Rahúl Thapa, Alex Gustafson, Peter Lyu, Michael DiGiacomo

Professor Obermyer

CSC 4480-002

18 February 2019

Project 3 Design Rationale

We could have created a ternary relationship between Vendor, Computer\_User, and Technology\_Part labeled “Sells”, but this assumed certain things that weren’t necessarily true. It would be unclear if a computer user would be able to sell a part to a vendor (because a computer user has the ability to sell), which we don’t want as a possibility. We would need to introduce role names that would be complex, as a computer user can sell to another computer user, but not to a vendor, and a vendor can sell to a computer user. So a computer user has multiple possible roles in the ternary relationship, and this just seems unnecessarily complex when we could just use two simple binary relationships for “Sells”. Concerning the entity type of the computer user, we considered splitting this entity into several more descriptive, weak-entities to classify the type of user. However, we felt this would overcomplicate the design of the database. The practical differences between different users weren’t great enough to warrant separate entities, so instead we decided to create an attribute “Classifications” (with its own domain of possible users) in order to note the differences in users without creating the need for redundant relationships that would have worked the same way for each different user entity. Similarly, we also decided to give each computer users a surrogate ID apart from their own username and password. Also, we created a composite attribute Tech\_Id for technology parts because we felt that the serial number is not unique enough to uniquely identify the technology parts from different vendors in our database.

One of the decisions made was to keep the Device entity in the diagram. Originally, we didn’t think to have that entity would be very important because knowing what device(s) a user owns does not contribute to the database. When we considered how a device relates to certain technology parts - specifically, which parts are compatible with a certain device based on model, manufacturer, and capabilities - we agreed it was important to not only keep the Device entity but to also connect it to the Technology Part entity via the “Compatible\_with” relationship.

We also discussed about having a recommender system in our database so that our database can suggest users with the parts based on past searches and purchases. However, we did not know how to represent it in our ER Diagram. We were also confused about the uniqueness of technology parts as it is stored database and if the same part can be bought by multiple users.

We experienced a lot of ambiguity when analyzing the participation constraints on many of the relationships. We made many of the entities participate partially in their relationships (except for Device in the “Owns” relationship) because we thought they were more “may” than they were “must”. For instance, a Computer User *may* buy, sell, or search for parts, but that user does not necessarily *have* to do those things. Similarly, a vendor does not *have* to sell parts to the database, especially if the vendor is unaware of our database, and parts do not have to be sold from vendors since they can also be sold by users.