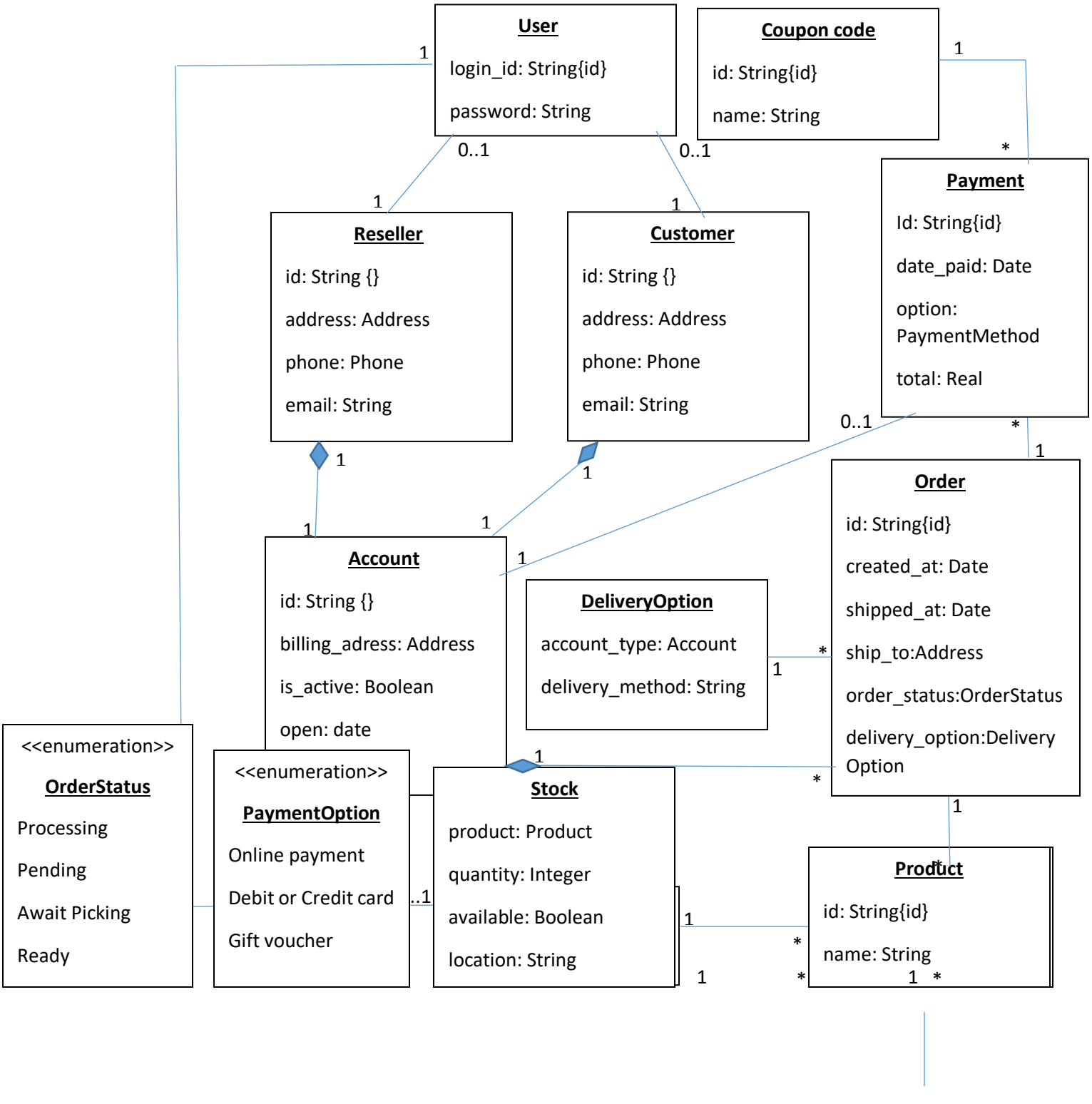
A Class diagram

In Unified Modelling Language(UML), a class is a static structure diagram that describes a system's classes, attributes, and relations between the objects (Planas and Cabot, 2020). While designing a class diagram for a certain system, they can take one of the following perspectives:

1) Conceptual perspective- where the diagrams are interpreted to describe objects in the real world. These perspective diagrams denote the concepts of the domain under a study.
2) Specification perspective- These diagrams are interpreted to describe the abstraction or rather the components of the system under study and their interfaces without commitment to a particular implementation.
3) Implementation perspective- The diagrams describe the implementation of software in certain technology and language

From the information above on various perspectives, it's clear the specification perspective will be used to describe the objects of our system under study.

Online store system



Class diagram description

For any user to successfully transact with the system, they must have an account. An account can be of two types, a regular customer account or a reseller account. For an account to exist, there must be a customer or a reseller to create the account as indicated in the diagram. A customer or a reseller can only have a single account and there can only be one account for each customer or a reseller.

Again, a single account can make several orders and make several payments, and the inverse of that orders can only be placed by a single account, and also only a single account can initiate payment of an order.

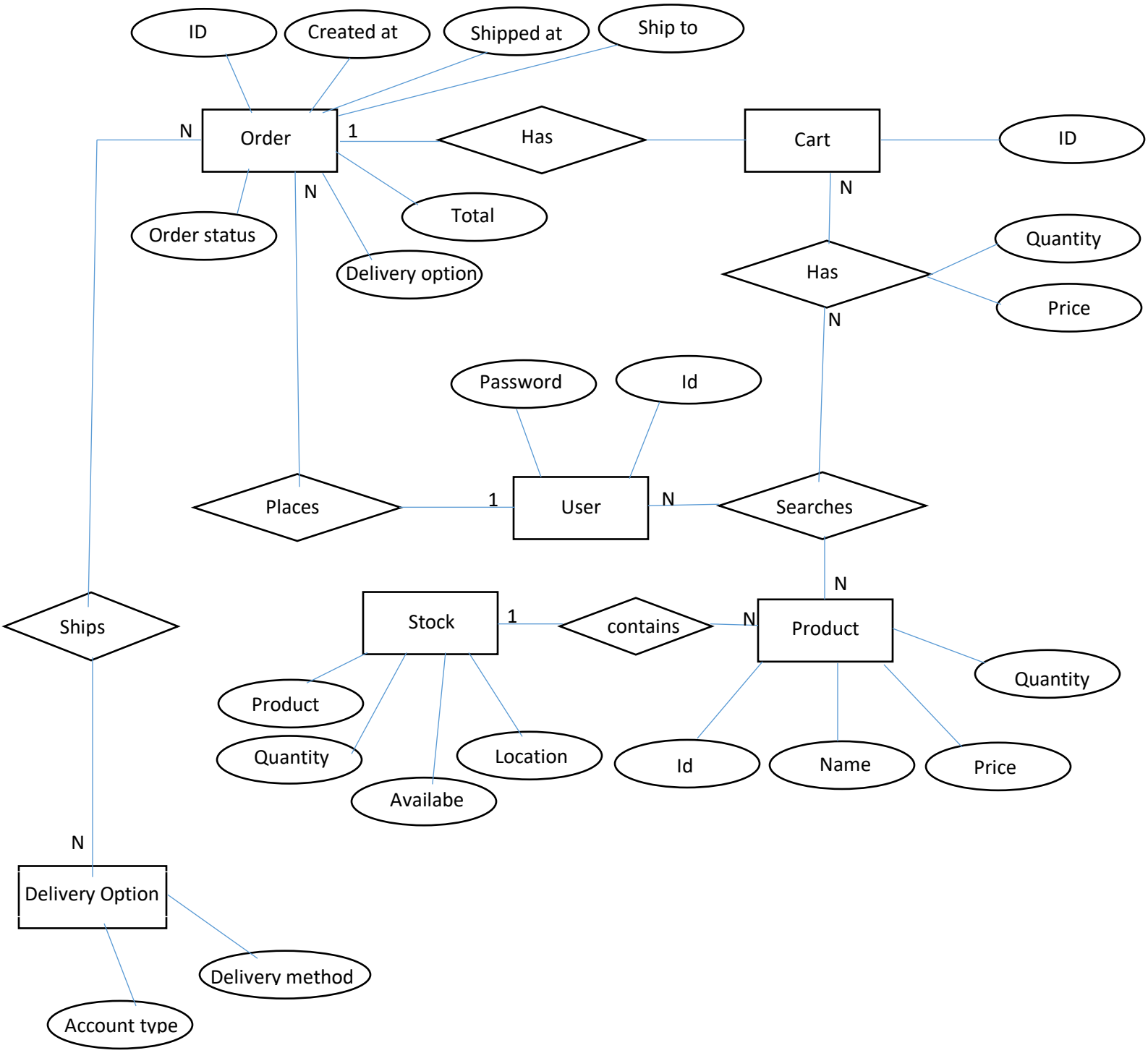
A logged-in user can view the shopping basket and can have one or many shopping basket records. A shopping basket contains products to be ordered so the relationship is one-to-many products. A stock class contains the products details including the quantity of a product and its location in the warehouse.

