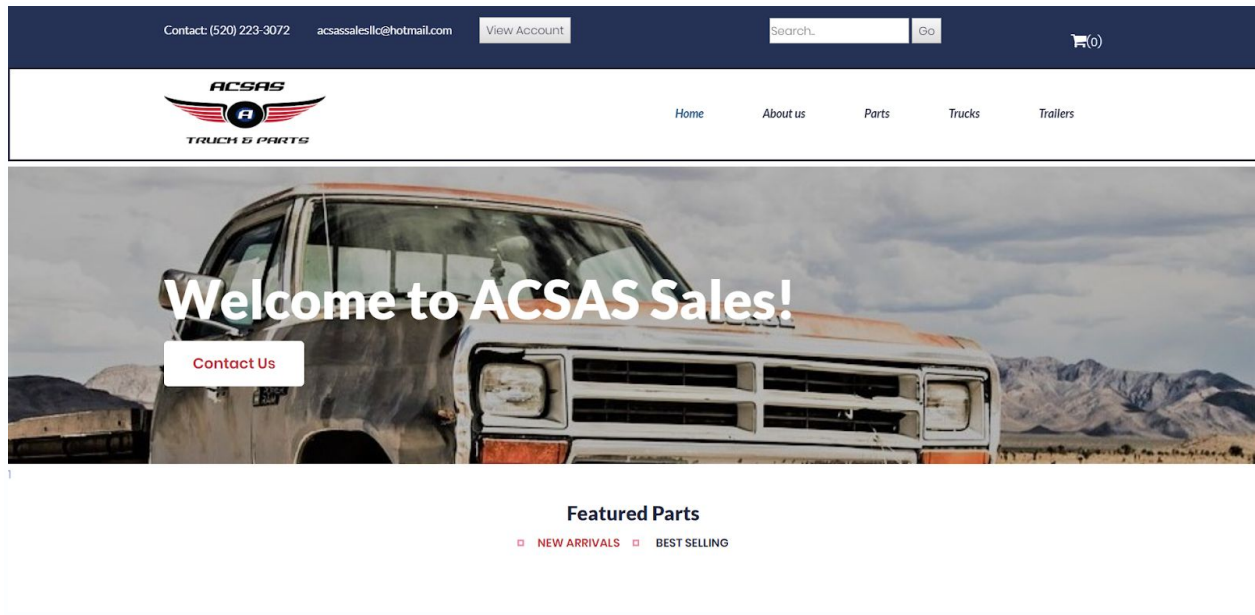
User Interface/Navigation/HCI/Design Principles

We designed the user interface with the intention to be aesthetically pleasing as well as user friendly for our customers. We also designed it in the attempt to stick to a theme that we felt best represented ACSAS as a whole. We utilized freethemescloud.com, which provides open source frameworks for building sites. We found a basic design that we could manipulate, and it required us to put their site link in the footer of ACSAS' site to comply with their policy.

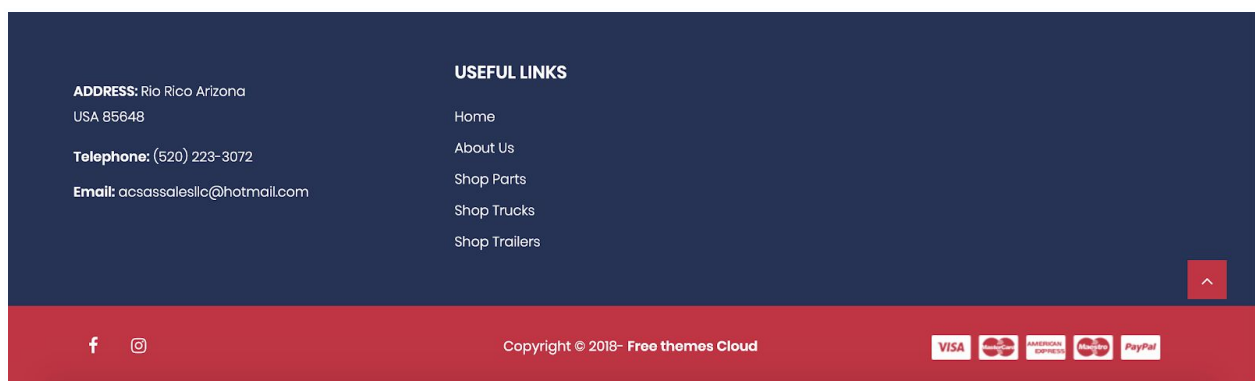
We based our site colors off of ACSAS' logo which is the following, as our client requested we incorporate it into the site:



The similar colors of the logo can be seen on all our pages, such as in our homepage as seen below:



We utilized the same header on all of our pages, that provides contact information in order to make it easier for customers to reach out to ACSAS if necessary. We also provided access to the search bar and cart at the top because these features are commonly located there, making it more intuitive for a new user to utilize this site. Then we provide all the links to the other pages of the site, again for easy access for the user. We also provide them in the footer of every page should they have scrolled down to look at something further in the page:



Having uniformity in the layout of each page is important for users to get used to finding important links and features and thus for overall user retention.

The homepage provides a brief overview and link to every main function of the site, such as viewing parts and listed trucks and trailers, or access to your account. For example, below shows access to seeing newly added parts or best selling parts from the homepage:

Featured Parts
NEW ARRIVALS BEST SELLING

			
3090942 Cummins M11 / N14 / ISM Fuel Pump \$398	Automatic Slack Adjuster, 28 Spline, x 5.5, Replaces Haldex 40010211 H-26411 \$65	Crankcase Oil Pan Pressure Sensor For Volvo D12 D13 VN VNL VHD Engine 630 670 \$48	4921517 Engine Oil Pressure Sensor With Electrical Connector For Cummins ISX ISM \$58

Each image can be clicked to find more information, and hovering over it highlights it to indicate to the user that it is clickable, as this function is common in websites. We also provide an arrow to automatically return the user to the top of the site, which is found on each page and can be seen in the image above.

