

Catalog

TobiasBird, Victor Campus, and Cooper Johnk
access.engr.oregonstate.edu:13375

1. Introduction

The catalog website is a user friendly store for anyone to use. The website controls all of the buying and selling so there is no user interaction. The user will need a username and password to make any actions on the website. With the connection to the database, the users information will be able to be saved and checked when buying or selling.

2. Detailed Functionality & Requirements

The website will record the limited items posting date and time to record when the item will be deleted 24 hours later.

The website allows any item to be bought or sold for any price, there is no checks on the items or prices. The users are in charge.

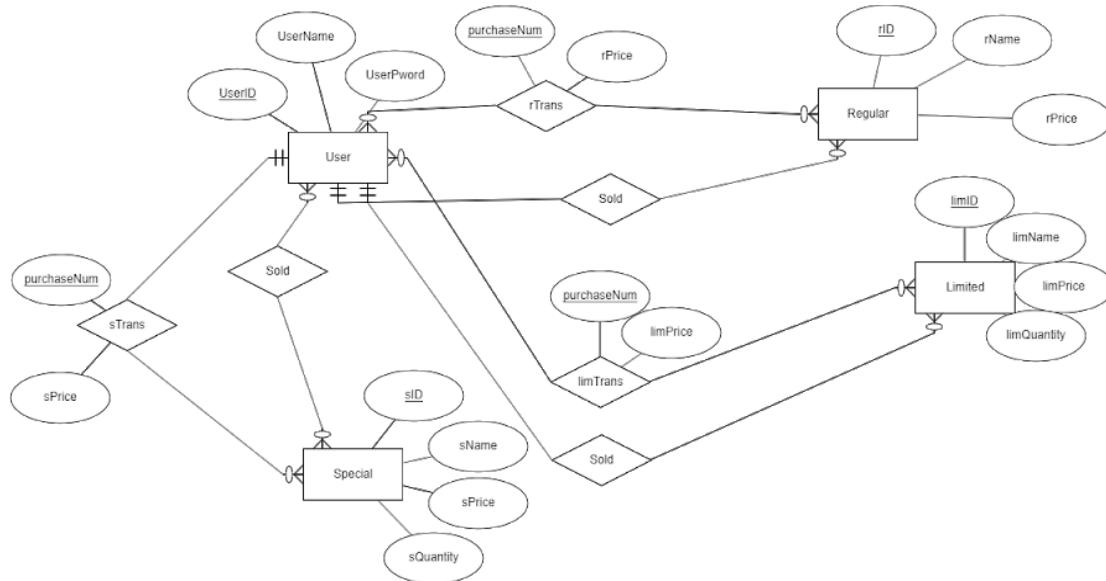
The website purchases the items and then sells it to the users so there buyers and sellers won't need to communicate.

There will be 3 lists that the user can look through to purchase an item, the list will either be regular, limited, or special.

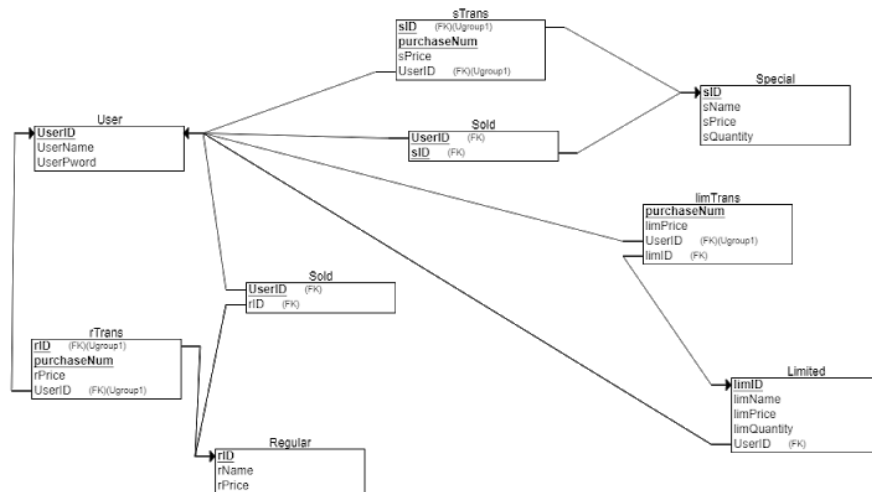
.

