Project: NittanyMarket Design

Author Name: Yihua Yang (yqy5329@psu.edu)

Course: CMPSC 431W

Date: 3/2/2022

Contents

Introduction

This project report provides **Requirement Analysis**, **Conceptual Database Design**, **Technology Survey**, **Logical Database Design**, and **Schema Normalization** about *NittanyMarket*, which is an online platform designed for people in Nittany State University (NSU) and the surrounding area of NSU to buy or sell goods. More details about *NittanyMarket* will be discussed in the **Requirement Analysis** section.

In the **Conceptual Database Design** section, several entity-relationship (ER) diagrams will be provided based on the requirements discussed in the **Requirement Analysis** section to show general designs of the databases in *NittanyMarket* and the relationships among those databases.

In the **Technology Survey** section, some software or tools, such as Python and PyCharm IDE, that might be used in building the system of NittanyMarket will be introduced. A comparison of those technology and substitutes for some of them will be given to readers, which may be helpful for the system-building process.

In the **Logical Database Design** section, the ER diagrams from **Conceptual Database Design** will be transformed into schema for the data models of *NittanyMarket* and then the schema will be normalized in the **Schema Normalization** section so that the efficiency of the databased can be improved.

In the **Suggestions** section, a few suggestions which may help *NittanyMarket* work better will be given by the author. This section can be considered as the Extra-Credit part of the project.

All cited resources from the Internet or CMPSC 431W course materials will be listed in the **Citation List** section at the end of the project description.

Requirement Analysis

From a database-system view, the system functionalities of *NittanyMarket* can be divided into four parts: **Users**, **Products**, **Orders**, and **Category hierarchy**. As mentioned in the previous section, *NittanyMarket* is a platform for buying and selling among members of NSU and local people. Students, faculty, and local vendors are the target users of *NittanyMarket*. Rather than selling the latest big-brand products, the main kinds of the products sold on *NittanyMarket* are unused goods and some unique products provided by local businesses. When finding a specific product, users may need a category list, or a category hierarchy, so that they can narrow down the range of products and find the one they want more quickly. After buyers picked their wanted products, they should be able to place the order on *NittanyMarket*. These points are the basic requirement of the system functionalities. The following parts are the detailed description and analysis of the requirements.

● <u>Users</u>

There are three types of users in *NittanyMarket*: **Buyers**, **Sellers**, and **Administrator**. **Buyers** are members of Nittany State, including students, faculty, and staff. NSU members and local vendors can be the **Sellers** in *NittanyMarket* after they get the approval from the **Administrator**. **Administrator** manages the *NittanyMarket*. Many requests on *NittanyMarket* need to be approved by **Administrator**. The details of each type of users are described below:

■ Buyers:

As mentioned, **Buyers** are restricted to Nittany State members only for now, although local people may be included in the future. When creating a new buyer account, a buyer needs to fill in his/her personal information (age, gender, username, phone number, and NSU ID). In addition to the personal information, the buyer also needs to provide his/her home address and billing address, which contains street, city, state, and zip code. The buyer will need to agree to the User Agreement of *NittanyMarket* to create an account, since the buyer's information may be collected and analyzed for marketing analysis. A default password will be given to the new account. It is encouraged for users to change the default password when they log in next time. To sign in the *NittanyMarket*, buyers need to provide their email address as user ID and the corresponding password. Phone number and username cannot be used as user ID for now, and this might be changed in the future.

To be specific, **Buyers** can do the following things:

- Modifying account information (*except* user ID)
- Purchasing products and placing orders
- Rating from 0* (worst) to 5* (best)

- Leaving comments to a bought product and the seller of the bought product
- Browsing products based on categories
- Managing payment information (adding/deleting credit cards or debits cards information)
- Contacting a seller before/after buying a product
- Contacting the Administrator if a buyer encounters a problem
- Submitting a request form if a buyer wants to have the access to sell a product

■ **Sellers:**

For now, only members of NSU and local businesses (approved by the **Administrator**) can hold seller accounts on *NittanyMarket*. Other vendors outside the NSU area may be eligible to apply **Seller** accounts in the future. Like **Buyers**, when **Sellers** create accounts, they will also need to provide their personal information, including age, gender, username, phone number, and NSU ID (if they have). An initial password will be also given when a seller account first creates. It also needs the email address (user ID) and the password to log in a seller account. Those above features are the points **Buyers** and **Sellers** have in common. However, it is not required for the **Sellers** to provide the home address and the billing address.

