**QuickSell eCommerce Portal and Product Database**

**COMP 421-Database Systems**

Rishabh Tandon – rishabh.tandon@mail.mcgill.ca Deepak Sharma – deepak.sharma@mail.mcgill.ca Abhishek Khanna – abhishek.khanna@mail.mcgill.ca

**Table of Contents**

**Introduction 3**

**Entities and their attributes 4**

**Relationships 6**

**Application Description 6**

**Preliminary Calculations 7**

**Functional Examples 7**

**E/R Model 8**

**Entities 9**

**Appendix 10**

**Introduction**

The main purpose of our application is to create an online ecommerce platform to supply the growing demand for online shopping. Our platform will provide a convenient one-click checkout option to registered users to get their favourite item without getting into complications. The User Interface will be extremely simple to understand and use. Our team expert in Human Computer Interaction has been giving his best to design the perfect User Interface.

The website will act as a bridge between retailers and users all over the world. A registered retailer will be able to submit products to multiple categories and earn good rating on their products which could then be shown on the homepage of the website. The users can purchase items of their choice and rate retailers and/or products. The users can also make a wish list of their favourite items, which they can buy in the future. There will also be an option on checkout to send the item as a gift.

We will be using Ruby on Rails to develop the application and will be using some pre-developed modules to make the development process simple. This requirement analysis document gives a detailed description of the architecture of the application. For any questions please do not hesitate contact us on the company emails mentioned on the cover page.

**Entities and their attributes**

The following entities will be stored in tables of a relational database:

1. **Customers** - A customer is the main user of our ecommerce application. Every customer has an account registered with us with their personal details such as name, address, contact number stored to identify them in the real world. Information such as the email and password information stored as a hash (SHA 512) is used to identify them virtually. A customer buys products on our website by clicking the ‘Buy’ button on every webpage of a particular product. We also provide the facility of storing their Search history. A customer can add products to their wishlist, which can be gifted to other addresses. To create a seamless experience for the customer, our company stores some rudimentary information regarding the customers buying habits and style such as the average cost of the products they have bought, their gender and age. This allows us to provide them recommendations so that customers do not have to spend too much time on our website to buy a product.
2. **Retailers** - A Retailer is someone who is using the ecommerce platform to sell their products. A Retailer can sell products of multiple categories. Each Retailer has a rating, which will be given to him by the customer each time their product is purchased. The Retailer needs to give information like Name, Address, email, Customer Support Number, Categories of Products they sell, website for their store/company and Average Delivery Time for them to deliver the product to the customer. The email will act as the username and will be needed by the retailer to login to submit a new product/update inventory for existing products.
3. **Products** - A product is defined as the item of sale on the online platform. A product can be divided into various categories such as Books, Electronics etc. A product can be associated with more than one category as well. A product can only be submitted by a registered Retailer and would require fields such as Image, Product Name, Product Description, Retail Price and would need to be linked with the Retailer information.
4. **Products** **Sold -** Products sold is the main inventory of all products sold on the ecommerce platform. The main purpose of this entity is to keep track of all orders placed on the web application and will consist of Product ID, Address Shipped to, Date Ordered, Delivery Date and each entry will be uniquely identified by the order number which will be generated each time a new order is placed.
5. **Books** - A sub-category of the products table. Creating child tables of the products table gives us the flexibility to specify product information that is more specific and appropriate to the product category. For example, Having a single product table narrows the information we can specify for every product. A book and a video game are essentially two completely different products with different characteristic and essential features that need to be displayed to the customer. The books sub-category stores information such as the ISBN, Author, Date of publication.
6. **Music** - A sub-category of the products table. This table stores extra information regarding all products that can be identified as music labels, cds, albums and all other related merchandise. Every tuple of the table stores information such as the artist name, music duration.
7. **Electronics** -   A sub-category of the products table. This table stores extra information regarding all products that can be identified as music labels, cds, albums and all other related merchandise. Every tuple of the table stores information such as the creator of the product.
8. **Software** - A sub-category of the products table. This table stores extra information regarding all products that can be identified as e-books, games, application software and utility software. Every tuple of the table stores information such as the owner corporation.
9. **Tools & Automotive** - A sub-category of the products table. This table stores extra information regarding all products that can be identified as automotive tools and accessories, repair kits and security apparatus. Every tuple of the table stores information such as the name, the owner corporation, the image and the product description.

**Relationships**

1. **orders** - A customer buys a product. This is a many to many relationship since a product can be bought by multiple customers and a customer can buy multiple products
2. **sells** - A retailer sells a product. This is a one to many relationship. Each product is sold by only one retailer and is characterized by the product id. Every retailer will of course sell multiple products.
3. **wishes** - This is a relationship between a customer and a product. A customer can add a product to the wish list. But every product can be included in many wish lists of different customers. Therefore this is a many to many relationship.

**Application Description**

This application seeks to efficiently store, showcase and distribute products from different retailers to different customers across the world. Efficient handling of data will result in fast registration and retrieval of information to enhance the user experience and build relationships between our customers and retailers. As this relationship gets stronger it will create a feedback loop which will strengthen our business plan.