3. Database Design

- **ER Diagram of Database**































- **Relational Schema**



- **Database Tables**




Cs340_User:

+ Options					UserID	UserName	UserPword
							
<input type="checkbox"/>					0	GonnerS	asdf
<input type="checkbox"/>					1	DukasD	1234
<input type="checkbox"/>					2	BodhiD	qwer
<input type="checkbox"/>					3	UnknownG	asdf
<input type="checkbox"/>					4	BarkerB	1234
<input type="checkbox"/>					5	SandE	cat
<input type="checkbox"/>					6	Dandito	dog
<input type="checkbox"/>					7	Beema	bee
<input type="checkbox"/>					8	YanSky	1234
<input type="checkbox"/>					9	DeepDish	cat

Cs340_sTrans (Special Transactions):

+ Options					sPrice	purchaseNum	UserID	sID
								
<input type="checkbox"/>					15	0	6	0
<input type="checkbox"/>					15	1	2	0
<input type="checkbox"/>					50	0	4	2
<input type="checkbox"/>					5	0	2	3
<input type="checkbox"/>					30	0	3	4
<input type="checkbox"/>					5	0	1	5
<input type="checkbox"/>					5	1	7	5
<input type="checkbox"/>					44	0	7	6
<input type="checkbox"/>					12	0	8	9
<input type="checkbox"/>					12	1	6	9

Cs340_Special:

+ Options					sID	sName	sPrice	sQuantity	UserID
									
<input type="checkbox"/>					0	Monopoly	15	4	4
<input type="checkbox"/>					1	Dog Collar	7	23	3
<input type="checkbox"/>					2	Duffel Bag	50	1	0
<input type="checkbox"/>					3	KitKat	5	21	9
<input type="checkbox"/>					4	Spikeball Set	30	2	5
<input type="checkbox"/>					5	Car door	5	1	8
<input type="checkbox"/>					6	Doll	44	1	7
<input type="checkbox"/>					7	Rabbits Foot	47	2	2
<input type="checkbox"/>					8	Magical Wand	200	1	1
<input type="checkbox"/>					9	Magic 8 Ball	12	5	8

Cs340_rTrans (Regular Transaction):

+ Options

				UserID	rPrice	purchaseNum	rID
<input type="checkbox"/>	Edit	Copy	Delete	6	20	0	0
<input type="checkbox"/>	Edit	Copy	Delete	5	2	0	1
<input type="checkbox"/>	Edit	Copy	Delete	8	10	0	3
<input type="checkbox"/>	Edit	Copy	Delete	9	5	0	4
<input type="checkbox"/>	Edit	Copy	Delete	5	4	0	5
<input type="checkbox"/>	Edit	Copy	Delete	4	10	0	8
<input type="checkbox"/>	Edit	Copy	Delete	4	10	0	9
<input type="checkbox"/>	Edit	Copy	Delete	3	2	1	1
<input type="checkbox"/>	Edit	Copy	Delete	2	5	1	4
<input type="checkbox"/>	Edit	Copy	Delete	0	10	1	8

Cs340_Regular:

+ Options

				rID	rName	rPrice
<input type="checkbox"/>	Edit	Copy	Delete	0	Wooden Chair	20
<input type="checkbox"/>	Edit	Copy	Delete	1	Toy soldier	2
<input type="checkbox"/>	Edit	Copy	Delete	2	BB Gun	31
<input type="checkbox"/>	Edit	Copy	Delete	3	Water bottle	10
<input type="checkbox"/>	Edit	Copy	Delete	4	Pokemon Cards	5
<input type="checkbox"/>	Edit	Copy	Delete	5	Used Toothbrush	4
<input type="checkbox"/>	Edit	Copy	Delete	6	Black Paint	14
<input type="checkbox"/>	Edit	Copy	Delete	7	Orange Paint	15
<input type="checkbox"/>	Edit	Copy	Delete	8	Soccer Ball	10
<input type="checkbox"/>	Edit	Copy	Delete	9	Basketball	10

v

CREATE TRIGGER Reg_Purchase BEFORE INSERT ON Cs340_rTrans

FOR EACH ROW

BEGIN

IF new.UserName != old.UserName OR new.Pword != old.Pword

THEN

SETMESSAGE_TEXT = 'USERNAME OR PASSWORD IS INCORRECT';

END IF;