Note: For non-NSU member Sellers, they need to provide some additional information, which are listed below:

- Business name
- Business address
- Customer service phone number
- Quality Certification Proof of the products they want to sell

There are some extra features of **Sellers** compared with **Buyers**:

- Automatically publishing their products for listing
- Viewing both the product specification and the statistics of the product
- Responding to the comments of their products given by **Buyers**
- Listing products to an existing category
- Apply for a new category (**Administrator**-approval needed)
- Editing prices/details/titles (except for the category) of the products they sold
- Viewing the name, phone number, username, and the address of the **Buyers** who buy their goods
- Receiving payment from **Buyers** after an order completed
- Removing a product sold by the **Seller** from the listing

■ Administrator:

Administrator accounts can process the applications from **Sellers** and **Buyers**, handle the issues of orders, manage the product information, and maintain the running of *NittanyMarket* system. Thanks to the First Amendment of the Constitution[1], no one can delete negative comments from buyers, even the **Administrator**. Specifically, the **Administrator** account has the following permissions:

- Approving a **Buyer** account to have the access to sell products (i.e., converting an account type from **Buyer** to **Seller**)
- Approving a new category suggested by a **Seller**
- Managing the product categories in *NittanyMarket*
- Canceling an order if requested by **Seller**/**Buyer**
- Allowing an eligible local vendor to become a **Seller** on *NittanyMarket*
- Maintaining the running of *NittanyMarket* system
- Resetting the password of an account if needed

● <u>Products</u>

In this part, all information relating to a specific product will be included here. For the products sold on NittanyMarket, they can be divided into two kinds: common products and unique products.

What is the difference between common products and unique products? Unique products are sold by local vendors and are only available to buy in the NSU region, such as a pair of lovely shoes with the Nittany Lion logo made by our local business M&A, while the common products are mostly sold by NSU members, and you can probably buy a product as same as the common product outside the NSU region.

Regardless of the type of the products, both types share some common features. To be specific, the information listed below are necessary (except the tag) for a **Seller** to list a product on *NittanyMarket*:

- Title of the product
- Category of the product
- Product Price
- Product's detail or specification.
- (Optional) Tag that describe the product's characteristics, which helps **Buyers** to find the products they want

1 https://www.constituteproject.org/constitution/United_States_of_America_1992

As noted in the given requirements, every product must be classified to exactly one category, and every product item can be only listed and sold by one seller. From a DBMS view, this means a key constraint between the **Products** and **Sellers** and a key constraint between the **Products** and **Category hierarchy**. Therefore, it is needed to reflect the key constraints in the **Conceptual Database Design** section. Besides, each product should have its unique ID. As a product will be removed from a list after the Seller of the product leaves *NittanyMarket*, it suggests that there should be a weak entity in the database.

● Orders

After the **Buyer** clicks the "Buy" button and finishes to pay for the product, an order will be placed to the **Seller**. The order view from a **Seller** and a **Buyer** is slightly different, but both views share some common features, which are listed below:

- Product name
- Amount of the product the **Buyer** buys
- Information of the **Buyer** (name, address, phone number)
- Information of the **Seller** (name, link to the Seller's page, contact information)
- Status of the order (ordered, shipped, out of delivery, and completed)
- Link to sending order cancellation request to the **Administrator**

Note: the funds will be returned to the **Buyer**'s account after the **Administrator** approves the cancellation. Both Seller and Buyer can view the information of the order if the order has been cancelled.

The difference between the **Seller**'s view and the **Buyer**'s view is that the **Buyer** can still see the contact information of the **Seller** in case of after-sale service after the order completed, Due to privacy concerns, the **Seller** cannot view the Buyer's information, such as address and phone number, after the order completed. More details about the privacy will be listed in the User Agreement of *NittanyMarket*.

● Category hierarchy

A category is a list of a specific kind of products; all products in the same category share some common characteristics. Although categories already provide a primary classification to **Users**, to narrow down the range of products for **Users**, there exist subcategories within each category.

All initial categories are created by the **Administrator**, while a new category can be added after the **Seller** a request to the Administrator and get approval from **Administrator**. Similar to the categories, creating a new subcategory will also need the approval from the **Administrator**. However, there are no default subcategories in a category; all subcategories need to be created by **Sellers** and approved by the **Administrator**.

Conceptual Database Design

In this section, the ER diagrams based on the descriptions and the requirements in the **Requirement Analysis** section will be provided in the order of the functionalities in the previous section, and the full ER diagram of the NittanyMarket will included at the end of this section. Notice that some features may not be expressed in ER diagrams.