**Preliminary calculations**

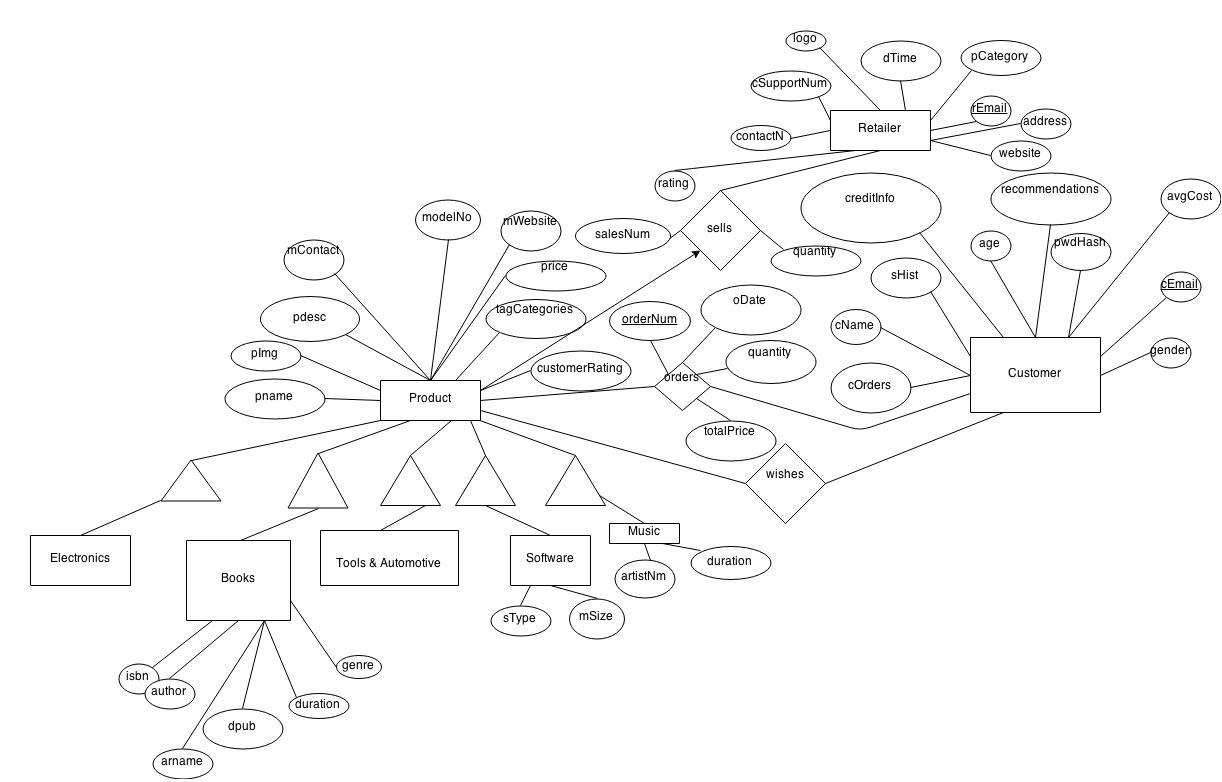
Due to budgetary limitations we will offer a limited set of recommendations to every customer. Suppose the number of recommendations provided to the customer is N. The algorithm to select a list of recommendations for a particular customer is as follows:

1. Divide N in the ratio of a:b:c…:z where a,b,c… are the number of products bought by a customer from every category offered.
2. Take the floor function of the every numeral a,b,c. After flooring if the total number of recommendations is less than N then offer the remaining number from the category which is most popular among all customers. This information can be gathered from the product list table quite easily.
3. Now take the most popular products from each category and recommend the an appropriate number of products from each to the customer.

**Functional examples**

* **getRatio(Customer customer) -** This function gets the ratio a:b:c…:z based on the number of products bought by the customer from every category. A category will not be included in the ratio if there are no products that have been bought by the customer from that category. If the customer has not bought any products then the ratio is calculated from the list of all products sold.
* **getProductsFromRatio(Ratio ratio, Customer customer) -** With the given ratio retrieve and display the products by selecting products from each category in the order of their popularity.

**E/R Model**



**Entities**

1. Product(pID, pname, pImg, pdesc, mContact, serialNo, mWebsite, price, tagCategories, customerRating)
2. Electronics(pID, creator)
3. Books(pID, isbn, author, arname, dpub, duration, genre)
4. Tools&Automotive(pID)
5. Software(pID, sType, mSize)
6. Music(pID, artistNm, duration)
7. Retailer(rating, contactN, cSupportNum, logo, dTime, pCategory, email, address, website)
8. Customer(cOrders, cName, sHist, creditInfo, age, recommendations, pswrd hash, avg\_cost, cEmail, gender, recommendations)
9. Relationships
10. orders(cEmail, pID, orderNum)
11. wishes(cEmail, pID)
12. sells(pID, rEmail)

**Appendix**

* Most products under the “Software” and “Music” labels will be able to be downloaded directly through the website after payment.
* “Ruby on Rails” is an open source web application framework we will be using to develop our application
* Credit information - This is the payment information of customers. It will be encoded using SHA512
* Books - For every book that has an ebook version, it will be compatible and available to download on Amazon Kindle, iPad(s) and Nook. This has been worked out after a deal
* Please refer to [www.amazon.com](http://www.amazon.com) and [www.flipkart.com](http://www.flipkart.com) for examples of the kind of service we would like to provide