Logged in users can order none or many products from the system, an order class contains the id of the order, date created, date the order was shipped, the shipping address, order status, the delivery options which returns either the in house delivery or the postal delivery option.

Online store system

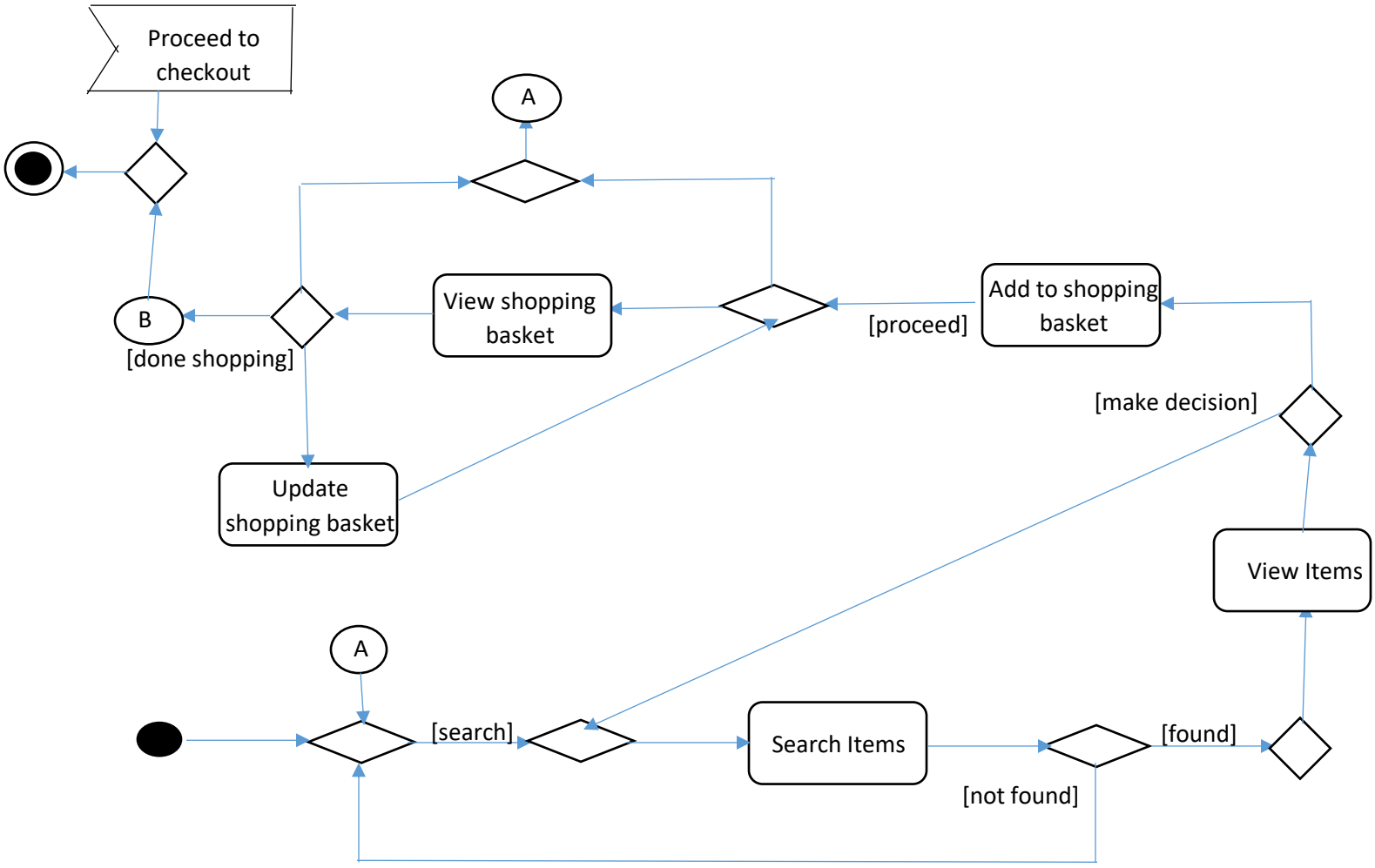
An entity-relationship diagram

E-R diagram shows the relationships of object sets stored in a database.