The other main pages of our site include the parts, trucks, and trailers pages. In keeping with the idea of uniformity within the aesthetics of our site, when viewing parts, trucks, and trailers we used a similar format:


Search..

Go

Sort By:

Name

Sort




Auto Slack Brand: HD Value Part#: HDV-10143

HD Value Automatic Slack Adjuster, Haldex Style, 5-1/2 or 6-1/2 Arm Length with 10 Splines and a 1-1/2 Spline DiameterCross Reference Information40010143, ASI032, ASI034, OTR0143

\$72

Add to cart



Air Dryer Brand: Power Products Part#: 955205P

P WABCO Monitor Style System Purge Air Dryer - 12V/Cross Reference Information 170955205, 955205P, 955205X, 955300X, D6336, OTR955205, R955205, SI3728, S4324130010

\$103

Add to cart


Search..

Go

Sort By:

Name

Sort




Dodge Truck

Brand: Dodge **Fuel Type:** Diesel **Drive:** Four Wheel

My Dodge Truck

herb.glaser1@marist.edu



New truck

Brand: Nisaan **Fuel Type:** Diesel **Drive:** 4 Wheel

My truck good condition

herb.glaser1@marist.edu


Search..

Go

Sort By:

Name

Sort



My Old Trailer

Brand: Ford **Length:** 12 **Width:** 32

Good Condition

herb.glaser1@marist.edu

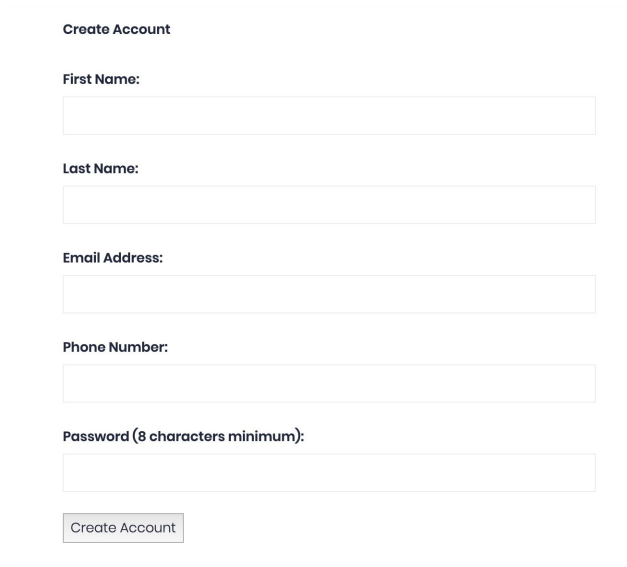
<

1

>

Each item provides a brief overview but can be clicked to be brought to a page with more information. A customer can also add an item to their cart from both the overall parts page and the individual part details page, which makes it easier for the customer as having to perform less clicks is always optional.

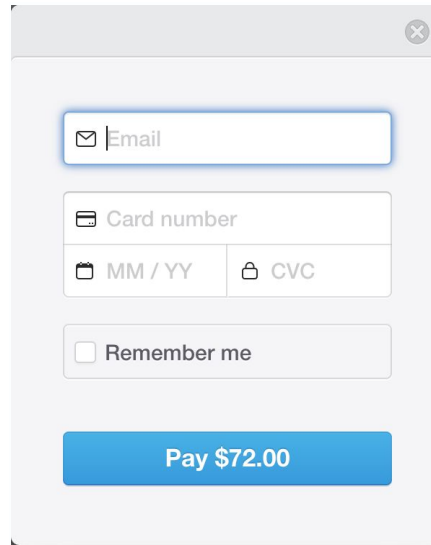
We also kept the way fields to be filled out looked the same. Making an account, listing a truck or trailer, billing and shipping, and more require users to have to fill out information on the site. To make it simple, we provide clear field boxes, as shown below in the create account page:



The image shows a 'Create Account' form with the following fields and labels:

- Create Account** (Section Header)
- First Name:** (Text input field)
- Last Name:** (Text input field)
- Email Address:** (Text input field)
- Phone Number:** (Text input field)
- Password (8 characters minimum):** (Text input field)
- Create Account** (Submit button)

As this page shows, the fields are very straightforward and the text boxes very obviously are matched with the text directly above them to indicate to the user what to fill out. The only fields on our site that look different are the payment fields due to the fact that we utilize Stripe, a third-party system:

A payment form modal with a close button in the top right corner. It contains an email input field with an envelope icon, a card number input field with a card icon, and two input fields for MM / YY and CVC with calendar and lock icons respectively. Below these is a checkbox labeled "Remember me". At the bottom is a blue button labeled "Pay \$72.00".