● Users

As mentioned in the last section, **Users** contains **Buyers**, **Sellers**, and **Administrator**. However, it is not required to draw the ER Diagram for Administrator. While there exist differences between Buyers and Sellers, both of them share some common features. Therefore, the ER diagram for Users can be drawn like below (See **Figure 1**):

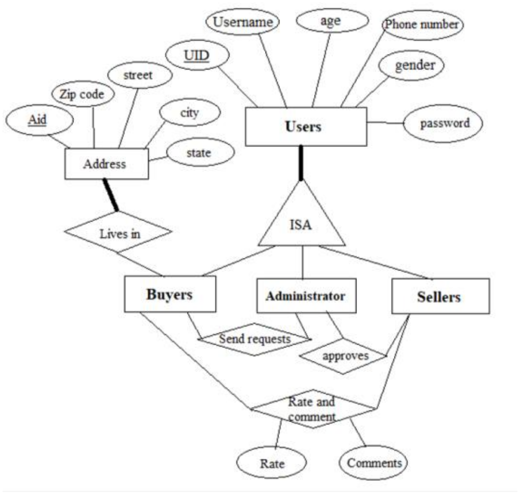


Figure 1

As the **Figure 1** shows, there is a Class (ISA) Hierarchy in the diagram. That means, **Buyers** and **Sellers** belong to **User** class. All attributes of **Users** will be inherited by **Buyers** and **Sellers**. Thus, both the **Sellers** and the **Buyers** needs to have UID (user ID, which is the email of the **User**), username, age, phone number, and gender for their account. Since either **Sellers** or **Buyers** must be the **Users** of *NittanyMarket*, it needs the bolded line between ISA and **Users** to indicate the total participation. One of the important relationships between **Buyers** and **Sellers** is that the **Buyers** can rate and leave comments to the **Seller**.

Compared to the **Sellers**, **Buyers** are required to provide their home address and billing address, and the “Lives in” relationship shows that requirement. The Address entity contains Aid (address ID), zip code, street, city, and state, and the Aid is the primary key for the entity. Both home address and billing address can be counted in the Address entity.

In addition, there is a relationship “send request” between the **Buyers** and the **Administrator** and a relationship “approves” between the **Administrator** and the **Sellers**. As mentioned in the last section, a **Buyer** will need to send a request to the **Administrator** to become a **Seller**, and the **Administrator** can approve the request of creating new categories/subcategories from the Seller. These relationships indicate such requirements.

Products

As described in the **Requirement Analysis** section, Products should include some necessary information that helps the **Buyers** to choose. Price, title, product detail, and category are such the necessary information for a product. Thus, the corresponding ER diagram for Products can be designed as the following (See **Figure 2**):

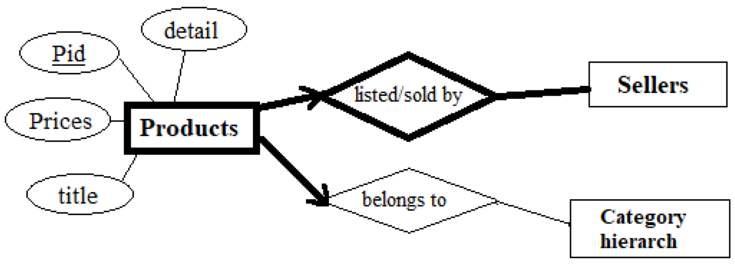


Figure 2

As shown in **Figure 2**, Pid (Product ID) is the primary key, which is unique for each product. Therefore, there is no need to worry about the case that two **Sellers** sell same books (same book names and same ISBN), because the books will have different Pid. The **Products** entity contains price, title, and detail as attributes for a product. Note that the tag mentioned in the previous section is optional, so the tag is not listed as an attribute in the diagram. The attributes and relationships of **Sellers** and **Category** hierarchy are neglected in this diagram and will be discussed in the corresponding diagrams.

There are two relationships here, one is “listed/sold by” between **Products** and **Sellers**, and another one is “belongs to” between **Products** and **Category hierarchy**. As mentioned in page 5 and page 6, every product must be classified to exactly one category, and every product is listed by exactly one **Seller**. To meet these requirements, there are total participation (the bolded line) and key constraints (the arrows) in the diagram. Because a product will be unavailable when the Seller who sells the product leaves *NittanyMarket*, **Products** is a weak entity set, and its owner entity set is **Sellers**.

Orders

As discussed in the **Orders** part of the **Requirement Analysis** section, information such as a delivery address, the total price of the order, number of Products in the order are needed for an order. As common sense, there must be a buyer who buys a product, or some products, a seller to make an order. Based on the understanding of the concepts of **Orders**, an ER diagram for **Orders** can be drawn like the below figure (See **Figure 3**):

