

TECHFAM

CMPSCI 465

Phase One: Shoe Retailers' Database

Ben Myers

Chao Guo

Hiren Patel

Guojing Wang

Jeff Li

Dishti Gurnani

TABLE OF CONTENTS

Table of Contents	i
List of Figures	iii
1. Introduction.....	1
2. Analysis and Conceptual Design.....	2
2.1. Sales Item.	2
2.2. Categories	4
2.3. Suppliers.....	5
2.4. Registered Users.....	6
2.5. Rating.....	7
2.6. Browsing.....	9
2.7. Searching.....	10
2.8. Sale.....	11
2.9. Bidding.....	12
2.10. Order and Sale Reports.....	13
2.11. Delivery.....	15
2.12. Custom Shoes.....	16
2.13. Customer Service.....	17
3. Conclusion	19
Appendices.....	20

Appendix A – GitHub Graphs.....	20
Appendix B – Asana Summary.	21
Appendix C – Category Tree.....	22
Works Cited.....	23

LIST OF FIGURES

Figure 1 - Sales_Item ER Diagram	2
Figure 2 - Sales_Item and children ER Diagram	3
Figure 3 - Address ER Diagram.....	4
Figure 4 - Image ER Diagram.....	4
Figure 5 - Category ER Diagram	5
Figure 6 - Suppliers ER Diagram.....	6
Figure 7 - Registered_Users ER Diagram	7
Figure 8 - Rating ER Diagram	8
Figure 9 - Searching Process Diagram.....	10
Figure 10 - Sale Diagram.....	11
Figure 11 - Auction ER Diagram.....	13
Figure 12 - Bid ER Diagram.....	13
Figure 13 – Order Sale Report Process Diagram.....	14
Figure 14 – Shipping and UPS ER Diagram	15
Figure 15 - Custom_Shoe ER Diagram	17
Figure 16 - Customer_Service ER Diagram	18

1. INTRODUCTION

When forming our startup, we considered several different items to sell. We wanted to distribute a popular product that was easy to categorize for simple website navigation. We brainstormed and decided to join the \$52-billion-dollar shoe industry.^[5]

After choosing to create an online shoe store, we looked at competing companies to see how they displayed their products. We noticed Amazon provides a name, description, price, image, company name, delivery information, and availability (sometimes including number in stock).^[1] While looking at eBay, we found additional important details, like current bid amounts and condition of the product.^[3] This gave us a general idea of what information we needed to store for each sales item.

Next we looked directly at the competition, and began browsing shoe sites. We noticed that stores like Designer Shoe Warehouse allowed you to search for shoes based on category, brand, size, and color.^[8] We also discovered that shoe stores often sell other products like socks, backpacks, and shoe care items.^[2]

Using our knowledge on Database Management Systems we will develop an envisioned model that helps customers shop for shoes in the most intriguing way possible. We will meet the following requirements described further in this analysis.

2. ANALYSIS AND CONCEPTUAL DESIGN

In the following section, we analyze the different entities that our database will store, as well as several processes that will be implemented on our site. The goal is to provide a brief introduction to the user experience and explain some of the behind the scenes functionality of the website.

2.1. Sales Item

With our initial research complete, we designed the **Sales_Item** entity to represent each product (See Figure 1). It is identified by its unique key, *item_id*. Every sales item has a *name* and *description* so sellers can properly market their product. We added a *brand* attribute so users can search for their favorite brands (See Section 2.6). Each sales item has two prices recorded, the *list_price* and the *reserved_price*, which are explained in more detail in Section 2.8 and 2.9. The *keywords* attribute helps the user retrieve their item when searching (see section 2.7). So if a person types in terms in the search bar that match one of the keywords in the product, then that product will be retrieved. The *condition* attribute describe the quality of used products. Finally, the *count* attribute tracks how many items are in stock.

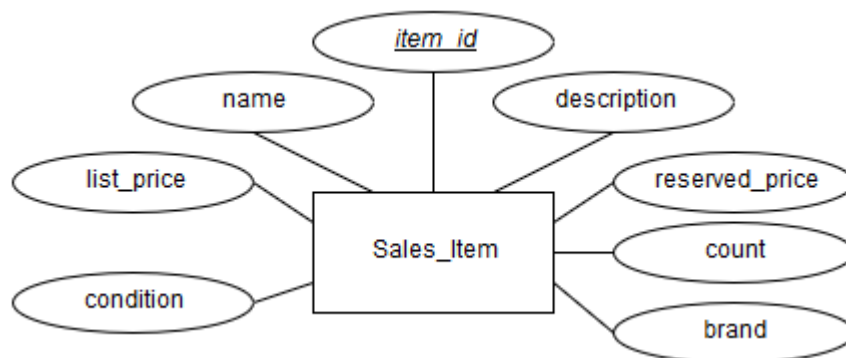


Figure 1 - Sales_Item ER Diagram

Like other shoe websites, we wanted to sell non-shoe items. We provide bags, sunglasses, shoe care items, and socks. Bags, sunglasses, and care items can be fully described by the **Sales_Item** entity, but socks and shoes require additional information. As a result, we created the footwear entity (See Figure 2) to account for these missing fields, which includes the *size* attribute (for socks and shoe size), as well as a *footware_id* unique key. Shoes also require a *width* description, so a **Shoe** entity (See Figure 2) with a *shoe_id* unique key and a *width* attribute was created as a sub-entity of **Footwear**.