Email

Card number

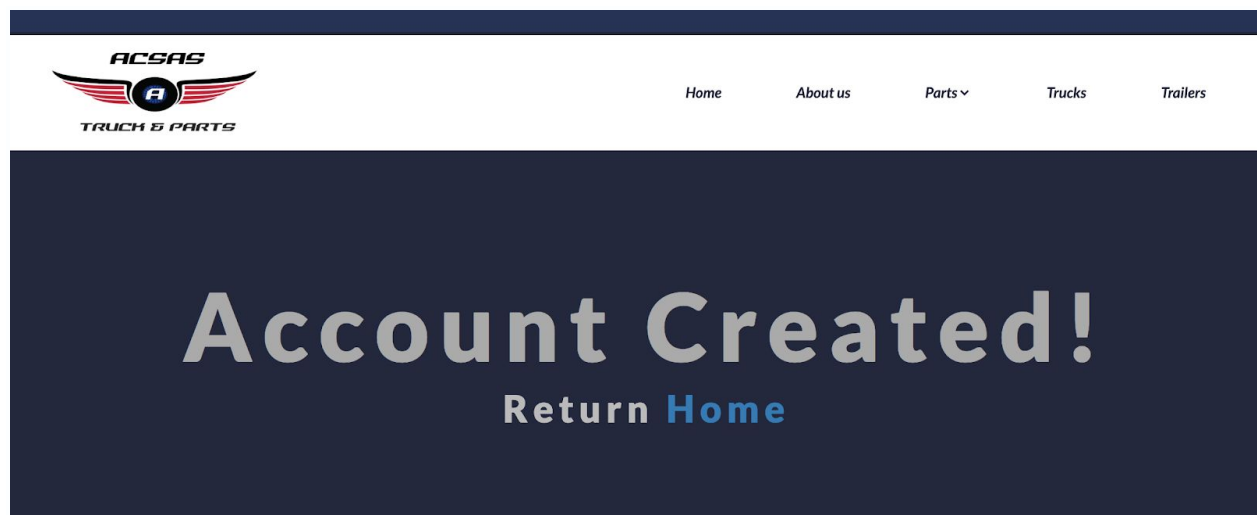
MM / YY CVC

☐ Remember me

Pay \$72.00

These fields are also very clearly marked and is overall simple for the user to fill out.

Another important feature of our user interface is having confirmation pages for different actions such as confirmation for the creation of an account:



This ensures that the action the user took was carried out successfully, so there is little room for confusion.

In general, we wanted the site to provide a clear, responsive, and efficient experience for ACSAS's users. We believe we were able to do this by mimicking the functionality of existing

e-commerce sites we were familiar with, as well as by fulfilling what was requested by the client. After all the testing conducted by our team as well as external users, we feel we have established good UI/UX principles.

Infrastructure Specifications

Creating the Servers:

To create a VM, we went to capping.ecrl.marist.edu and logged in with the credentials we were given. The vmware vSphere Web Client website allowed us to create our 2 servers. One server is used for our database while the other is used to host our site.

When creating both servers the steps were the same:

1. Right Click on Capping Cluster and create a new virtual machine.
2. Create the Virtual machine, give it a name, and select the location of creation you want.
3. Select a compute resource to run the server on. (Cluster 3)
4. Select the storage that you would like to use. (Current Capping)
5. Select the compatibility. (ESXi 6.0 and Later)
6. Select an OS. (Microsoft Windows Server 2016 (64-bit))
7. Customize hardware - for this step we chose all the default settings, and only changed a few things.
 - a. 2 CPU with 2 Cores Per Socket
 - b. 4 GB of memory
 - c. 60 GB of Hard Drive
 - d. Change the Disk Provisioning to Thin Provisioning
 - e. Add a USB Controller 2.0 to allow mouse use
 - f. Lastly Select the Data store Windows 2016 ISO File
8. Once these steps were complete, we created our server.

Setting up OS and IP:

Now that we have correctly set up our server, we can power it on and begin to set up the OS and connect an IP address.

1. Power on the server.
2. Select the default options.
3. Select Windows Server 2016 Standard Desktop Experience for Windows with GUI
4. Select Custom download and click on the Hard Drive
5. Windows should now Download.
6. Create a Password
7. Set up the IP
 - a. Go to network Connections
 - b. Right click on Ethernet0 and click properties
 - c. Select IP version and click properties
8. For our servers we used:

IP Address	10.10.9.150 or 151
Subnet Mask	255.255.255.0
Default Gateway	10.10.9.1
Preferred DNS Server	10.11.12.2
Alternate DNS Server	10.11.12.3

9. We changed the settings to allow remote desktop

From here on out we used remote desktop to log into our server.

Database Server Info:

IP Address	10.10.9.150
------------	-------------

Subnet Mask	255.255.255.0
Default Gateway	10.10.9.1
Preferred DNS Server	10.11.12.2
Alternate DNS Server	10.11.12.3

- Downloaded MySQL to create Database.
- Downloaded Node.js

Web Server Info:

IP Address	10.10.9.151
Subnet Mask	255.255.255.0
Default Gateway	10.10.9.1
Preferred DNS Server	10.11.12.2
Alternate DNS Server	10.11.12.3

- Added Web Server IIS Role with IIS Hostable Web Core
- Added .NetFramework 3.5 and 4.6 Features
- Added FTP Server Service

After our first two initial servers were up we realized we needed to switch to Ubuntu.

Ubuntu Test Server:

Followed the same steps as previous servers but instead used:

- Linux 18.04 Operating System
- 25 Gigs of Hard Drive
- Used 10.10.9.152 as IP address

We Installed:

- Apache to run the website
- Node.js
- Screen to keep node running.
- MySQL to run the database

Deployment:

We deployed our website onto Google Cloud Services. To create this server we:

1. First logged into google cloud services and Clicked on Compute engine.
2. We then created a new VM instance with the following info:
 - Name - web-server-1
 - Machine Type - f1 micro (1 vCPU, 0.6 GB memory)
 - Zone - us-west2-a
 - We allowed HTTP and HTTPS traffic
 - Image - Ubuntu 18.04 LTS
 - Size - 17GB
 - Also created an SSH key to be able to login
3. On the server we installed
 - Apache
 - Node.js
 - MySQL
 - Screen
4. We last used Certbot to get our HTTPS certification on our website

Reliability:

We expect the site to always be up unless for scheduled maintenance. Response time should be as fast as possible in working with the system.

Recoverability:

The administrator will have access to all backups. We believe that it is best to hold onto as much data as possible. We suggest backups to be performed monthly to the cloud.