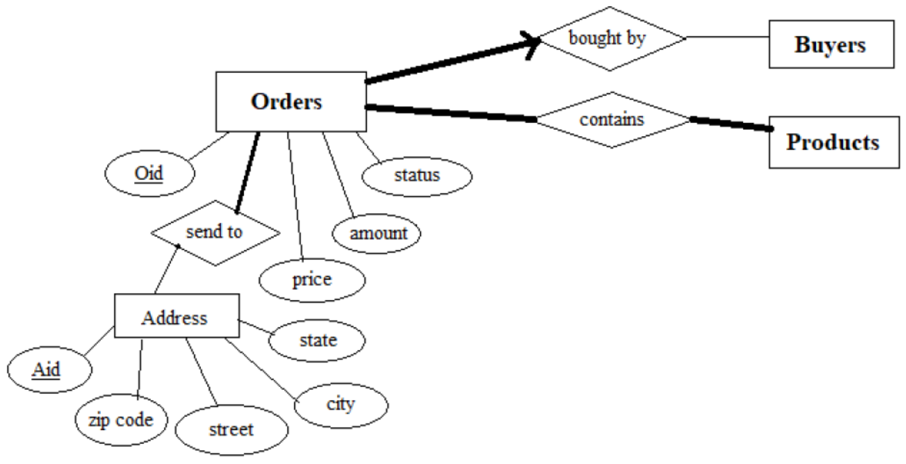


Figure 3

According to **Figure 3**, Oid (order ID) is the primary key of the **Orders** entity, and the entity has price, amount, status as its attributes. Address is an entity, which contains zip code, street, city, state as its attributes, and there is a relationship called "send to" between **Orders** and **Address**. The key constraint (the arrow) and the total participation (the bolded line) in the "bought by" relationship indicate that every order must be bought by a **Seller**. Besides, a valid **Order** must contain at least one **Product**. Thus, in the "contains" relationship, there is total participation between contains and **Orders** and total participation between contains and **Products**. Note that the attributes and relationships of **Buyers** and **Products** are neglected in this diagram.

● Category hierarchy

In the ER diagram of **Category hierarchy**, each category can include subcategories and zero or more Products. As discussed in the **Requirement Analysis**, new categories and subcategories can be added by the Administrator if a Seller sends the request. However, this feature will not be represented in the ER diagram. A possible ER diagram for **Category hierarchy** may look like the below figure (See Figure 4):

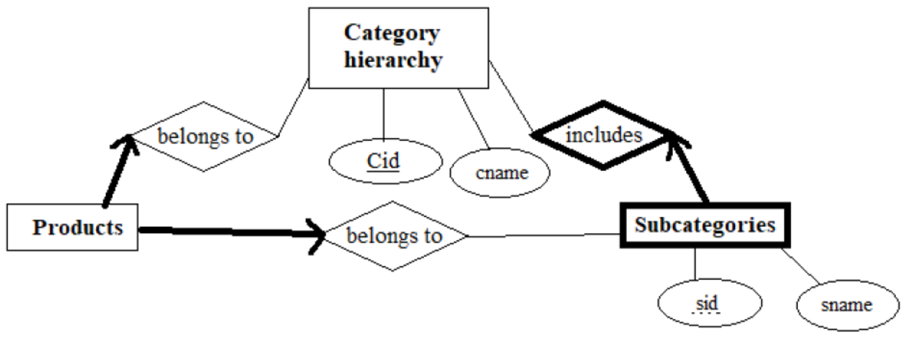


Figure 4

As Figure 4 shows, Cid is the primary key for the **Category hierarchy**. There is a "belongs to" relationship between **Products** and **Category hierarchy**, which has been discussed in the **Products** part in the section. A Category hierarchy may contain a few subcategories, as the relationship "includes" shows. When a category which contains at least one subcategory be removed by **Administrator**, all subcategory under this category will be also deleted. Therefore, Subcategories should be a weak entity set, and **Category hierarchy** is its owner entity set. Notice that the sid (subcategory ID) is the partial key of the weak entity and the cname (Category name for **Category hierarchy**) and the sname (Subcategories name for Subcategories).

● Full ER Diagram

As all parts of the ER diagram has been discussed above, if these separate diagrams are put together, the full ER diagram will look like the following figure (See **Figure 5**):