From the illustration above the entity-relationship diagram can simply be explained as the user searches for a product that is contained in stock, the user adds the product to the shopping cart if they like the product (Clarisó, González and Cabot, 2017). The cart has the price and quantity of the product required, while the user can place an order for the product. If they place an order the order is shipped by one of the shipping methods offered by the store.

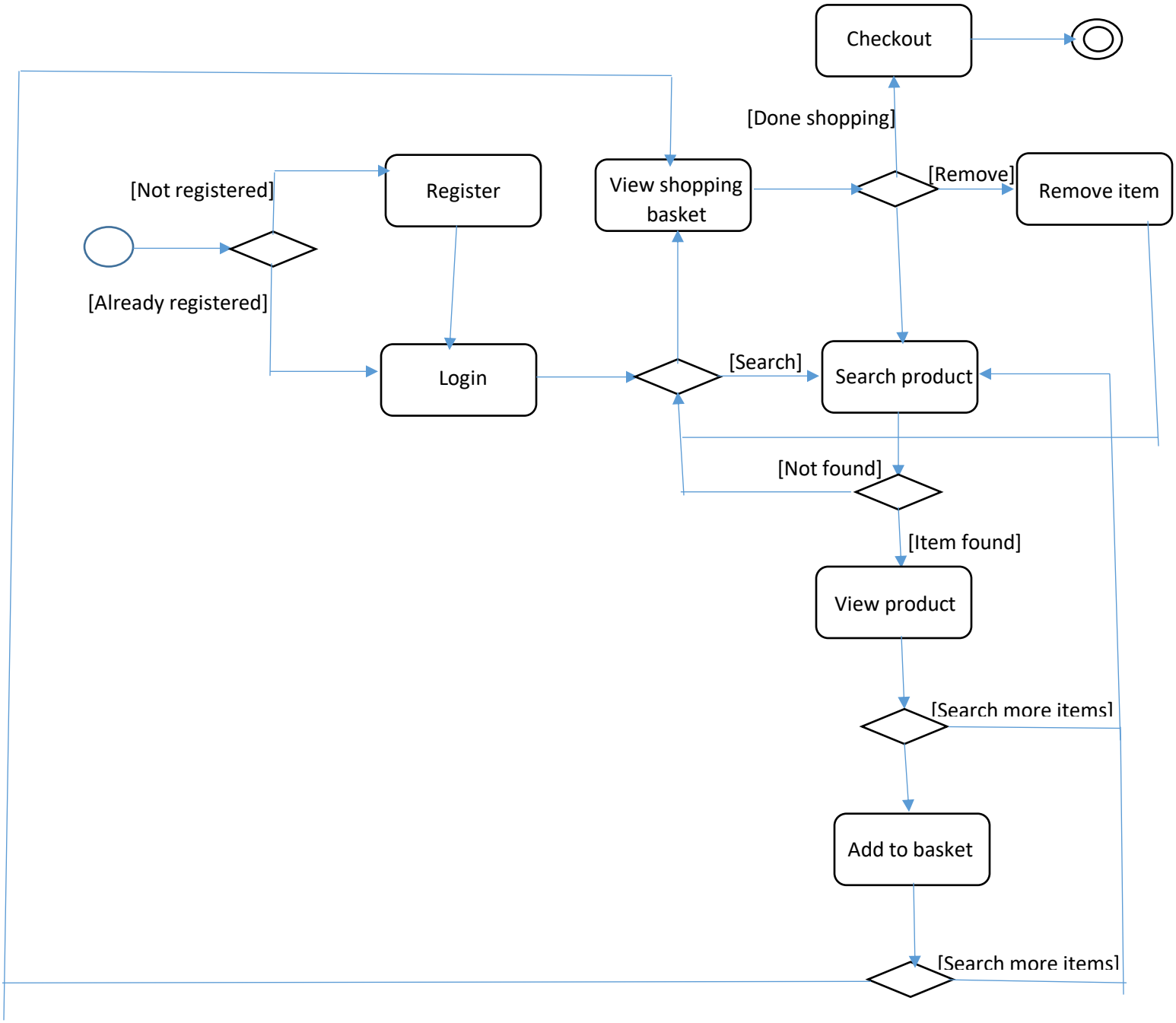
An activity diagram for the process of a customer completing an order