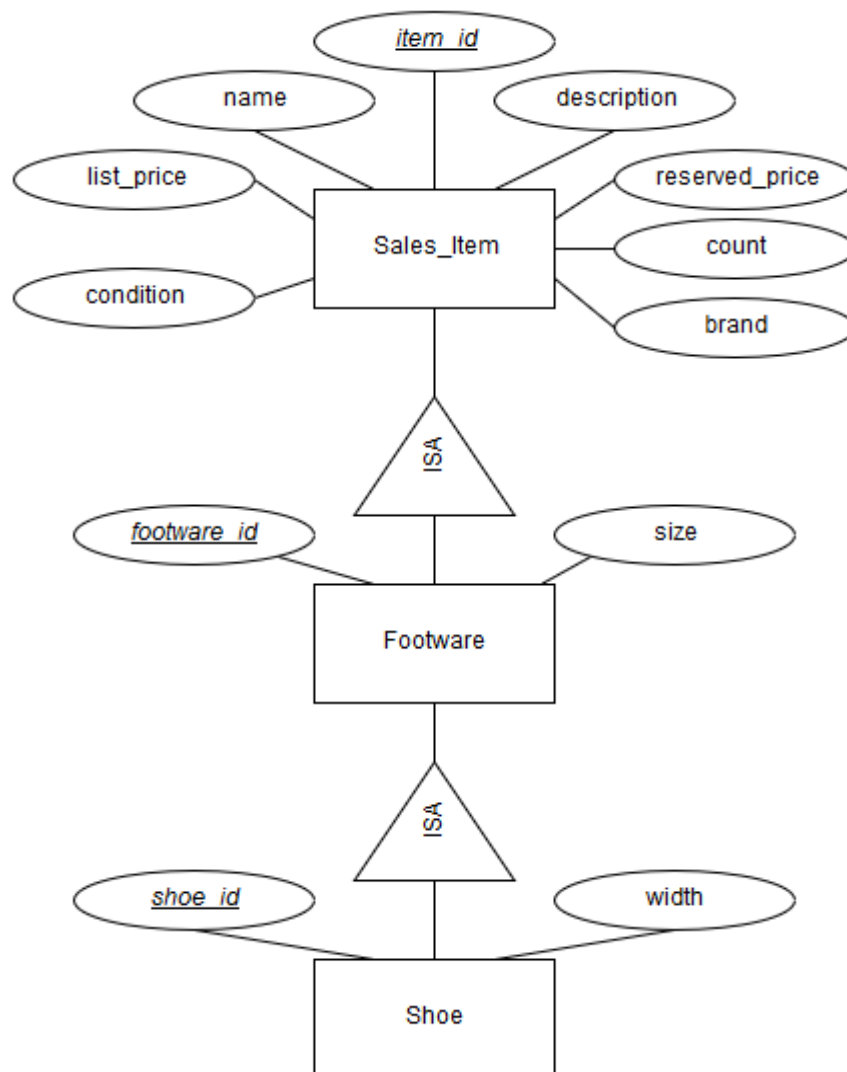


Figure 2 - Sales_Item and children ER Diagram

Also, **Sales_Items** require information that is too complex to represent as attributes. The **Address** was created as a separate entity because attributes may be reused by other entities, such as users and shipping (See Figure 3). Note that every **Sales_Item** is required to have at least one address. Attributes of the **Address** entity include the *street_address*, *app_num* (apartment number), *city*, *state*, and *zip_code*. The other separate entity is the **Image** entity (See Figure 4). Every **Sales_Item** is required to have one image minimum, which will be displayed on each product's webpage. Each image has a unique *img_id*, as well as an *image* attribute, which holds the actual photo data. But the image entity also has an attribute called *color*. This attribute allows the user to filter out products when searching or browsing, based on the color of bag, shoe, or sock that they desire (See sections 2.6 and 2.7).

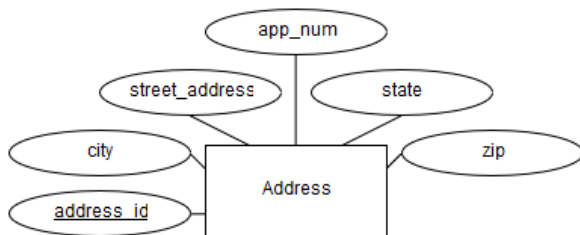


Figure 3 - Address ER Diagram

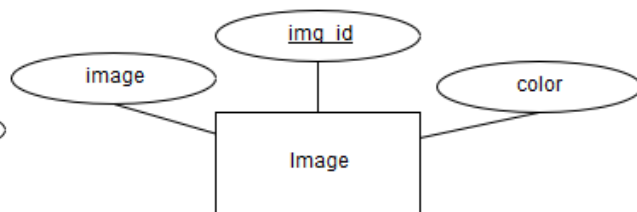


Figure 4 - Image ER Diagram

2.2. Categories

There is a variety of footwear to categorize. This categorization will dictate the databases needed for this online business. TechFam must distribute enough styles of footwear to become a central hub for all individuals looking to purchase shoes. However, it is critical to be efficient when creating the databases, avoiding any data redundancies.

TechFam's shoe selection targets people of all genders and ages. As a result, the root node, "All," must have three children, "Men", "Women", and "Kids". The children nodes of these three nodes are categories of the different types of shoe, such as "Boots," "Sneakers," etc. Next, the

different types of shoes are broken down even further by subtypes. In addition to the first three children nodes, there is a fourth node, “Accessories,” contains items that are not shoes. See Appendix C for an in-depth visualization of the categories.

The ER diagram in Figure 5 explains the basic outline for how each category will be stored in the database. **Categories** have primary key, *categoryID*, to uniquely identify each other. The *description* attribute briefly details the kind of product that is contained within the category itself. Note that there are relationships between the category and its parent node and children node(s) which follow the classification tree in Appendix C.

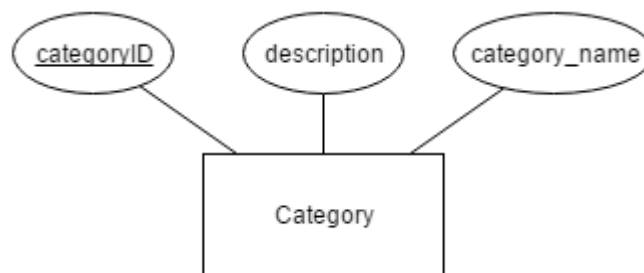


Figure 5 - Category ER Diagram

We researched a few different shoe websites to obtain the categories. Zappos^[7] is a general shoe distributor that contains the vastest selection, contributing to the majority of our category ideas. JustFab^[11] delved deep into the various types of women’s shoes.

2.3. Suppliers

Suppliers are companies or sellers that sell the shoes that they manufacture on our website. The attributes for the **Suppliers** entity include the company *name*, company *category* and point of *contact*. Multiple phone numbers of the suppliers is also recorded. The name of the company, the point of contact, and phone number helps the customer contact the seller in case there are any questions or concerns. The company category is a useful tool that helps enhance the searching for products. The supplier is assigned a *company_name* and is used to identify the supplier.

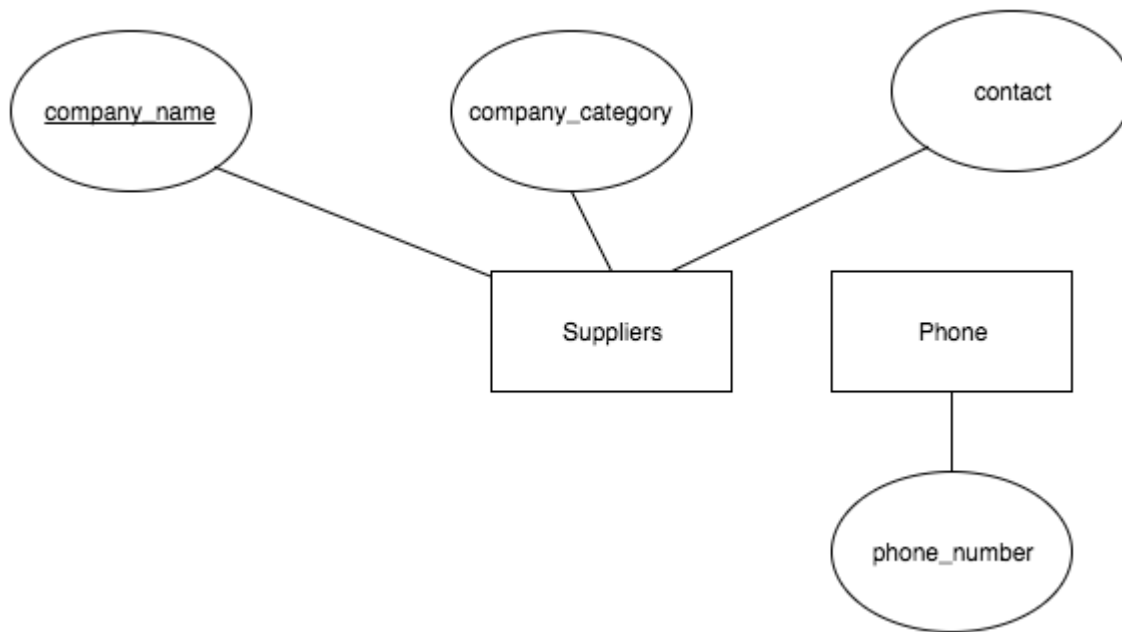


Figure 6 - Suppliers ER Diagram

2.4. Registered Users

Registered Users can bid or sell products. This feature asks the user to fill out the following information while they register:

- A user ID or username, which helps users stay anonymous and allows us to identify them.
- A password, to secure their account.
- Gender.
- An email and multiple phone numbers as a mode of contact to the registered user.
- Multiple credit cards and addresses (as described in section 2.1 in figure 3) of the registered user, which helps process payments, billing information, and return information.
- Age and income is used to calculate a statistics for each user. The income here stands for the income the user has made on our website.

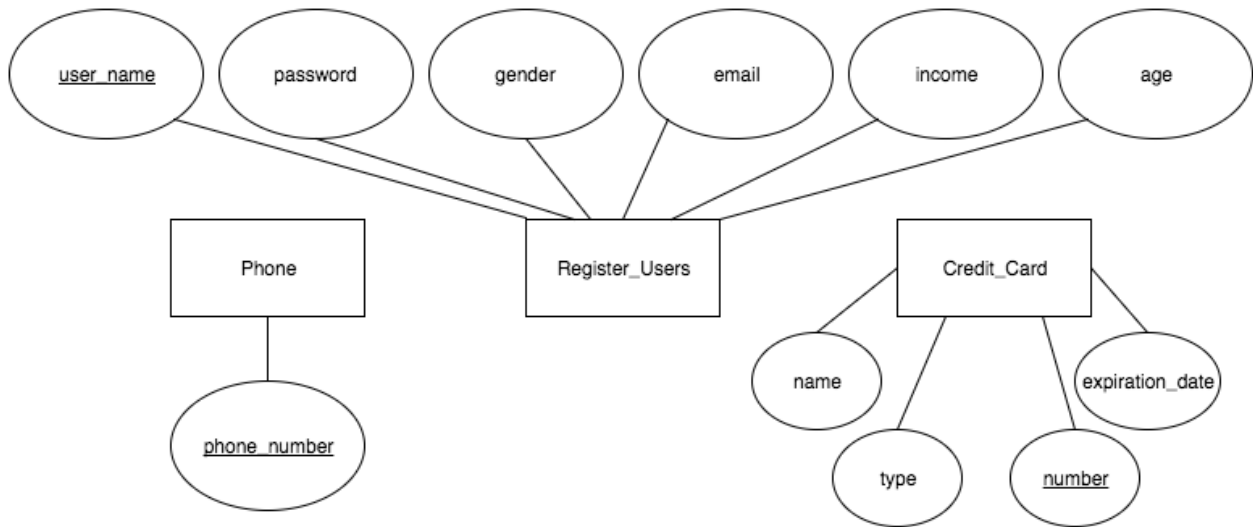


Figure 7 - Registered_Users ER Diagram

2.5. Rating

Rating is an important measure for a seller's reputation. A user might want to know what rating a seller might have. Many websites now have rating system for sellers. For example, sellers on eBay can earn stars based on the number of ratings they get^[4]. The rating system in this database can be modeled by the relationship, *User sells Items*, as shown in Figure 7. The application quality entity **Rating** is created with primary key *rating_id*, a scale based non-key attribute *value*, and a textual non-key attribute *explanation*.

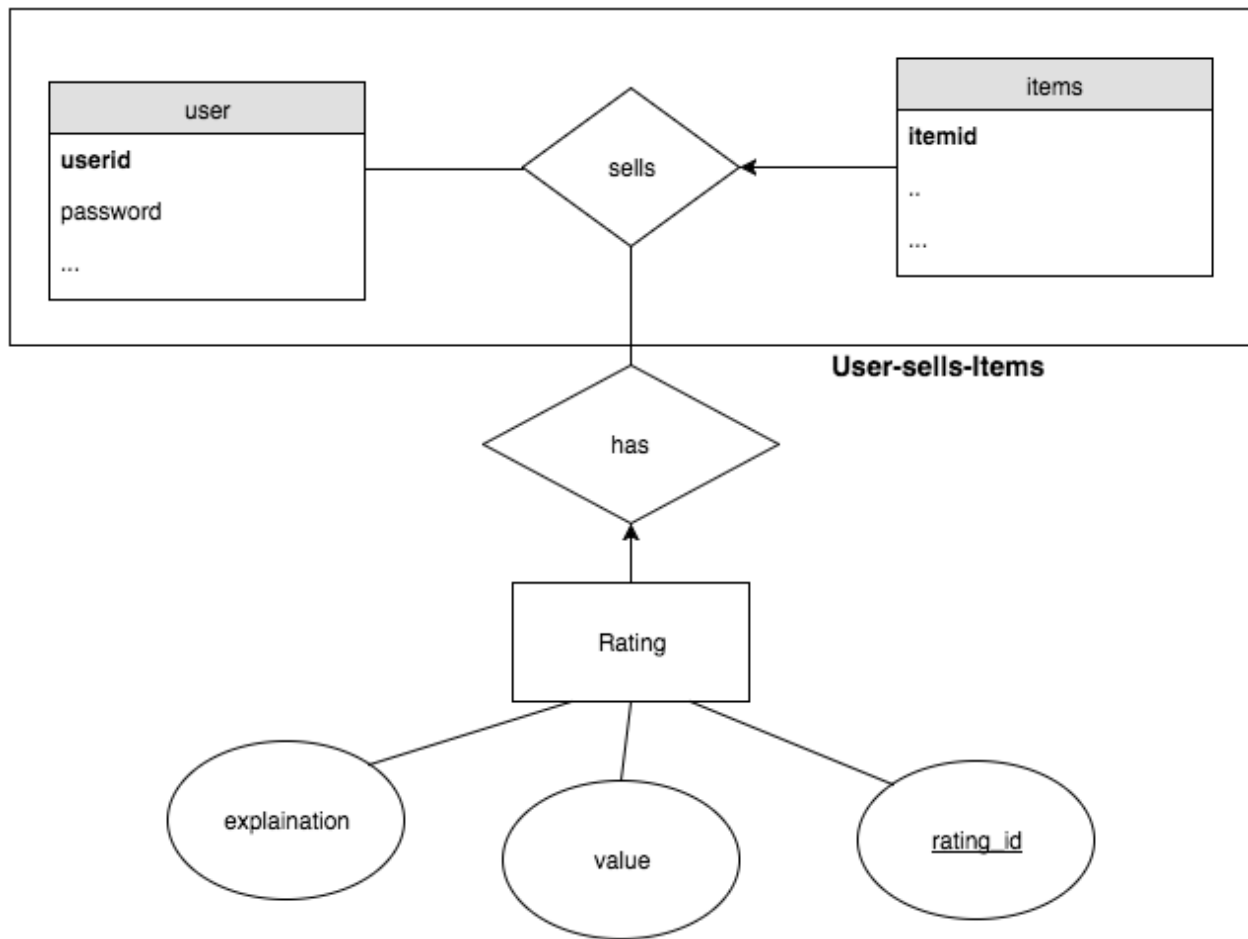


Figure 8 - Rating ER Diagram

Each occurrence of the relationship “User sells ” *Items* has a rating value. Since the “User sells Items” is a many-to-many relationship, the primary key of the gerund is the concatenation of the keys of *User* and *Items*. Then a new relationship “User sells Items ” has ratings has been created. The *value* is a scale attribute that can be selected from 1 star to 5 stars. The *explanation* is a textual attribute that allows a user to explain why one gives such scaled-rating or any additional comments. A seller will also have an average rating showing on his/her profile. The average rating calculates the mean of each rating value then rounds to the nearest star scale.