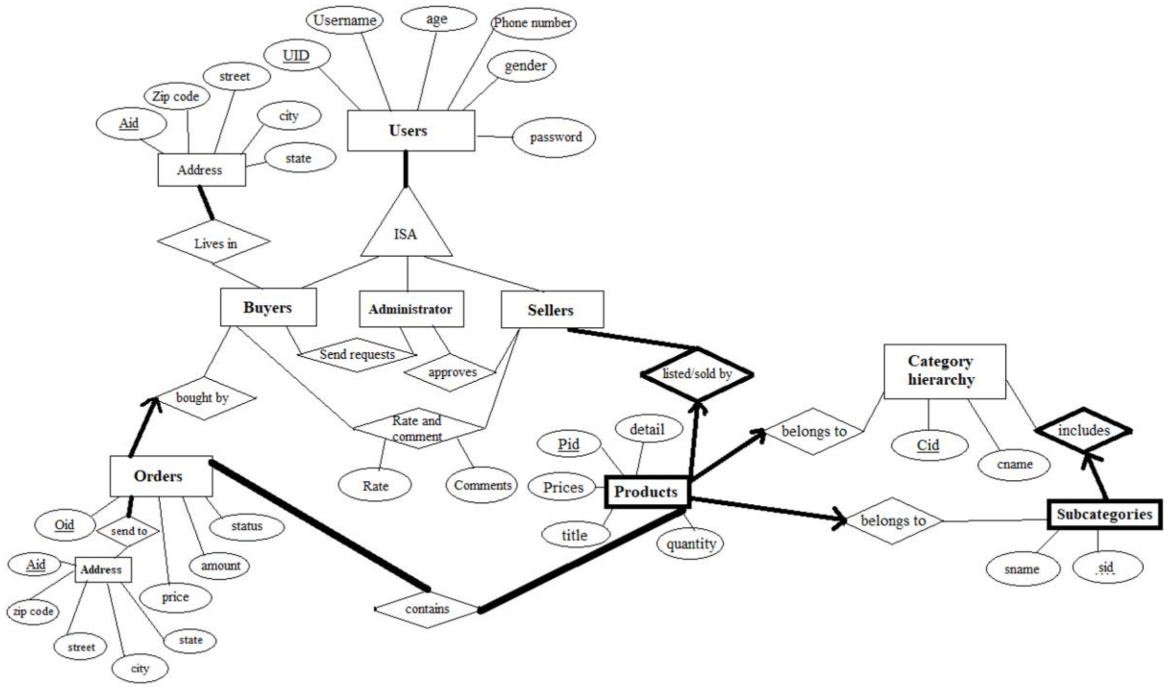


Figure 5

Technology Survey

To build such an elaborate online platform for the Nittany State community and locality, it is important to do a technology survey to find out appropriate tools before actually starting the programming work. Although several web programming frameworks, programming languages, and database management systems are suggested in the CMPSC 431W course, which may be helpful during the programming process of the *NittanyMarket*, there are many substitutes to them.

In this section, comparisons between the tools (web programming framework, programming language, query language) offered in the CMPSC 431W and their alternatives will be provided to you so that you can make a better choice for building the *NittanyMarket*. After comparing the technology tools, the impact and relevance of these technologies to the computer science field and society/industry will be discussed.

● Web Programming Frameworks

According to the definition of web programming frameworks from Wikipedia, it is a kind of software that "designed to support the development of web applications including web services, web resources, and web APIs"[2] and "provide a standard way to build and deploy web applications on the World Wide Web"[3]. In the CMPSC 431W class, a web programming framework called Flask is suggested by the teaching team of the course. However, there is another one called Django which is popular and always be considered as a substitute to Flask. Some characteristics of them from hackr.io[4] are listed below:

◆ **Flask:**
- Simple and easy to implement
- Light-weighted with minimalistic features
- Developers can keep control of the core of the application
- Does not have any features to handle administration tasks
- Requires much fewer lines of code for a simple task

◆ **Django:**
- Full-stack web framework that follows the batteries-included approach
- Contains more functionalities than Flask
- May need more lines of code to build a same function
- Complicate, and needs more time to learn

Based on the above characteristics, it will be better to choose Flask as the web programming framework for *NittanyMarket*, due to the following reasons. Firstly, Flask is more simple and easier

[2] https://en.wikipedia.org/wiki/Web_framework
[3] https://en.wikipedia.org/wiki/Web_framework
[4] https://hackr.io/blog/flask-vs-django#:~:text=Flask%20is%20a%20micro%2Dframework,are%20used%20in%20different%20applications.

to implement, compared with Django. Due to the limit of time, it may be hard to learn about Django. Secondly, although Django offers more functionalities than Flask, considering the current requirements of *NittanyMarket*, which are not very complicate, lightweight Flask is good enough to fulfill the requirements of *NittanyMarket*. Thus, it is better to use Flask here, which is suggested by CMPSC 431W teaching staff.

● Programming Languages

The definition of programming languages is that "one kind of computer language and are used in computer programming to implement algorithms"[5]. For building such an online platform like *NittanyMarket*, it is unavoidable to use a programming language. Nowadays, there are plenty of programming languages in the world. Python, which is recommended by the teaching team, and Java, which is another popular programming language, can be used for the programming process in this project. Both of them are object-orientated programming language and have a lot of libraries. Several differences between Python and Java attained from the Internet[6] are listed below for comparison.

