**The League of Legends Store**

By Project\_X

### Members:

### Jiayi Liang, Naiqiang Lin, Berty Ruan, Shaohua Wang, Yusheng Wang

### Table of Contents

1. Introduction

# 

# Introduction

The League of Legends Store is a startup company that has partnered with Riot Games to deliver the best real world and virtual gaming experiences to the millions of League of Legends fans worldwide. The opening day for this store has been anticipated for months, and the new store has been hard at work to fill its warehouse with quality League of Legend cosplay items, and notify potential suppliers to utilize their store. A month before the release date, the founders convened for an emergency meeting. They had met with an independent contractor beforehand and were informed that their current item database was designed poorly and was deemed unusable. The contractor informed them that they needed to redesign their database before the opening day.

As a result of the emergency meeting, the founders have contacted a group of Penn State’s CMPSC 431W students to design a better database in order to sell their League of Legends items. The contractor had given the students a three step plan to deliver a quality product which is outlined below and then described in detail in the rest of this document.

First the students need to analyze the League of Legends Store business requirements and construct the preliminary database design using the Entity Relationship Model paradigm. Next the students need to decide which database management system to use, create a logical database design, refine and normalize the initial design and populate the database. Afterwards, the students need to write a set of representative transactions to access the database. When all three project phases are completed, then the students may begin implementing the League of Legends Store website, test their code, and finally demonstrate their product to the founding team. If all goes well, the CMPSC 431W students will finish on time and satisfy all the founding members’ database requirements. The success of the League of Legends Store is contingent on a well designed and implemented database.

# 2. Conceptual Database Design

## League of Legends store needs multiple functions to keep working normally. As our design, the store mainly separates into 13 functions, Sale Item, Searching, Biding and so on . All of functions should contain its own attributes and some of the attributes should relate to each other. And functions give the description of their uses of ER model, ER model follows the basic functions to form the tables’ relationship for the database.

## 2.1. Requirement Analysis

***Sale Item -*** Items sold in the League of Legends (LoL) store must be usable in the LoL game environment or provide enhancements to the LoL gaming experience. Before an item can be uploaded onto the LoL database and store, the supplier must include basic information about their product and customize how they want to sell their product. Both requirements will be described fully in this section.

A valid LoL item must be either a physical object that incorporates LoL gaming themes, or compatible with the LoL gaming engine, and is usable either in a LoL user’s profile and account or in at least one of the four LoL’s gaming modes, Summoner’s Rift, the Twisted Treeline, Howling Abyss, or the Crystal Star. All items need not be endorsed by the LoL community. Both types of items may be made by third-properties manufactures and unendorsed by the LoL community.

When selling an item on the LoL store, we ask the suppliers to provide rudimentary descriptions of their product and other information. All items must list the name of the product and supplier. A source may be a company, individual or an LoL affiliated organization. The supplier needs to provide a short description and location of shipment. The type of item must be either listed as physical or virtual. When the item has been validated by the LoL store, it will be automatically assigned a unique identifier.

The seller must also specify to sell the item by listed price or by auction. If by listed price, the static price in dollars and the amount of stock available should be associated with the item. If their is no more stock for an item, it will be removed from the LoL store. If by auction, the seller must provide the reserve price, hidden from LoL customers, and the start and end date of the auction. LoL customer may bid on auctioned items between the start and end dates. Otherwise those items will not be visible on the LoL store. In addition the LoL store will automatically cancel auctions that pass their end dates if the highest bid price is lower than the reserve price. All auctioned items must be sold individually.

An example of the required information needed from the supplier for items sold by listed price and auction is provided below.

**Table 1 - Required Information for Items with Listed Prices**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item Name** | **Supplier name** | **Supplier’s Location** | **Descript.** | **Type** | **Price ($)** | **Stock** |
| Big Sword | Grapefruit | NYC, USA | Very Useful | Virtual | 9.99 | 7 |

**Table 2 - Required Information for Items Sold by Auction**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Item Name** | **Supplier Name** | **Supplier’s Location** | **Descript.** | **Type** | **Reserve Price ($)** | **Start Date** | **End Date** |
| Guardian Angel | LoL | Chicago, USA | Durable | Physical | 1,899 | Dec. 1, 2015 | Dec. 1, 2016 |

***Categories -*** All items listed in the LoL store will be categorized using a predefined classification tree, and can be found in the store through the tree. Assigning a category will be the responsibility of the seller. The properties of the classification tree will be described here.