2.6 Browsing

Users are able to browse the items by traversing the category tree (Figure 2.1). When a user chooses a category, the user will want to know all of the categories that are directly linked to that current category. The tree traversal algorithm to be implemented will follow a derivative of the breadth-first search algorithm. At each point, they are given a summary of all the items that appear in that category. For example, if the user is in the “Women’s” category, the provided categories are “Boots”, “Flats”, “Pump”, “Sandals”, “Sneakers”, and “Wedges”. However, in traditional breadth-first search algorithms, the other nodes in the same level as the previously mentioned categories, will be shown. So our form of the algorithm will limit the traversing within the current node and its children.

Another important browsing feature is to allow the user the ability to traverse both ways through the tree, including moving downwards and upwards through the tree. The user will be able to return to the parent node of their current category. In addition, there will be a home button to traverse the user back to the root node.

In the browser bar, user can access their account information. This includes their profile, feedback, and settings. Profile contains their personal information, items they are selling, current bids, and history. Feedback has ratings and reviews. Settings include login information, credit card information, and shipping/billing address.

There will be a homepage which consists of the current deals and other features usually found in homepages. This homepage is not located anywhere in the classification tree, yet contains various items from different categories.

During our research of other shoe/clothing sites, we were attracted to a certain type of user interface that made browsing very easy and fast. Particularity, Macy’s online store featured

a row of tabs near the top of the page. These tabs were the major categories of clothing, “Men”, “Women”, etc. When the user selects one of the tabs, a drop down menu appears, listing all of the categories within that major category. Furthermore, under those categories were subcategories that were indented to easily show its lower levelness. We found this interface in other sources as well and we find this way of browsing to be the most user-friendly because it gives the user a broad overview of the classification tree. TechFam will follow this structure of browsing.

2.7. Searching

On shopping websites, like Amazon, a user might want to search for a series of specific items. This can be modeled by two kinds querying, “User search Items”, and “User choose Categories then search Items”, as shown in Figure 9.

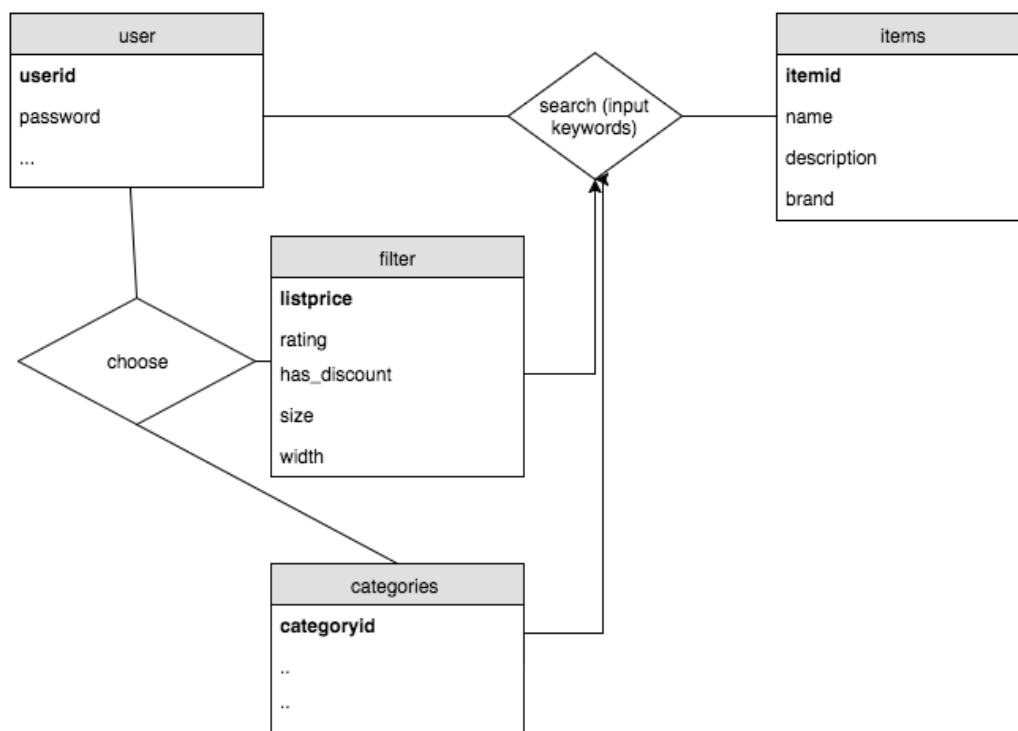


Figure 9 - Searching Process Diagram

If a user chooses to input keywords, the database will look for items that contains these keywords. A user can also choose certain categories or other filters first then input keywords to narrow down the results. Attributes in **Sales_Item** like *name*, *description*, *brand*, etc. shall be searchable.

2.8. Sale

Items on sale will be the ones not auctioned off. We will create an option for buying the item now. When the user enters a specific item description, that item information will be pulled from our database. The user interface will display all the items matching that description. The user will be allowed to add items to the cart, use their credit card information and check out. After the transaction has completed the item count will decrement. The transaction information will remain in the system for six months and will be stored using the ER diagram in Figure 10.

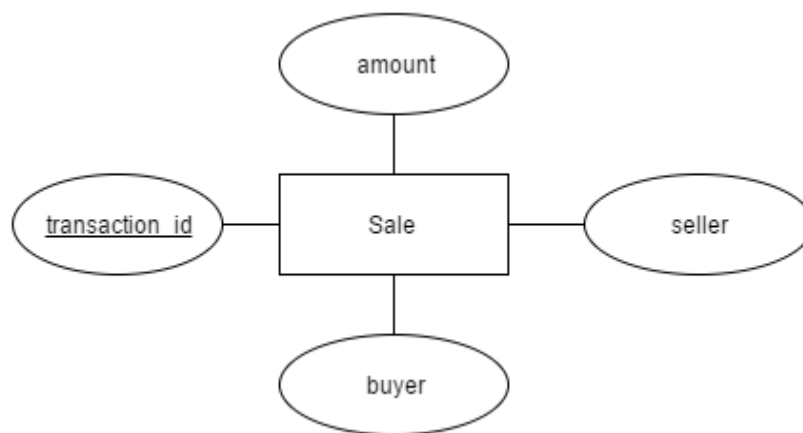


Figure 10 - Sale ER Diagram

2.9. Bidding

Unlike many shoe websites, our database will have the option of auctioning items. A user can access these items via searching the pull-down menu and clicking on an item through the categories included, as well as entering a specific item name through the search bar. The database will retrieve information and after clicking the auctioned items option the user will be able to view the item. The user has the option to sort items based on price from highest to lowest or vice versa.