◆ **Python:**
- Compiles code at runtime
- Dynamically typed variables
- Easier syntax
- Whitespace matters, semicolons do not matter

◆ **Java:**
- Compiles code in advance
- Statically typed variables
- More complicate syntax
- Semicolons matter, whitespace does not matter

Based on the features given above, it may be a better choice to use Python rather than Java. Firstly, Python is easier to learn compared to Java due to its simple syntax. Secondly, because whitespace matters in Python, the readability of a Python code will be higher than a Java code. Thirdly, although the performance of a Java code may be higher than a Python Code, because Python compiles code at runtime instead of in advance, the performance of Python is good enough for this project based on the given requirements. Therefore, it will be better to choose Python as the programming language for building *NittanyMarket*.

● Query Languages

Obviously, it needs a tool to process data in the databases of *NittanyMarket*. The query language is such a tool to manage the database system, and SQLite and MySQL are query languages.

[5] https://en.wikipedia.org/wiki/Programming_language
[6] https://raygun.com/blog/java-vs-python/

SQLite is the one suggested by CMPSC 431W teaching staff, while MySQL is its strong competitor. Based on the information from hostinger.com[7], although both of them are open-source project for Rational Database Management System, there exist differences between them, which are listed below:

◆ **SQLite:**
- Only supports a few datatypes (Blob, Integer, Null, Text, Real)
- Smaller size, about 250 KB
- Does not have any specific user management functionality
- Does not have an inbuilt authentication mechanism
- Easy to setup and use

◆ **MySQL:**
- Supports more datatypes than SQLite
- Larger size, about 600 MB
- Well-constructed user management system
- Have a lot of inbuilt security features
- Requires more configurations to setup

Due to the points listed above and the requirements of *NittanyMarket*, using SQLite may be a better choice compared to using MySQL for the following reasons. Firstly, the smaller size of SQLite compared to the size of MySQL and the ease of setup and use is the most attractive point for SQLite. This is good for a beginner to start a basic program. Secondly, although SQLite contains fewer functionalities than MySQL, it can still finish the goals of NittanyMarket so that there is no need to worry about the lack of some features in SQLite. Hence, it is better to choose SQLite for this project.

Impact and Relevance

Modern people's life changed a lot with the development of technology. Database management system, which includes query languages and programming languages etc., is such a technology that impacts people's daily life. That is why the impact of database technology and the relevance to computer science fields and society will be discussed in this part.

From a computer scientist's view, the emergence of database technology provides many opportunities for people to work and study. Today, database-related topics are one of the most popular subject areas among universities. Lots of scholars are working hard to optimize current database systems and try to apply the database system to people's daily life for improving efficiency. Besides, many jobs about databases are created due to the emergence of this technology, which provides opportunities to those computer science students who are still confused about what to do. These points mentioned above are the impact and relevance to the computer science field.

[7] https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/#:~:text=SQLite%20is%20a%20server%2Dless,to%20interact%20over%20a%20network.

Database technology also impact the society. Rather than storing files on bookshelves, many companies and institutions are using database to manage their files for a better efficiency. Using database technology can also provide a timely access to data so that the productivity will be increased. Those are the impact to the society.

Logical Database Design and Normalization

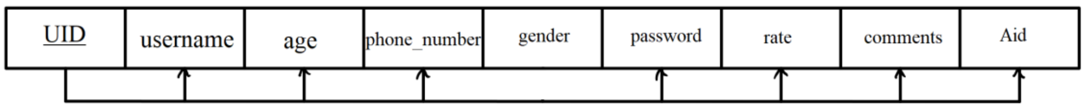
In this section, the ER model diagrams given in the **Conceptual Database Design** section will be finalize to relational schema and be refined to reduces data redundancy. All the details regarding the logical design of *NittanyMarket* and the normalization of those schema will be provided below.

● Users

Based on the ER diagram of **Users** in the **Conceptual Database Design** section, the schema of Buyers, Sellers, Address can be described as below. Notice that UID is the email address of a **User**, and the schema of **Administrator** is neglected.

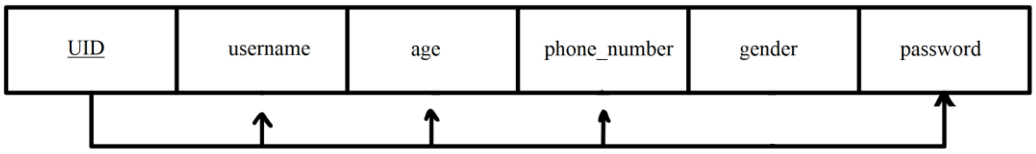
Buyers (UID, username, age, phone_number, gender, password, rate, comments, Aid);
Sellers (UID, username, age, phone_number, gender, password);
Address (Aid, zip_code, street, city, state, UID).