Prototype Deployment Instructions

Important Links:

GitHub Repository: <https://github.com/jilloestreicher/Triple-P>

Site Link: <https://acsastruckparts.com/>, domain currently = 34.94.61.58

Stripe Dashboard: <https://dashboard.stripe.com/dashboard>

Gmail: acsastruckparts@gmail.com

Google Cloud: acsastruckparts@gmail.com (Google accounts linked)

Google Analytics:

<https://analytics.google.com/analytics/web/?authuser=2#/report-home/a152418607w215453159p206020527> (Google accounts linked)

GitHub:

All our code used for the deployment of this project can be found in our GitHub repository at <https://github.com/jilloestreicher/Triple-P>. To clone a repository in order to create a local copy on your computer for syncing between the two locations, you can follow the steps here:

<https://help.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository>. It should be noted that this GitHub also contains all assignments required for this capping class.

Site Overview:

The site is located at the following domain: <https://acsastruckparts.com>. Utilizing the site was created to be simple, as is explained in more detail in our project paper, specifically the section about user interface design. In order to utilize this site as an admin, we give details below.

Site Pages:

html:

about-us: General information about ACSAS business

account-created: Confirms new account has been created after user makes new one

adminHome: Home page for admin users

adminlogin: Login for admin users

checkout: Checkout page for customers to purchase items, collects billing and shipping

create-account: Allows users to create site account

edit-order-admin: Admin page for editing customer orders

editItem: Admin page to edit items listed on site

EditListing: Page to edit customer listings

error-500: Page for if user encounters a 500 error (general HTTP status code that means something has gone wrong on the web site's server, but the server could not be more specific on what the exact problem is)

list-success: Confirms user successfully posted a listing

listing-details: Gives more details regarding a listing

login: Login page for users

Logout: Page if user successfully logs out of account

postItem: Admin page to post new items to site

postTrailer: Page for users to post trailer

ejs:

adminOrders: Admin page to view all orders

adminParts: Admin page to view all parts

adminTrailers: Admin page to view all listed trailers

adminTrucks: Admin page to view all listed trucks

cart: Page for users to view and edit the items in their cart

checkout: Page for users to purchase items in their carts, collects payment

editListing: Page for users to edit a truck they've posted

editPart: Admin page to edit parts

editTrailer: Page for users to edit a trailer they've posted

index: Homepage for all users, gives general site overview and links to pages

listing-details: Details about listed trucks

manage-users: Admin page for managing user accounts

my-account: Page for user to view and edit accounts

order-history: Displays past orders for users

order-template: Template for filling out order information

postlisting: Page for user to post truck listing

posttrailer: Page user to post trailer listing

shop-details: Gives details about parts available for purchase

shop: Lists all parts available for purchase

trailer-details: Details about listed trailers

trailers: Lists all listed trailers

trucks: Lists all listed trucks

Admin Login:

Admins do not use the same login page as the normal users. To get to the admin-specific login page, you must use this link: <https://acsatruckparts.com/Front%20End/adminlogin.html>.

Admin Home Page:

After the admin has logged in, they will be redirected to the admin-specific home page. This page lists all the links to the functions the admin can do. To reiterate, these pages can only be accessed by the admin and not by any of the users. Admins have the ability to edit and delete parts, truck listings, and trailer listings and manage both users and orders. All of these operations are linked on the admin home page.

Add Parts:

This link will bring you to the Post Parts page. This allows the admin to add parts to the site based on the inventory. Once the submit button is hit, the part will be added to the Parts page on the site.

Edit and Delete Parts:

This link will bring you to a page that looks very similar to “Parts”. The admin can search for different parts or sort all the parts by name, newest, or price. Admins have the ability to edit or remove parts based off of their inventory. To edit the part, press the “Edit” button. This allows any field in the item description to be changed. To remove the part, press the “Delete” button.

Edit and Delete Truck Listings:

This link is very similar to the Edit and Delete Parts page, except now it applies to all the truck listings. This page shows all the trucks currently listed on the site with both sort and search functionality. Admins have the ability to edit and delete each listing posted to the site. Since these listings are posted on the website by users, the admin should review this regularly to make sure each listing posted is legitimate.

Edit and Delete Trailer Listings:

This link has the same functionality as the Edit and Delete Truck Listings page, except it now applies to all the trailer listings on the site. Admins have the ability to edit or remove each listing. This also needs to be reviewed regularly to make sure each listing is legitimate.

Manage Users:

This page allows the admin to delete users from the system. The page lists all the users currently registered in the system along with a delete button for each one. By pressing the button, that user will be removed from the system and will no longer be able to login.

Manage Orders:

This link brings you to a “View Orders” page. This lists all the orders in the system.

Listing Process:

Customers have the ability to list their own trucks and trailers on the site, paying a commission in return to ACSAS. To do so, there are separate listing pages for trucks and trailers. Both require the user to make an account, or to sign in if they already have one. Then the user fills in the fields that provide details regarding their listing. For the listing be posted, the user then has to pay the initial fee, which is \$5. They will be continuously charged \$5 a day for however many days they leave it up. To change this price, log into the server (as explained below), and go to items.json. Where it says merch, you can change the price where it says 500 (it is in pennies, so 500 = \$5, 400 would = \$4, and so on). Customers need to upload JPGs for photos.

Stripe:

The Stripe account is located at <https://dashboard.stripe.com/dashboard>, with the account Acsastruckparts. This dashboard provides all transaction history and the total current balance of your account as Stripe handles all payments on the site. Stripe has lots of documentation regarding site features located here: <https://stripe.com/docs>. On the Stripe page there is a section called Payments which shows each payment made and how much each order was. If you click a payment it will take you to a page that shows all the payment details including the Stripe processing fee. If someone cancels an order you can refund them by clicking on the [...] and then click refund payment.

Database:

We used MySQL for the database component of the site.