Bidding Rules:

- Users must remember each bid is a contract.
- Each item will have a specific time stamp listed under description.
- Bids can only be canceled 24 hours prior to time stamp deadline.
- Bids will be in 2 dollar increments.

Selling an item:

- Under options, select selling button and Users can only sell one item at a time.
- Select the item for sale and give a brief description.
- List the timestamp value for the starting and ending.
- Select the reserve price for the item.
- Review everything and select the confirm button.
- You will be notified when the auction has ended and who had the winning bid.

We will feature manual bidding and automatic bidding. First let's talk about the manual way, where the user enters a bid value that is at least \$2 more than the current highest bid. The user will be notified when someone overtakes them as the lead bidder. On the other hand, we have automatic bidding, where the user will be prompted to enter the maximum value they are willing

to pay for the item being auctioned off. The system will bid \$2 more than the current highest bid until your bid reaches the maximum value you are willing to play or the auction ends. The user with the highest bid wins, and pays the amount they bid. For both bidding options at the end of the auction, the user with the winning bid will be notified, that he or she has won. The system will charge the user's credit card with winning bid amount and after receiving the payment, the seller will be required to ship the item. The transaction will stay in the system for 6 months. The user can view their current bids under the account options.

The **Auction** will have an *auction_id* pertaining to each item in system that is on auction. It will have a *time_stamp_start* for the start of the auction and a *time_stamp_end* for the end of the auction. The auction will record all the bids that users place and these will have a relationship to the sale items.

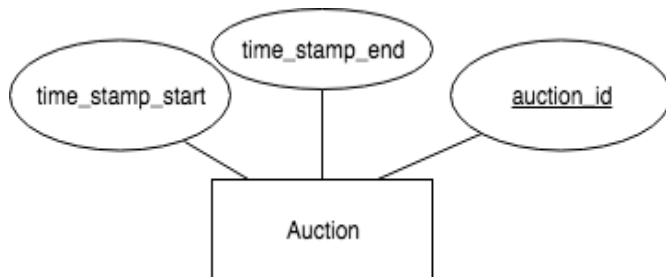


Figure 11 - Auction ER Diagram

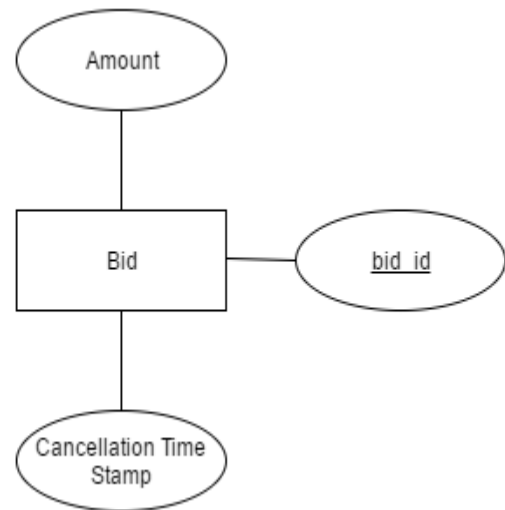


Figure 12- Bid ER Diagram

2.10. Order and Sale Reports

With everything set up, it is important to consider how our system keeps track of all the orders that go through the system. We want to be able to generate a weekly report for each

category that we are selling. This will not only inform us on how we are doing in terms of revenue but also help us monitor consumer behavior. By recording and analyzing reports of user activity, companies can improve their sites functionality.^[9] Having this functionality in our application and design, we believe our marketing team will be able to take a full advantage of this report function and produce the most effective marketing strategy that will help to maximize company's profit.

Moving into the implementation of our report, each report will carry a specific report id that is relevant to the seller account detail and the time period for which the report is on. Our goal is to make our report as informative and dynamic as possible. We want to create an interface that allows our users to easily assess their data with a few clicks of a button.

Our design will include a bunch of useful keywords that relate to type of data that will be displayed such as total revenue, total cost, number of purchases, etc. In the backend of this functionality, we will use these keywords and put together a query that will access our database and pull up the request data and present them in a neat view.

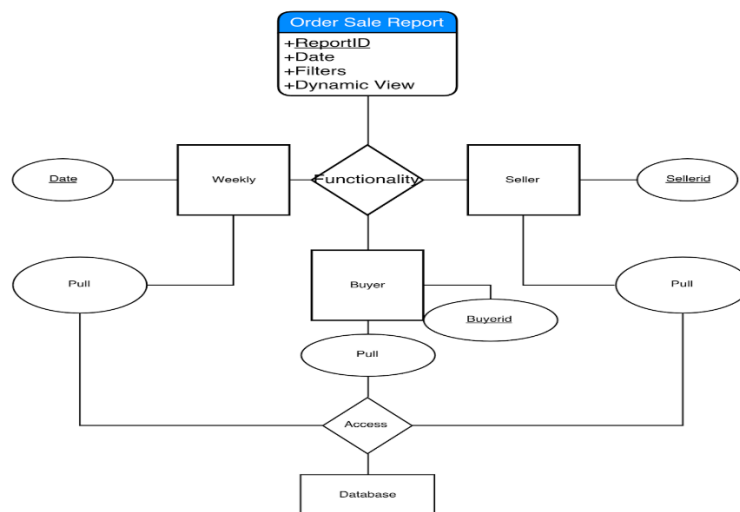


Figure 13 - Order Sale Report Process Diagram

2.11. Delivery

The other important thing that we are focusing on is our delivery system. We understand that it is our duty to fulfill our customers' orders with the best service possible. This is why we are taking extra steps to secure every single transaction amongst our users.

With our delivery system, we will generate a unique id for every delivery that is only available for the buyer, seller, and our engineers. Nevertheless, this id will only be created after a confirmation of payment has been posted. This means there will be less of a chance of messing up shipping confirmation and confusing our shipping API providers. Why compete with shipping companies when companies like UPS, FedEx, and DHL have proven themselves to be among the best in the industry. While companies like UPS help to provide their customers with useful developer API, we aim to utilize their API to create the best interface for our users without traversing to another website.^[10] We want to provide the tracking number for the shipment to allow users to see the status of their shipment from our website. We hope that our efforts in providing a friendly user interface will keep our customers satisfied with our products and services.