The diagram above describes the events the user will trigger and actions that they will take until an order is completely placed. A user's search is the entry point, they can search an item if found, they can view the item if not, they can go back to searching an item. If they view items, they could also come back and search for more items and add them to the cart then come back again to view their shopping basket. From the basket, they could decide to update their cart and come back to view their basket, or they could

purge all the items and go back to search for items. If they approve their basket, they can go ahead to checkout their basket.

A state diagram highlighting the states of order (and their transitions) for the system.



From the state diagram above, the entry point is when a user logs in or registers and logs in if first time. After they are logged into the system, they can search for a product. If not found, they can decide to search again or if the item is found, they can view the product and then add it to the basket. After adding the item(s) to the cart, they can go back to search for more products or they can view their shopping basket. From the shopping basket, they can remove an item and come back to search for an item or they can decide to check out if they are done.

Referances

Planas, E. and Cabot, J., 2020. How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus. *Computer Standards & Interfaces*, *67*, p.103363.

Clarisó, R., González, C.A. and Cabot, J., 2017. Smart bound selection for the verification of UML/OCL class diagrams. *IEEE Transactions on Software Engineering*, *45*(4), pp.412-426.

Sergievskiy, M., 2017. Description logic application for UML class diagrams optimization. *International Journal of Advanced Computer Science and Applications*, *8*(1), pp.268-272.

Ahmad, T., Iqbal, J., Ashraf, A., Truscan, D. and Porres, I., 2019. Model-based testing using UML activity diagrams: A systematic mapping study. *Computer Science Review*, *33*, pp.98-112.

Beckmann, M., Vogelsang, A. and Reuter, C., 2017, September. A case study on a specification approach using activity diagrams in requirements documents. In *2017 IEEE 25th International Requirements Engineering Conference (RE)* (pp. 253-262). IEEE.

Bian, W., Alam, O. and Kienzle, J., 2019, September. Automated grading of class diagrams. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)* (pp. 700-709). IEEE.