END

v

INSERT INTO Cs340_Regular(rID, rName, rPrice) VALUES (regID, data[4], data[5])

v

```
SELECT * FROM Cs340_Regular
```

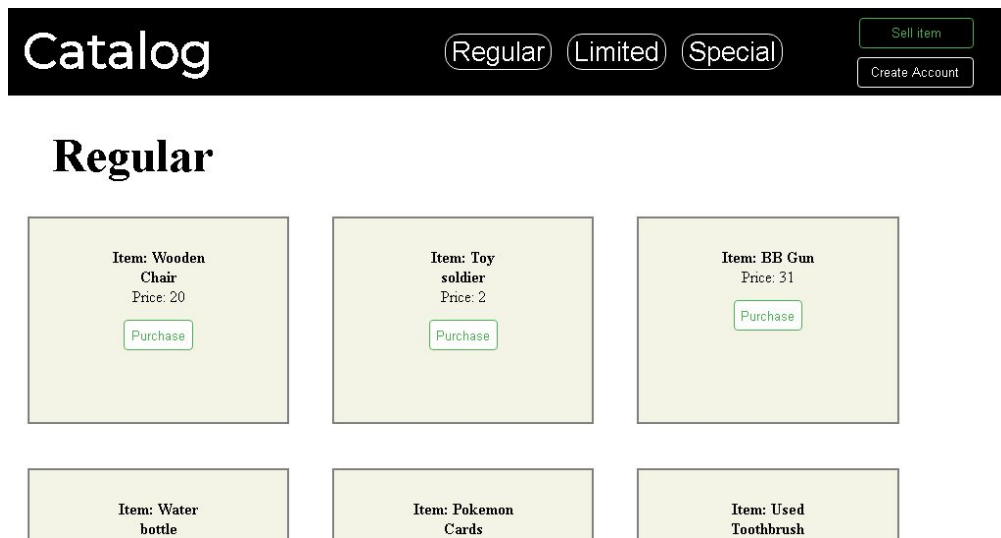
4. Website Design

The category website is broken up into two pages. One is the home page that shows either the regular, limited, or special items for sale, and the other page displays the item chosen and allows the user to buy.

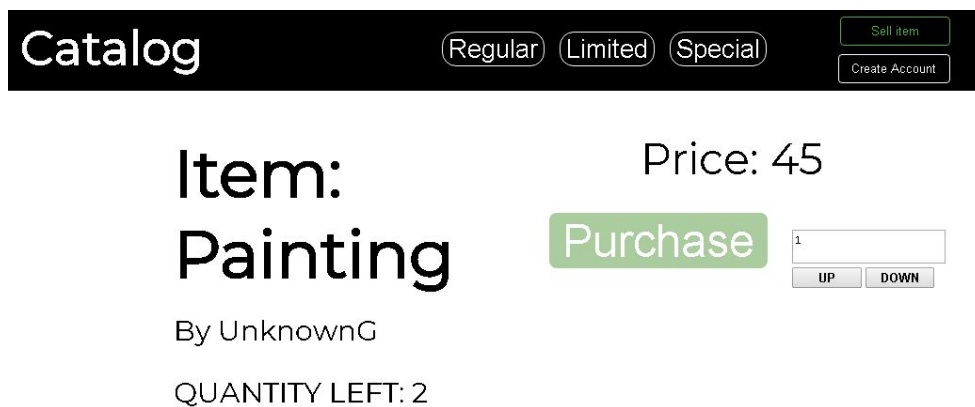
- **Website Layout**

The home page has a list of items for sale and each of the items has a purchase button on it to redirect the user to the individual item page.

- **User Interface**



^ main page: browse for regular, limited, and special items, create a listing, or log in.



^ item page: select an amount of an item to buy, and view specifics about the item

- ***User Manual or Help page***

5. Application Implementation

- We use handlebars to put the page together. There are two css files designing each page. The .js files were used to connect to the database and reload the web page with the new information.
- SQL queries were done both back and forth from the user to the server. Sending data from the server to the user was simple, since it only took a query within the server javascript, but sending from the user to the server was a lot tougher. Eventually, we settled on sending to the server by using the URL of the page and catching arguments on the server end, which could then be sent to SQL databases by the server javascript.

6. Evaluation

We tested our application mostly by running node servers on the engr servers to test every change we made. Once we completed a part of the program, we aimed to push that to the master branch or another branch that group members could access and update their projects with.

7. Future Work & Lessons Learned

It was challenging to get the sql code to work with the .js files because we normally worked on the phpMyAdmin but actually implementing it in web code was a different challenge. It was also challenging to test application due to having to copy local files to ssh to run it on server. Our team were always communicating to make sure no one was stuck on an issue. We also worked together to research and look at template to figure out how to use sql code. Next time we would work harder to finish the html design so we have more time to work on the sql code.

Appendix – Team Report

Tobias Bird: Worked on html formatting and sql connection

Victor Campa: Worked on html formatting and sql connection

Cooper Johnk: Worked on sql connection