■ According to the **Buyer's** schema, the relation can be drawn like:



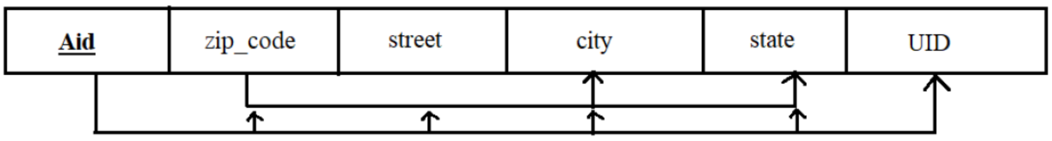
This relation is already in the 3rd normal form. Note that gender is an attribute that cannot be normalized.

■ According to the **Seller**'s schema, the relation can be drawn like:

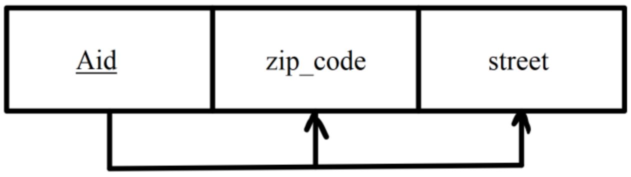


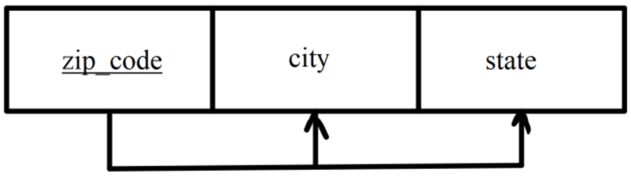
This relation is already in the 3rd Normal form. Note that gender is an attribute that cannot be normalized.

■ According to the **Address**'s schema, the relation can be drawn like:



This relation is not in the 3rd Normal from. To make the relation to be a 3rd normal form, a possible solution is separating the origin table into two tables, one contains Aid (Key), zip_code, and UID, and another contains zip_code (Key), city, and state. Therefore, the 3rd normal form of Address's schema will look like this:



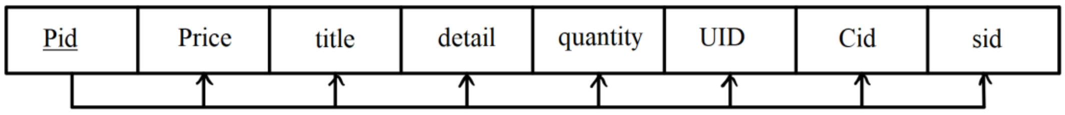


● Products

Based on the ER diagram of **Products** in the **Conceptual Database Design** section, the schema of **Products** can be described as below.

Products (<u>Pid</u>, Price, title, detail, quantity, UID, Cid, sid).

■ Based on the **Product**'s schema, the relation of it can be drawn like below:



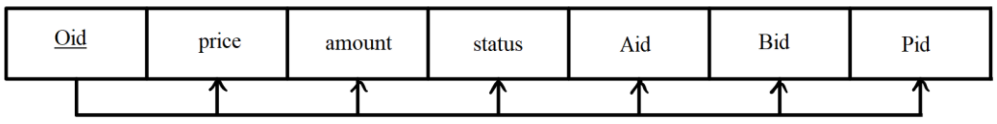
This relation is already in the 3rd normal form. Notice that the UID here points to a **Seller**'s UID.

● Orders

Based on the ER diagram of **Orders** in the **Conceptual Database Design** section, the schema of **Orders** can be described as below.

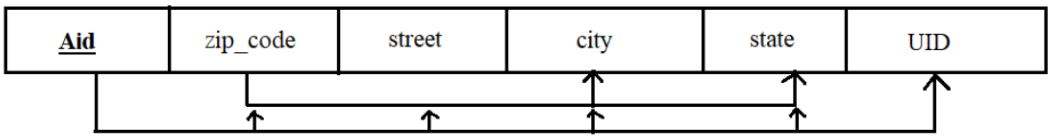
Orders (<u>Oid</u>, price, amount, status, Aid, Bid, Pid);
Address (<u>Aid</u>, zip_code, street, city, state, UID).

■ Based on the **Order**'s schema, the relation of it can be drawn like below:

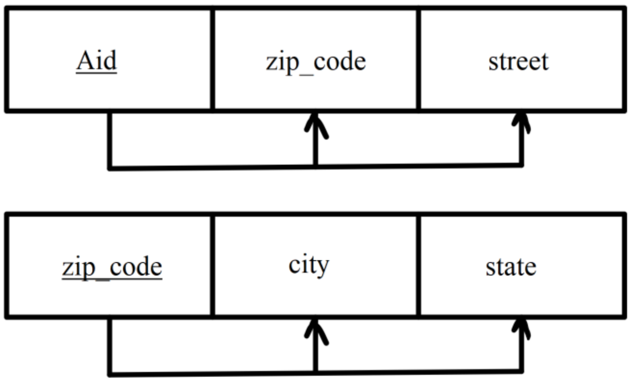


This relation is already in the 3rd Normal form.