Server:

We used Google Cloud to host our site, the console of which can be found here using the gmail we created:

https://console.cloud.google.com/home/dashboard?project=acsas-website&authuser=2&_ga=2.3289712.-1973809523.1571771916&_gac=1.112813558.1575262778.CjwKCAiA5o3vBRBUEiwA9PVzajRfgrCG1XdRws7SwAktlC5cfEp0D0IbIs9TJGPpA3KTOOqQ9D34GBoc7kgQAvD_BwE

We recommend to add additional storage in the future, and to upgrade your server refer to the following steps:

1. First Log into the Google Cloud platform and Click on compute engine.
2. At the top of the screen google will suggest upgrades to help better the server but these are not always needed.
3. Click on the server name “web-server-1”. This will take you to the the instance details page.
4. Under Machine type we are currently using f1-micro (1 vCPU, 0.6 GB memory) but google is suggesting we upgrade to a larger instance since our instance is being over utilized. The next upgrade would be g1-small(1 vCPU, 1.7GB)
5. Click Resize and follow the steps. This should only require a server restart. Cost will increase slightly.

In order to turn the server on and off, you need to ssh into the final production instance in with the project is deployed as a self-contained entity (both web component and db here). To ssh into it, download the acsas.key and, in the shell, go to the folder where you downloaded the key and issue this command: `ssh -i acsas.key ad@acsastruckparts.com`. After which you can issue the command `sudo service apache2 start/stop`. Then move to the `/var/www/html/` folder and issue `sudo node server.js` (if starting the server).

Reboot Documentation:

- 1) Start the apache server using: `sudo service apache2 start`
- 2) Start the MySQL database using: `sudo service mysql start`
- 3) Start the node api

- a) Navigate to the apache server files located in the /var/www/html directory. If you are located at the home user directory (the default location after rebooting), use the command: `cd ../../var/www/html`
- b) Start the screen application using the command: `screen`
- c) Hit enter to accept the terms of screen
- d) Start the website api using the command: `sudo node server.js`

At this point the website should be fully functional after a reboot. You can safely close the server connection window. Do not do this using the 'exit' command as this will shut off the api.

Some General Linux Notes:

Ssh: opens up shell

Icon in corner is where you upload files for updates

Pwd: gives current path

Ls: lists all files in directory

/var/www/html: server lies here

Uploading to the Server:

- 1) Navigate to the VM Instance page on your Google Cloud Console.
- 2) Click on the dots next to server and select "Upload File"
- 3) Navigate to the file you want to upload and select it.
- 4) SSH (click the SSH button on the VM Instance page) into the server. The file you downloaded will be located at the path "/home/acsastruckparts"
- 5) Use the command "`sudo mv /home/acsastruckparts/*yourfilenamehere* /var/www/html`" to move the file onto the web server.

Note: When you make coding changes to site (including new photos), you must upload new changes to server.

Changing Image for Homepage in GitHub:

- 1) Go to GitHub (<https://github.com/jilloestreicher/Triple-P>)
- 2) Navigate to the Front End folder
- 3) Navigate to the img folder
- 4) Navigate to the slider folder
- 5) Remove the current images being used for slider (can check in index.ejs file in views folder slider section, we left the images on homepage as 9.jpg and 10.jpg)
- 6) Upload new .jpg images using the same name (or you can change the name of .jpg images, but make sure you reference correct name in index.ejs file in slider section)

Note: Images for homepage should be around 953x227 in dimension in order to look clear. If the .jpg doesn't fit these dimensions well enough it will look blurry.

Note: If you want to add more pictures to the slideshow, you can copy the format of the slider area of the index.ejs file. Make sure you link another .jpg image that's in the same folder as mentioned above, and copy the format of the code by adding image to the slider images section and slider direction section.

Changing IP Address:

Should the IP address change for the site, complete the following steps:

1. Log in to Wix
2. Go to your domains
3. Click the domain acsastruckparts.com
4. Click Advanced
5. Click edit DNS
6. Edit the domain acsastruckparts.com points to
7. Click Save DNS

Google Analytics:

We created a Google Analytics account for the site to monitor user activity. We placed this script on user interactive pages:

```
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-152418607-1"></script>
<script>
    window.dataLayer = window.dataLayer || [];
    function gtag(){dataLayer.push(arguments);}
    gtag('js', new Date());
    gtag('config', 'UA-152418607-1');
</script>
```

Adding this script to additional pages will have it be tracked as well. The dashboard displays pertinent information such as users, sessions, bounce rate, and session duration. There is a Google Analytics tutorial course that can be registered for here:

<https://analytics.google.com/analytics/academy/course/6>. You can use Google Analytics to see things such as how long users retain a session on the site and which pages they visit most, which can be helpful in order to cater your site to your users better in the future.

Cost Analysis

It is difficult to ascertain a cost assessment of this development project, but we can make some general estimates. If we take into consideration the process of creating this site from its inception, all the way through its prototype phase into production level development, testing, and implementation, then there are a variety of pricing factors to account for. Firstly, we are providing this development service to our client for free, but if we were to be a paid service, then freelancers are usually paid around \$1,000 - \$3,000 on average. It's generally much cheaper to build a site yourself, although obviously more time intensive and requires development skills.

To build a site, you need to be able to host it somewhere, which also generally requires a cost. We utilized Google cloud server, which is priced at about \$5.75 per month, however we received \$300 of credit for a year to use. The only other real cost for our client to account for is Stripe, which takes a flat rate of 2.9% + 30¢ of each successful transaction (as long as the company does under \$1,000,000 in volume per year).

Now having examined all this, we can expect that our client will have a strong ROI. If after a year when he starts to pay for the server every month, he still only needs to sell one listing (which is currently priced at \$5 per listing per day), or one truck part as all products are significantly more than \$5.75 a month (the least expensive parts are around \$50, but most are more than \$100). In doing so, he would have a largely positive ROI, even if the amount of business he does doesn't increase, although in having this new site we expect better business for his company overall.