Categories are described as nodes and represents a set of related items, and nodes are related to each other as parents or children. Parent nodes are strict supersets of the current node while chilren nodes are strict subsets of the current node. The root of the tree is labeled “All” to represent all items. The number of children for each node can reach up to no more than fifteen, and the height of the classification tree will be at least 10 nodes deep. Each node is given a descriptive name to help the suppliers and customers navigate through the store. Multiple nodes of the same descriptive name may exist, and items may be placed into multiple nodes. Items added by suppliers must be assigned to a leaf of the classification tree. Parent nodes of the leaf and the leaf itself will have access to the given item. An example classification tree is given below.

1. All -> Physical -> Cosplay Item -> Summoner’s Rift -> Consumable -> Elixer of Wrath
2. All -> Virtual -> In Game -> Summoner’s Rift -> Illegal -> Over Powered -> Bonus Attack Speed -> Basic Stats -> Champion Selection

***Suppliers -*** All suppliers must register themselves onto the LoL store before uploading their products. Supplier’s information will be stored in the store’s currently unsecured personal databases. For all new suppliers on the LoL store, their type of supplier, individual or organization, must be specified. If it’s an individual, the seller’s name, permanent home address, cellphone number, home phone number, email address, annual revenue, personal identification number (i.e. social security number, driver’s license, and etc.), bank account number must be provided. If the seller is an organization, its organization’s name, main address, and annual revenue; point of contact’s name, phone number, email address, and hours of availability; and the bank account’s routing number must all be provided.

***Registered Users -*** In this part, a user will need to agree with the terms that we provide, and then sign up with his email address and set up the password. The email address will be the user ID or the username later.

Then one will go to the personal information part, which includes name, address (street, city, state, and zip), credit card information, and emergency contact phone number.

If a user want to sell items he will need to give more information like age, gender, SSN, state ID, photo of himself, company information, and annual income.

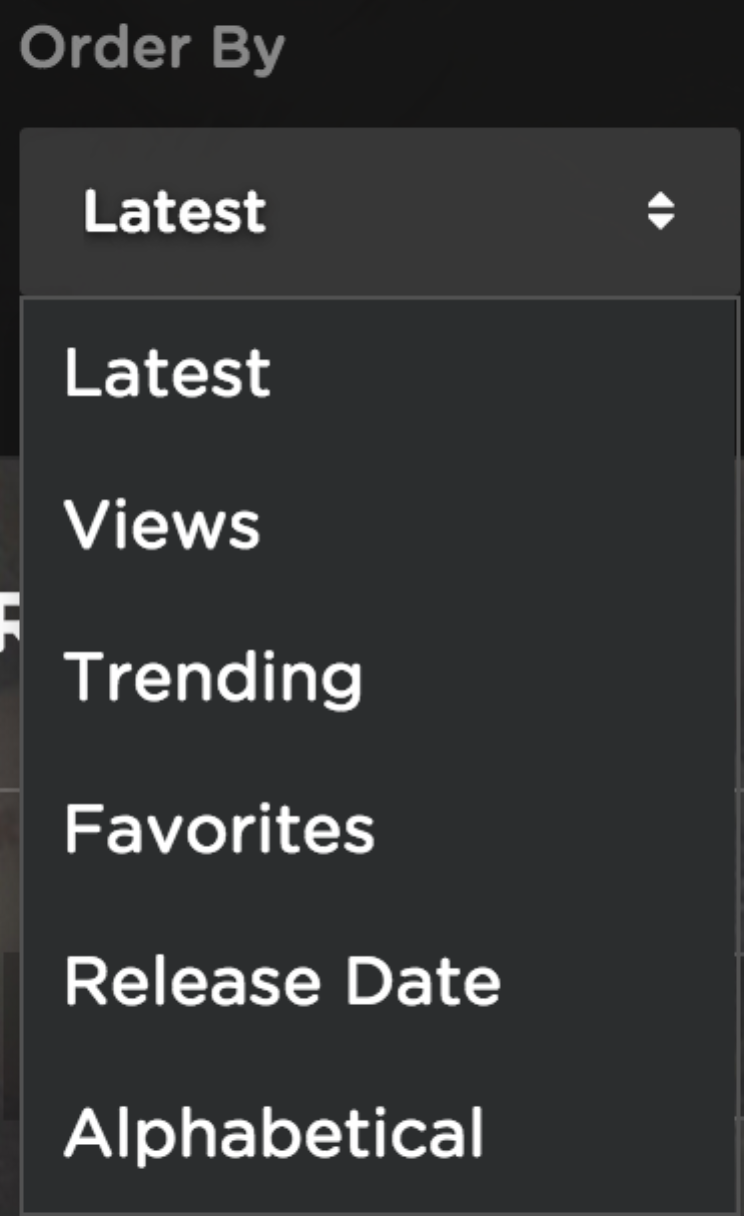
***Rating -*** We are going to make a rating system for both customers and sellers. Once a item is paid and sold out, the money will be kept buy the system for 7 days until the customer has received and command on the quality of the product, the satisfaction, the serve attitude of the seller, and the service of the express by rating them from 1 star to 5 stars and write down a brief explanation with pictures.