- According to the **Address**'s schema, the relation can be drawn like:



This relation is not in the 3rd Normal from. To make the relation to be a 3rd normal form, a possible solution is separating the origin table into two tables, one contains Aid (Key), zip_code, and UID, and another contains zip_code (Key), city, and state. Therefore, the 3rd normal form of Address's schema will look like the below figure. Note that the Address here is as same as the one for **Buyers**.

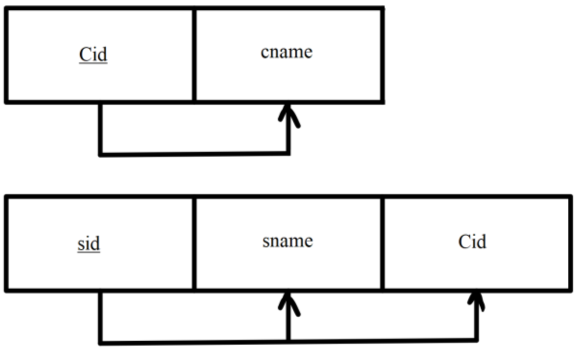


● Category hierarchy

Based on the ER diagram of **Category hierarchy** in the **Conceptual Database Design** section, the schema of **Category hierarchy** can be described as below.

Categories (Cid, cname);
Subcategories (sid, sname, Cid).

- Based on the **Category hierarchy**'s schema and Subcategories's schema, the relation of it can be drawn like below:



These two relations are already in the 3rd Normal form.

Suggestions

Although there is a complete structure of the *NittanyMarket* elaborated in the previous sections, some extra features and functions can be added to the platform to make *NittanyMarket* more useful. Recommendation page, history price record of a product, product reservation are such features/functions that might help *NittanyMarket* do better, and those features/functions can be considered as extra credits for this project report.

- **Recommendation Page**

When a consumer has money and is confused about what to buy, why not provide him/her with a recommendation page to inspire his/her idea about shopping? After browsing the products on the recommendation page, the consumer may buy something. That will be great for *NittanyMarket*! Therefore, it is necessary to have a recommendation page for the Market.

To build such a recommendation page, an algorithm from machine learning called Matrix Factorization can be used here. According to Wikipedia, "matrix factorization algorithms work by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices" [8]. After using this algorithm, for a specific User, products that are highly rated by other Buyers whose preferences are similar to the User will be displayed on the User's recommendation page. However, implementing matrix factorization for the recommendation page of *NittanyMarket* might be time-consuming.

- **History Price Record**

Buyers are always comparing prices of a product among plentiful shopping websites and looking forward to buying the product at the lowest price. It will be a good idea to show the lowest price and price trend on *NittanyMarket* to attract consumers. Rather than using unreliable history price recorder made by others, why not to make a built-in history price recorder for *NittanyMarket*? The recorder records daily prices of each product sold on the market. Undoubtedly, Buyers cannot wait to buy the goods on *NittanyMarket* after seeing the historical price. Besides, the history price record can offer data for marketing analysis, which is also benefit to *NittanyMarket*.

- **Product Reservation**

Thanks to the historical price recorder, the products on *NittanyMarket* are out of stock now, and there are still many Buyers willing to buy products on the market. It is time to offer a Product Reservation function for those crazy Buyers. When a Product is currently unavailable, Buyers can click the Reserve button on the product's page. Then the Buyers can be notified and pay for the products when their wanted products are available to buy so that *NittanyMarket* can get money as soon as the products become available. Hooray!

[8] https://en.wikipedia.org/wiki/Matrix_factorization_(recommender_systems)#:~:text=Matrix%20factorization%20is%20a%20class,two%20lower%20dimensionality%20rectangular%20matrices.

Citation List

All sources cited in this report will be listed below (not in order):

1. https://www.constituteproject.org/constitution/United_States_of_America_1992
2. https://en.wikipedia.org/wiki/Web_framework
3. https://en.wikipedia.org/wiki/Web_framework
4. https://hackr.io/blog/flask-vs-django#:~:text=Flask%20is%20a%20micro%2Dframework,are%20used%20in%20different%20applications
5. https://en.wikipedia.org/wiki/Programming_language
6. https://raygun.com/blog/java-vs-python/
7. https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/#:~:text=SQLite%20is%20a%20server%2Dless,to%20interact%20over%20a%20network.
8. https://en.wikipedia.org/wiki/Matrix_factorization_(recommender_systems)#:~:text=Matrix%20factorization%20is%20a%20class,two%20lower%20dimensionality%20rectangular%20matrices.