Project Plan

Date	Milestone	Tasks	Responsible*
8/28/19	Project Start	First class: make introductions; trade contact information; get assigned project: ASCAS e-commerce site; set-up team meeting	All
8/29/19	Team Meeting; 1st Homework Started	Schedule 1st team meeting - and attempt to choose recurring team meeting times that work for everyone (generally Mondays, Thursdays, or Fridays); establish a communications plan (share emails, text messages, Google Drive, Trello, and WhatsApp); Collect client information and questions for first client meeting; Meet with Dr. Rivas to confirm meeting times; Email client; Initiate project plan in Trello; Create GitHub repository	All, Jill initiate correspondence with client and will be liaison for future

9/4/19	Team Meeting; 1st Homework Continued	Further develop this project plan (it is part of HW1) - THINK through your time management and what you will accomplish as the weeks progress; Consider more possible client questions and requirements before meeting tomorrow; IT students meet; Talk with Dr. Rivas	All, Jake leave for first IT meeting
9/5/19	1st Client Meeting	Discuss requirements with client and Dr. Rivas; Record questions and answers; After call work on UML diagram and continue HW1	All, Jill lead discussion with client
9/9/19	Team Meeting; 1st Homework Continued	Meet with team and Dr. Rivas to go over what we've completed for HW1; Email project plan and user requirements to client by the end of today	All
9/11/19	Team Meeting; 1st Homework Due	Submit HW1 on GitHub and bring hardcopy to class; Present status report in class; By now will have completed gathering user requirements, writing up project plan, and designing our UML document; HW2 assigned	All, Jake set up first server
9/16/19	Team Meeting; 2nd Homework Started	Meet with team and Dr. Rivas to discuss HW2 (draft should be done by today) - the E-R diagram and its supporting documentation and our IT requirements; Should be keeping up with client via email on what we have completed and are planning to do	All, Jake handle IT requirements, Alex drafts ERD
9/18/19	Team Meeting; 2nd Homework Due	Submit HW2 on GitHub and bring hardcopy to class; Present status report in class; By now will have also completed our E-R diagram with documentation and our IT requirements; HW3 assigned	All
9/23/19	Team Meeting; 3rd Homework Started	Meet with team and Dr. Rivas to discuss HW3 - UI/UX design; Utilize InVision to create mockups; Communicate our ideas with client via email; Continue updating E-R diagram if needed and working on database	Jill and Herb work with InVision, Kaitlyn research UI resources, Alex work on

			database
9/25/19	Team Meeting; 3rd Homework Due	Submit HW3 on GitHub and bring hardcopy to class; Present status report in class; By now will have also completed five wireframes (or mock-ups) that show the most important views of the user interface; HW4 assigned; Dr. Rivas not here	All
9/30/19	Team Meeting; 4th Homework Started	Meet with team to discuss HW4 - creating a prototype; Discuss and update UI/UX design if needed; Communicate our ideas with client via email; Continue updating E-R diagram and database if needed	All, front end coding done by Alex, Kaitlyn, Herb, and Jill
10/2/19	Team Meeting	Finish final version of our E-R diagram and a functional database with some sample data that Dr. Rivas can query live; Dr. Rivas not here	Alex and Jake setup database on server
10/7/19	Team Meeting; 4th Homework Continued	Meet with team and Dr. Rivas to test database; Discuss and update UI/UX design if needed; Communicate our ideas with client via email	Alex and Herb setup Stripe
10/9/19	Team Meeting; 4th Homework Continued	Dr. Rivas queries database; Present status report to class regarding work on HW4 and any updates on UI/UX design; Jake goes with IT to start setting up second server; Jill goes with IS group to discuss life-boat exercise	All
10/14/19	Team Meeting; 4th Homework Continued	Meet with team and Dr. Rivas to continue discussing and updating HW4; Discuss and update UI/UX design if needed; Communicate our ideas with client via email	All
10/16/19	Team Meeting; 4th Homework Early Demo Due	Submit HW4 early demo on GitHub and bring hardcopy to class; Present status report in class; By now will have also completed early demo of system that includes user interfaces that are connected and navigable, a proper connection to the DBMS, all external APIs and accounts tests, and implemented test cases in a functional stage; Mature demo assigned;	All

		This is midterm checkpoint	
10/21/19	Team Meeting; Mature Demo started	Meet with team and Dr. Rivas to discuss mature demo; Discuss and update UI/UX design if needed; Communicate our ideas with client via email	Herb works on checkout session, Jill cleanses and parses data dump from client, Alex imports data into database, Kaitlyn works on logging into account for sessions, and Jake starts setup for Google Cloud server
10/23/19	Team Meeting; Mature Demo Continued	Present status report to class regarding work on mature demo and any updates on UI/UX design, run into Stripe account issue to be resolved	All
10/24/19	Meeting to Setup Google Cloud Server	Jake meet with Dr. Rivas to setup Google Cloud server	Jake setup server with Dr. Rivas
10/28/19	Team Meeting; Mature Demo Continued	Meet with team to discuss mature demo; Communicate our ideas with client via email	All
10/30/19	Team Meeting; Mature Demo Continued	Present status report to class regarding work on mature demo; Meet with IBM employees to get feedback on site	Alex implements feedback into code