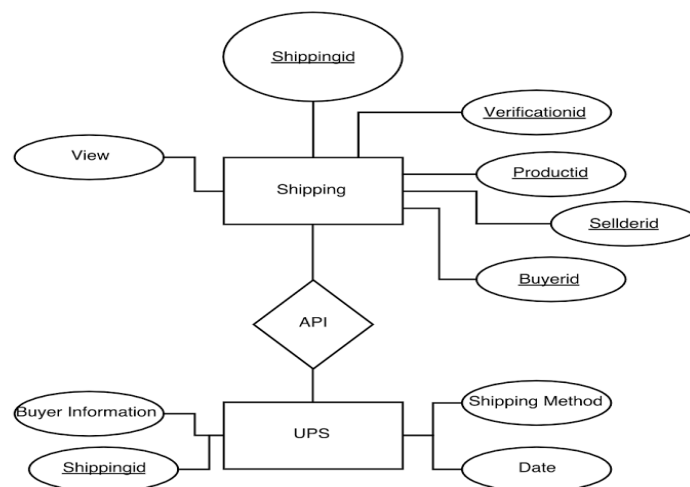


Figure 14- Shipping and UPS ER Diagrams

2.12. Custom Shoes

Unlike most sites, our team decided to create a custom shoe feature. We researched Nike's custom shoe system for guidance.^[6] First, the user will click on the custom shoe feature in the browser bar. Then, they can pick between a men's, women's, boys', or girls' shoe. After that, they can pick a particular category of shoe, such as athletic, casual, etc. and then pick a subcategory, such as running shoe or cleat. From there, the user has several features they can choose from a multiple choice list. These features include the size of the shoe, the width of the shoe, the side, back, and front image, the shoe style, tongue style, the shoe sole style, the types of laces, and the coloring of the shoe.

The way the custom shoe information is stored is in the **Custom_Shoe** entity. This entity is a sub-entity of the shoe entity, containing several additional information for the customize process. This includes a side, back, and front image labels, a tongue style, a shoe style, a sole style, a laces style. It also has a *custom_shoe_id* for the unique key. In our database, we will store every possible combination of shoes we will allow the user to make as a **Custom_Shoe** entity. Then, during the creation process, as the user selects new attributes, different custom shoes will be displayed on screen, based on the default selections and the user selections.

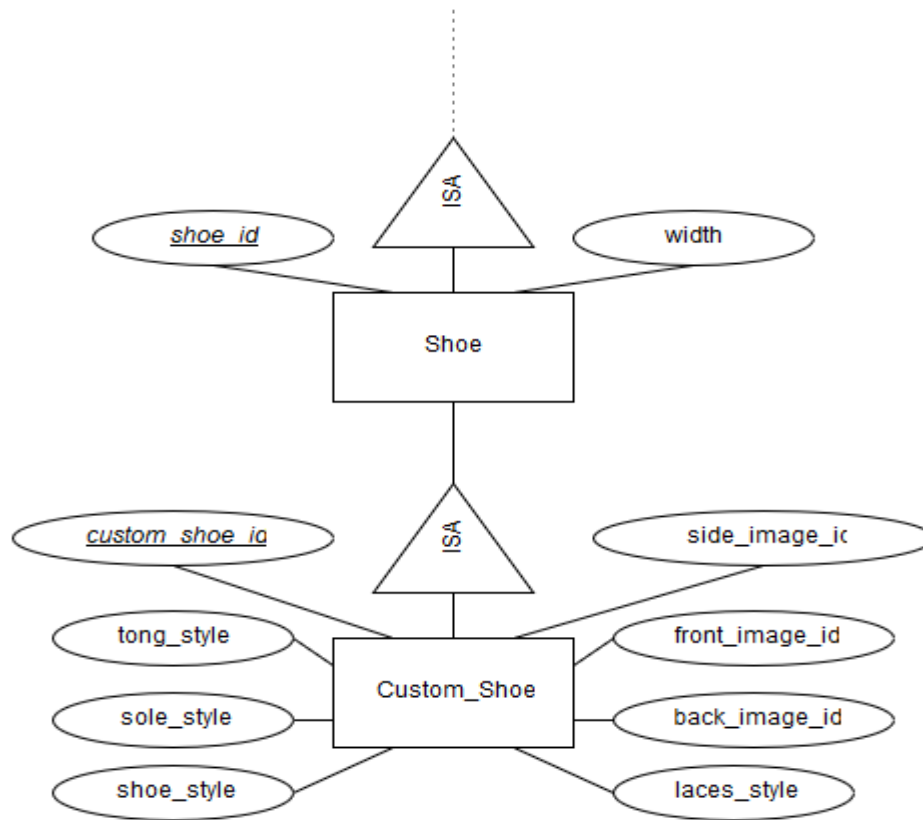


Figure 15 - Custom_Shoe ER Diagram

2.13. Customer Service

We are providing users a way to contact us. This bonus feature allows us to connect us with the users. Users can email or call us. We also provide a FAQ page that helps users navigate better. This page also helps users get answers to common questions. Customer service options will be located at the bottom of every page. This feature also helps us get feedback about our services, which provides a scope for improvement.