Once a command is done, the seller will have 2 days to negotiate with the customer about the rating and improve their service or change the product to have a batter rating from the customer. After that, the seller will be able to rate on the customer from 1 to 5.

Besides, one can also rate the satisfaction of the purchasing, and give some brief comments on the system that we provide.

***Browsing*** - When the user clicks on a category, he will see all the items under that tree. There will be 30 items on each page.

After that, one can also order the revealed items by using different key words like sold most, commanded most, highest rating stars, trending, favorites, date, alphabetical, price up, and price down. Just like figure 2.1 below. Once selected, the system will sort the items depend on that key word.



**Figure 2.1**

***Searching -*** Users can search for the item by keywords, if the user knows which category the item the user is looking for belongs to, the user can search by category, Such like the figure 2.2 indicates.

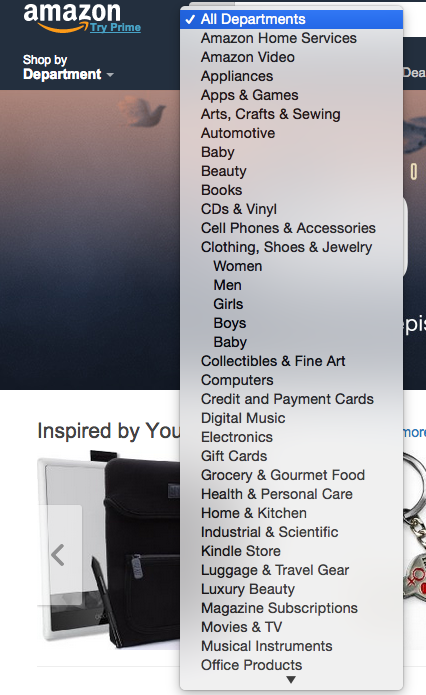


Figure 2.2

We could use MapReduce to calculate the rank for each item: The initial rank is the same to every item, then we record how many visit each item gets in one day, and how the stream flows. Using MapReduce algorithm we can update the rank for each item every day and by doing this we can provide a better search result by presenting the item with higher rank in the front.

However, it is optional to implement the MapReduce algorithm, it is also feasible to arrange the order of the result by the frequency of a item is being visited in a day.

## *Sale -* We can use a table to record which credit card is charged when a certain item is sold, the table should at least have three columns: item id, credit card number, price. Once the seller confirms that the credit card information is valid, a new row is created in the table meaning that the transaction is finished and this row will be kept in the table for at least six months.

The credit card information is retrieved from the buyer’s account profile table, and this information is processed in the backend to check if the card number length is correct, if the expiration date is later than half a month from the day the buyer purchases the item.

The supplier has the option whether to put the item in auction function of in the selling list. If the item is in both auction list and selling list, then the sale price is the reserve price.

***Bidding -*** To make every bid valid, we need a function to check that the bidder is not seller and the bid happens between the time the item is registered for auction and the end of the auction. When one wants to bid, he or she can bid at any amount higher or equal to $2. Only meeting these criterion can a bid be valid.

After the auction ends, the system sends message to every bidder and seller that who is the winner of the auction and the how much he or she bid. The winner will also get a message that confirms the credit card information, if the user confirms the message, the auction succeeds, the status of the item changes to “sold” and the user is charged. However, if the user refused to pay the price, the auction fails, the bidder gets a negative comment for the failure of the auction and the supplier can choose register the item for another auction.

***Order and rank reports:*** The Summary report will provide an overview of sale information. Every week, a report is generated to summaries the ordering and sales information based on categories of items just like Table 3 shown below.

**Table 3 - Sample table for category of summaries**

|  |  |  |  |
| --- | --- | --- | --- |
| category | average | total amount | total price |
| champions | 23 | 2 | 46 |
| rune pages | XXXX | XXXX | XXXX |

Also, for individuals, a similar report of sale information is generated everyday, shown Table 4.