11/4/19	Team Meeting; Mature Demo Continued	Meet with team and Dr. Rivas via Webex to discuss mature demo; Figure out GUI connection to server; Communicate our ideas with client via email	Herb finish cart session; Jake finish setting up server environment; Kaitlyn finish login; Alex setup database on new server; Jill work on documentation for Wednesday demo and figuring out future tasks to finish project
11/6/19	Team Meeting; Mature Demo Due	Meet before class to discuss goals post mature demo; Present status report to class regarding our final UI/UX design which is due and our mature demo; Present demo to class following status report; By now will have also completed UI/UX design and mature demo which includes: how our development environment currently is deployed to simulate the IT requirements, how our user interface matches the use cases, how our database matches our use cases, how our interface functions and connects with our database, how much progress we have made and if we are right on schedule, are there any risks of not completing the project, and if it is possible to deliver it early; Final project presentation, paper, and prototype assigned	All
11/11/19	Team Meeting; Final Project work started	Meet with team and Dr. Rivas to discuss project; Transfer from Wix domain; Setup Apache certification	Alex, Herb, Jake, and Jill meet

11/13/19	Team Meeting; Final Project Work Continued	Group meeting to discuss rest of project goals to be completed	All
11/18/19	Team Meeting; Final Project Work Continued	Meet with team and Rivas to discuss project	Jake upload new code to server
11/19/19	Team Meeting; Final Project Work Continued	Meet with team to reevaluate to-do list for current week and upcoming week; Decide to focus on admin functionality and finalizing order processing	Alex, Herb, Kaitlyn, and Jill meet
11/20/19	Team Meeting; Final Project Work Continued	Present status report about work completed for final project so far; Group work in class to be done; Review presentation requirements	All
11/25/19	Team Meeting; Final Project Work Continued	Meet with team to discuss project; Plan work and testing for break; Outline presentation ideas to finalize when we come back	All
11/27/19	Work from Home	No class for Thanksgiving break; Team will work from home to continue developing and testing project	All will be finishing up site; Everyone testing to catch any issues; Work on project paper and video
12/2/19	Project handoff	Final project handoff to client	All
12/4/19	Final Project Assignments due	Dr. Rivas not here; Submit all deliverables on GitHub and have presentation ready; Present to class	All
12/11/19	Optional Section-Wide Final Project Presentations	Possible presentations	All

[1] *Note about group responsibilities:

IS member (Jill) - Lead the design effort: gathering requirements, documenting the team's approach, and developing the design documents that will serve as the blueprint for the team(s); Debug and test code written by your team or other teams; Act as the team's project manager, managing the operations of the project day to day, ensuring progress is made weekly, and developing and delivering status reports regularly to the instructor and/or entire class; Liaising with the client and instructor as deemed appropriate by the instructor on all matters of clarification or other communications as required; Participate in developing code as necessary.

IT member (Jake) - Secure, configure, and maintain a physical and/or virtual working IT environment on equipment provided in the Hancock datacenter, or on your own equipment, for use by all team members to develop and deliver code; Debug and test code written by your team or other teams; Act as an integration engineer, actively engaged in bringing disparate parts of the coded solution together and making them work together efficiently; Liaise with the IT professionals at the client site, including IT contractors if necessary; Participate in developing code as necessary.

CS members (Kaitlyn, Alex, Herbert) - Develop the software required for the project, including the installation and configuration of any frameworks, applications, tools or other necessary software components; Participate in the requirements gathering session to ensure the right questions are asked and the best answers are received from the client; Act as a "team" of coders, coordinating efforts for efficiency and delivering quality code in a timely fashion; Document all code, including any instructions for the access, integration, compilation and installation of the code; Debug and thoroughly test all code you write, and support the test and integration efforts of your code with that of other teams; Contribute to the design effort; Provide accurate and timely status as required.

All members - Participate in status presentations as required; Participate actively in the creation and delivery of the final presentation to the client; Take on additional work assigned by the

instructor throughout the semester whether or not it is related to the project; Communicate well and contribute sufficiently; Provide timely feedback including an honest peer evaluation assessment of every other team member.

Ethics

In building a website, it is easy to overlook the ethical issues that might be at hand in favor of spending time on user interface design or implementing some cool new features.

However, creating and/or owning an e-commerce site means that there is a necessity to understand the ethical implications of what your site does. Now that e-commerce has become the major channel for selling products and services, it is essential to consider it with an ethical mindset in order to be a respected online brand. We believe we have created this ACSAS e-commerce site with careful deliberation over all ethical aspects involved.

Perhaps one of the biggest areas of ethics to consider in creating and maintaining an e-commerce site is data privacy and security. The saying that "data is the new oil," is not unfounded, as data has spawned a lucrative industry and been the motivation behind new regulation, much like oil has in the past. We understand that for an e-commerce site important and private information is being utilized, such as for payments and user accounts. We made sure to enable a password protected database using an SSH key, as well as implemented an HTTPS certification using Certbot. We also made sure that in order to create an account on the site, only pertinent information is requested and safely stored in the database. Since the site uses a third-party payment system, we had to carefully consider all the options available to us. We ended up using Stripe, because it had great reviews for being simple and secure, along with lots of helpful documentation for how to implement it. Stripe uses Ajax to transfer credit card data to their system, which is a good thing because then the customer's credit card data doesn't touch our

system. This creates limited liability and meets PCI (payment card industry) compliance requirements.

Another thing to consider about developing this site in regard to data privacy was the amount of personal information we as developers had about our client. In order to set up accounts for him through Google and Stripe, as well as in order to fill out information for the site in general, we gained a lot of personal details about him. This included banking information, business information, and other private details that we had to make sure stayed private. We made sure not to publicly share any of this information and had only one person in charge of the payment details in order to limit exposure of the material.

When developing this site, we also had to keep in mind matters of intellectual property. This was fairly easy for us, as we had permission to utilize ACSAS photos and logos from our client's old sites. Any other photos we used were taken from open source platforms. We used an open source web framework for the design of our site as well, and credit is given in a link at the bottom of our site pages. We made sure to comply with the requirements of the framework's stated policy found at freethemescloud.com that states the free download of this framework can be used for sales but needs to have the copyright credit. Our own code is not copyrighted and is posted on a public GitHub as we needed to be able to share the repository with more than three people. This theoretically means someone could find and use our code, but it is a risk we are willing to take in order to utilize GitHub's free account features.