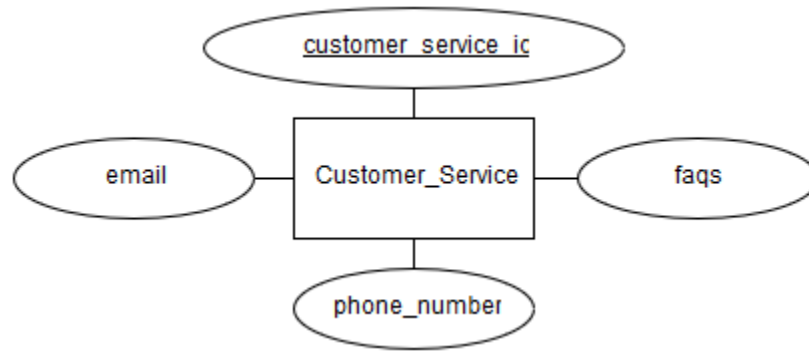


Figure 16 - Customer_Service ER Diagram

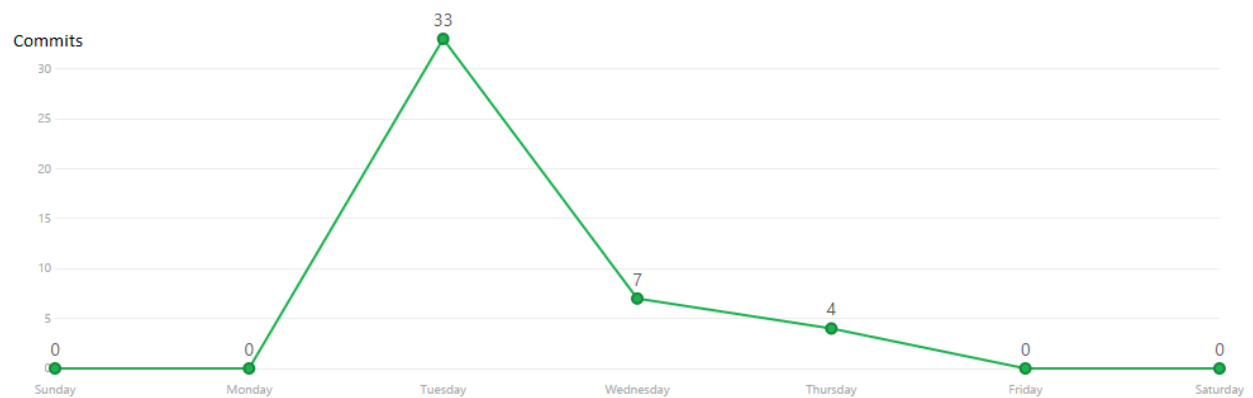
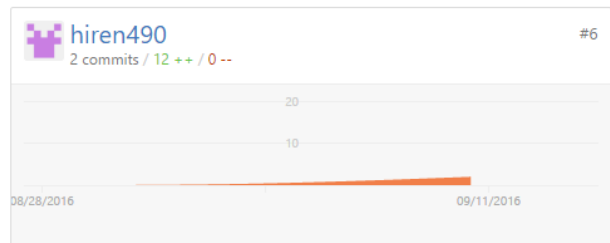
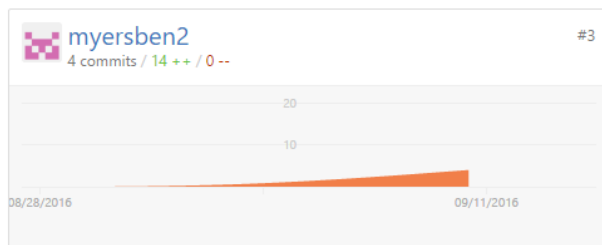
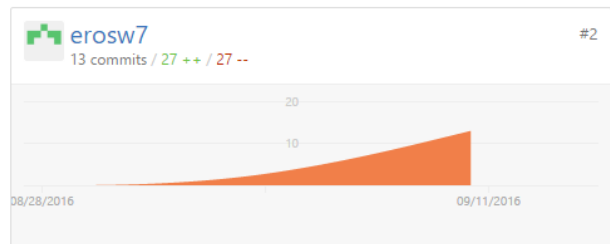
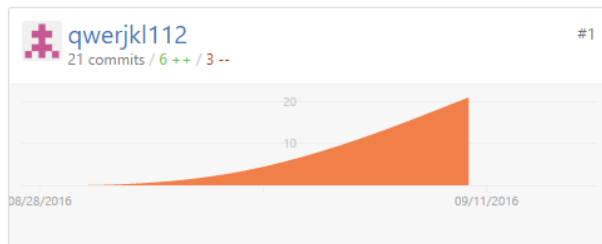
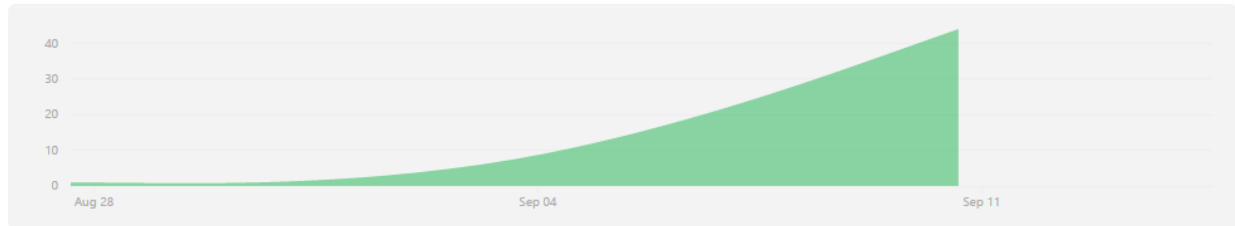
3. CONCLUSION

There were many options to choose from when selecting a database. After our first meeting we gathered ideas from everyone and ended with a shoe retailers' database. The shoe industry is enormous and there were many features that we wanted to talk about besides the given ones. Everyone did their research on ideas and how we would incorporate the given features. The bonus features we selected were custom shoes and customer service. We assigned each person separate features, where each would do research on their specific topic and write a rough draft for analysis, we discussed how each person should implement certain ideas pertaining to their feature. Next, we broke up into 2 teams, we had 3 on editing, while the rest on presentation. The biggest challenge we faced was making sure the information lined up perfectly between each topic because some features are closely connected to each other. Our main goal was making sure we developed a database that was user-friendly, yet offered complex features. Overall, phase one was a success where we implemented the design specifications according to the criteria.

APPENDIX A – GITHUB GRAPHS

Aug 28, 2016 – Sep 15, 2016

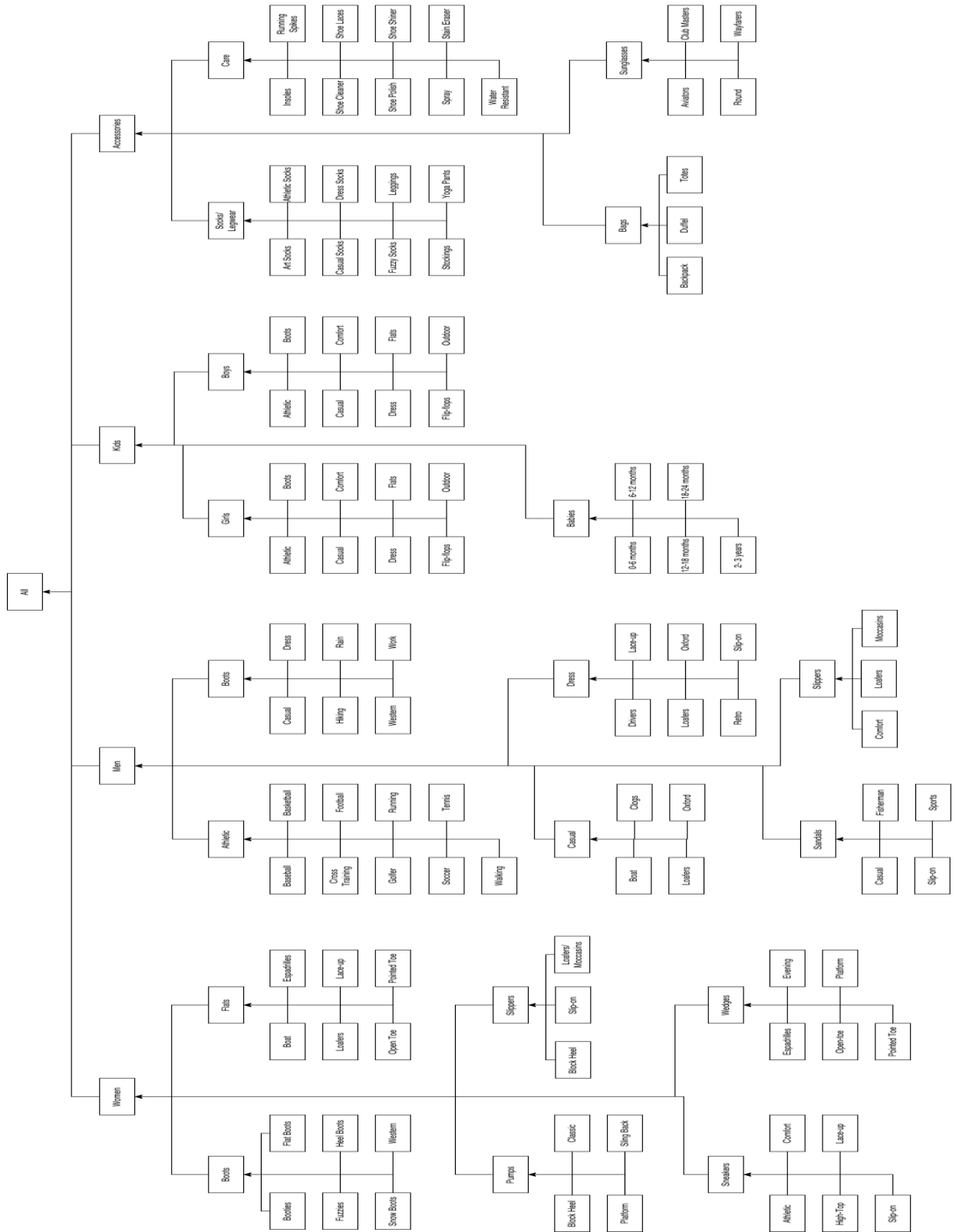
Contributions to master, excluding merge commits