**Table 4 - Sample table of Sale information**

|  |  |
| --- | --- |
| category | cost |
| champions | 32 |
| rune pages | 43 |

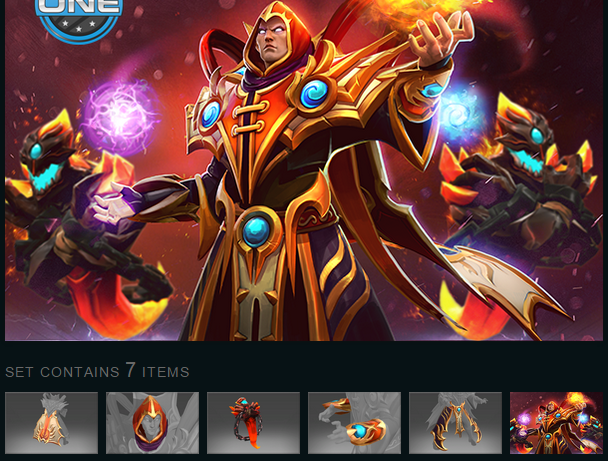
In addition, for different category, ranking reports of sale items are provided for clients to see which one is the most popular in the latest week, as shown Table 5.

**Table5 - weapon sale rank**

|  |  |  |
| --- | --- | --- |
| rank | item | amount sold |
| 1 | weapon 1 | 4543 |
| 2 | weapon 2 | 2333 |

From the reports, the developer will know what the favors of players are and based on the reports, the store can improve its business.

***Recommendation and Gift:*** Every week, based on information of the heroes played in the last seven days, the items of heroes that you play the most frequently are recommended for you figure 2.3. Or you can select a specific hero to see what items of the hero are popular in the past week.



**Figure 2.3**

In the gift system, anyone can send an item as a gift to others. Highly liquid market can attract more and more players to join in. Content Gifting includes skins, champions, rune pages, icons, ward skins, Mystery.

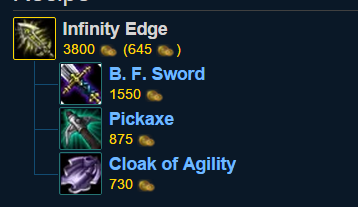


**Figure 2.4**

In addition, a trading system allows people to sell his item to the store.

***Delivery:***For physical items, the delivery system shows the order id, tracking number. And also, the system can record the delivery status to ensure the success of the order. For virtual items, it seems to be simple that the delivery is always completed, since it is delivered immediately.

***Combination System:***Since we are doing the League of Legends Store, it is necessary to have a combine system which combine multiple sub-item into a complete item. For example, A weapon named infinity edge needs a BF Sword, a Cloak of Agility and a Pickaxe to form (Figure 2.5).



**Figure 2.5**

Not even a weapon item can be combined, the skins of a champion can be combined. While our Skins product separate each skin into multiple parts, it is possible to form a customize skin that skin parts are from different champions(Figured 2.6). In this function, we will design a combine option to buyers who owns the previous items that fits the requirement to form a complete item. This function depends on buyers’ requests.



**Figure 2.6**

Runes system is another important parts in League of Legends (Figure 2.7). As first design, it should able to ask if buyer would like to combine randomly or directly, direct combination give more space for buyer to choose while random combination cost less. Whether it is a item combination, skin combination, or rune combination, the function should always look into the buyer’s account and check the prerequisites for a complete items in the database. While the combine request successfully made, the data of this account and selling history should recorded, the delivery function should get a message to deliver the item to buyer.



**Figure 2.7**

***News, Deal and Follow Ups:*** It is a common scene that every company will provide follow up news or sale deals to the buyer who bought products and provided their emails. And we think this will be a good way for our products as well. This follow-up function will sent the information of our newest deals and our activists to the buyers who would like to received the information. It should check the emails of each account which confirm sending information in the database.

## 2.. Entity Relationship models

# 3. Appendices

Appendices are labeled with successive letters of the alphabet, the first being Appendix A. One appendix should include the graphs from your team’s GitHub repository, and the other one be a summary of your team’s Asana project directly exported from Asana.

# 4. Conclusion

This concludes the League of Legends store requirements for Phase 1 of the project. The CMPSC 431W students have fully analyzed the requirements for implementing a database for the store, and then translated the requirements to an Entity Relationship model. By completeting these two first, we have set a solid foundation to create a versatile database that the clients have requested.

The next phase is to decide which database management system to use, create a logical database design, refine and normalize the inital design and populate the database with the League of Legends store items.