One possibly problematic aspect of our site is that it allows for people to pay to list their own trucks and trailers. We have the site setup so that customers can upload their own photos with their listing, and thus there is the possibility for individuals to abuse this system by

uploading something inappropriate. This means that there needs to be constant monitoring of the site by admins in order to catch any possible misuse of this feature. Considering customers have to pay to use this, we don't expect it to be much of an issue, but it is definitely something to be aware of.

As the creators of this site, it was our responsibility to represent ACSAS with a strong, professional image. This means being honest with website visitors by ensuring product photos are accurate, the about section conveys pertinent and real information about the business, and all statements on the site are true. Unfortunately, the listings portion is up to the paying customer to provide accurate information. If the admin thinks something is off about a listing it is their duty to look further into the posting, but ultimately it is a risk we are accepting to trust the listers to post truthful photos and details about their trucks and trailers.

E-commerce inherently provides a barrier between customers and retailers, so we had to take into account the ability for customers to be able to receive good customer service. Unethical e-commerce sites can make it difficult for customers to get help by either delaying their response or ignoring them altogether. We incorporated an about us page as well as links at the bottom of each page to ACSAS's social media and email to allow for easy means of communication should there be an issue. We wanted to make the contact information obvious for site visitors as well as provide multiple communication mediums.

As can be seen, we were careful to consider many potential ethical issues that might arise as a result of making an e-commerce site. As the web has become such an important resource in the realm of commerce, it was important for us to make the site ethically accessible to suitably represent ACSAS. We believe we did this to the best of our ability.

Appendices

Other documents created for project planning:

Risk Assessment

Event Description	Probability	Impact	Priority	Response	Owner	Details
Not being prepared with enough skills to do capping	L	M	M	Mitigate		Work hard precapping to learn necessary skills
Having team members with poor work ethics	H	M	M	Mitigate		Set goals, due dates, and meeting times early
Experiencing team conflict	H	M	M	Mitigate		Understand everyone's personality and how to work together early
How professor controls project	M	M	M	Accept		Understand professor's requirements and work around his plans
Can't find a good time to meet with team members	M	L	M	Avoid		Set meeting times early, make sacrifices with time
Dealing with a change in scope/requirements	H	H	H	Mitigate		Have contingency plans for this, be flexible
Client doesn't know what they want	M	L	M	Share		Talk with client and share ideas with professor to develop ideas
Not having contingency plan	M	H	H	Avoid		Necessary to develop early
Not having backup storage should data be lost	L	H	H	Avoid		Always have backup
Server going down	L	H	M	Transfer		Use other working technology if available
Not having available technology needed to do project	L	M	L	Transfer		Find technology through professors and other students
Using new technology/apps.	L	L	L	Accept		Can use to advantage, lots of tech./apps. that are free and useful
Relying on unfinished parts to continue project	M	M	M	Mitigate		Work around uncompleted work, work early and plan often
Falling behind in schedule	M	H	H	Avoid		Stick to schedule as much as can
Falling behind in other classes	M	M	M	Avoid		Allot time for self and plan schedule to be lighter this semester
Someone gets sick/unforseen absence	L	H	H	Accept		Unforseen circumstance, but work early to have as much done as possible should this happen
Not having finished deliverable by project end	M	H	H	Avoid		Work hard and often
Client being unsatisfied with deliverable	L	H	H	Avoid		Make sure understand what they want early, ask lots of questions
Not passing capping	L	H	H	Avoid		Work hard!

Used to identify possible risk that could affect project being delivered and how we could mitigate them. Luckily our team was not affected too harshly by these risks, but it was helpful to be aware of them.

Communications Plan

Stakeholder	What Do We Need To Tell Them	How Do We Communicate With Them	Frequency
Client	Initial meeting discussion; requirements discussion; progress of project; final deliverable	Initial emailing/contact through professor; potentially meet in person to gather requirements if possible; meet in person for final delivery of project; documents that explain project and decisions	Weekly; Meet for final presentation
Professor	About your group; your project decisions; any questions	Verbally; emailing; iLearn; project documents	Multiple times a week (every class); everytime issue/question is encountered
Project Manager	About group decisions; all project details	Verbally; emailing; iLearn; project documents; texting; phone calls; Slack	Every class; during set meetings; when new information needs to be shared; when in need of questions about direction of project
Group Members	All project details; whenever new information is discovered	Verbally; emailing; iLearn; project documents; texting; phone calls; Slack	Every class; during set meetings; when new information needs to be shared
End Users	To gather requirements and answers to questions	Through emailing; through professor or client; possibly meet in person to question them	In beginning upon gathering requirements and information; after project is utilized to discover results and opinions
Project Implementers (those who control technology of client)	IT requirements; questions about current system	Emailing; through professor and client; in person if possible; project documents, such as how to use technology	In beginning upon gathering requirements and information; after deliverable to help implement; after project is utilized to discover results and opinions
Other Groups	Project discussion; discover ideas	Emailing; texting; iLearn	Class times (especially in beginning upon project assignment); when questions arise

Used to identify who we will be in communication with throughout the project, and how frequently they should be contacted.

References

“Free Website Templates and Themes.” *Free Themes Cloud*, <https://freethemescloud.com/>.

“Build Software Better, Together.” *GitHub*, <https://github.com/>.

“Gmail - Email from Google.” *Gmail*, <http://mail.google.com/mail/>.

Google Analytics, Google, <https://analytics.google.com/>.

Google Cloud, Google, <https://cloud.google.com/>.

“Stripe.” *Online Payment Processing for Internet Businesses*, Stripe, <https://stripe.com/>.