APPENDIX B – ASANA SUMMARY

Task ID	Created At	Completed At	Last Modified	Name	Assignee	Due Date	Tags	Notes	Projects
182114864349546	9/15/2016	9/15/2016	9/15/2016	ER Diagram for Suppliers	Dishti Gurnani				Cmpsc 431 Project
182114864349551	9/15/2016	9/15/2016	9/15/2016	Citations	Eros Wang				Cmpsc 431 Project
182114864349554	9/15/2016		9/15/2016	Formatting	Benjamin Myers				Cmpsc 431 Project
182118051116087	9/15/2016		9/15/2016	Final Review	Hiren Patel				Cmpsc 431 Project
182118051116082	9/15/2016		9/15/2016	Final Review	Frank Guo				Cmpsc 431 Project
182114864349535	9/15/2016	9/15/2016	9/15/2016	Editing Team Meeting	Hiren Patel	9/14/2016			Cmpsc 431 Project
182115522284327	9/15/2016	9/15/2016	9/15/2016	Editing Team Meeting	Benjamin Myers	9/14/2016			Cmpsc 431 Project
182115522284332	9/15/2016	9/15/2016	9/15/2016	Editing Team Meeting	Dishti Gurnani	9/14/2016			Cmpsc 431 Project
180961628492192	9/15/2016		9/15/2016	Google slides	Frank Guo	9/15/2016		https://docs.google.com/presentation/d/15o3gSL1USxKx2hmKxa62fS1i6Bz7wc03y4_KV	Cmpsc 431 Project
								Presenting Team: Dishiti, Eros, Frank Editing Team: Jeff, Ben, Hiren TASK: Complete all Analysis onto Google Doc by Tuesday night!!!!(see below task) Next Meeting: 1. Review the Project Analysis 2. Come together to finalize Introduction and Conclusion and be ready to submit. 3. Editing Team and Presenting Team come together to decide how to present	
180925438009031	9/13/2016	9/15/2016	9/15/2016	Fourth Team Meeting					Cmpsc 431 Project
180925433959306	9/13/2016		9/15/2016	Google Docs	Frank Guo	9/15/2016		https://docs.google.com/document/d/1qITkzf9JGVTQUGitwNClqA2NM6yHAEIKmY3H	Cmpsc 431 Project
179623733913164	9/10/2016	9/14/2016	9/14/2016	Individual Analysis Ben	Benjamin Myers	9/13/2016		Sale Items, Custom Shoes	Cmpsc 431 Project
179623733913172	9/10/2016	9/15/2016	9/15/2016	Individual Analysis Dishiti	Dishti Gurnani	9/13/2016		Suppliers, Register Users, Membership	Cmpsc 431 Project
179623733913175	9/10/2016	9/15/2016	9/15/2016	Individual Analysis Eros	Eros Wang	9/13/2016		Rating, Searching	Cmpsc 431 Project
179623733913178	9/10/2016	9/15/2016	9/15/2016	Individual Analysis Hiren	Hiren Patel	9/13/2016		Sale Bidding, Bidding	Cmpsc 431 Project
179623733913181	9/10/2016	9/15/2016	9/15/2016	Individual Analysis Frank	Frank Guo	9/13/2016		OSR, Delivery	Cmpsc 431 Project
								Hey guys, the task is assigned based on their difficulty and relationship to one another. Requirements: - Take up at least one page (see professor's template) - Including a minimum of one source - Create an ER diagram if you can, else create tables or diagrams that you think can bet SEE UPDATED TASKS	
179623733913184	9/10/2016	9/15/2016	9/15/2016	Requirements on Individual Analysis				Categories, browsing	Cmpsc 431 Project
179623733913168	9/10/2016	9/15/2016	9/15/2016	Individual Analysis Jeff	Jeff Li				Cmpsc 431 Project
179623282417958	9/10/2016	9/15/2016	9/15/2016	Third Team Meeting		9/10/2016		Assign tasks to everyone to do by Monday Fixing any conflicting information	Cmpsc 431 Project
178970851346564	9/8/2016	9/15/2016	9/15/2016	Classification Tree					Cmpsc 431 Project
177362580084448	9/6/2016	9/15/2016	9/15/2016	Second Team Meeting	Benjamin Myers	9/8/2016		6:00 pm Library W102A	Cmpsc 431 Project
177368086282441	9/6/2016	9/15/2016	9/15/2016	Conceptual Ideas Phase 1		9/6/2016			Cmpsc 431 Project
177362580084442	9/6/2016	9/15/2016	9/15/2016	First Team Meeting	Benjamin Myers	9/6/2016		Starts at 6:00pm at East Paterno 307	Cmpsc 431 Project
173381809847679	8/29/2016		9/10/2016	Submit Written Proposal		9/16/2016			Cmpsc 431 Project
173381809847664	8/29/2016		8/29/2016	Present Proposal		9/16/2016			Cmpsc 431 Project

APPENDIX C- CATAGORY TREE



WORKS CITED

1. "Amazon." Amazon. N.p., n.d. Web. 13 Sept. 2016.
2. Caleres. *Shoes, boots, sandals - famous footwear online*. 2016. Web. 15 Sept. 2016.
3. "EBay." *Electronics, Cars, Fashion, Collectibles, Coupons and More*. N.p., n.d. Web. 15 Sept. 2016.
4. "Feedback Scores, Stars, and Your Reputation." Feedback Scores, Stars, and Your Reputation. N.p., n.d. Web. 13 Sept. 2016.
5. Harden, Seth. "Footwear Industry Statistics." Statistic Brain. N.p., 03 Sept. 2016. Web. 15 Sept. 2016.
6. "Official Store. Nike.com." N.p., n.d. Web. 15 Sept. 2016.
7. "Online Shoes, Clothing, Free Shipping and Returns| Zappos.com." Online Shoes, Clothing, Free Shipping and Returns| Zappos.com. N.p., n.d. Web. 15 Sept. 2016.
8. "Shoes, Boots, Sandals, Handbags, Free Shipping! | DSW.com." *Shoes, Boots, Sandals, Handbags, Free Shipping! | DSW.com*. N.p., n.d. Web. 15 Sept. 2016.
9. "The Purpose of Weekly Status Reports." Reportingadvice. N.p., 2013. Web. 14 Sept. 2016.____
10. "United States." UPS Developer Kit. N.p., n.d. Web. 14 Sept. 2016.
11. Women's shoes, boots, Handbags & clothing online. 2010. Web. 15 Sept. 2016.