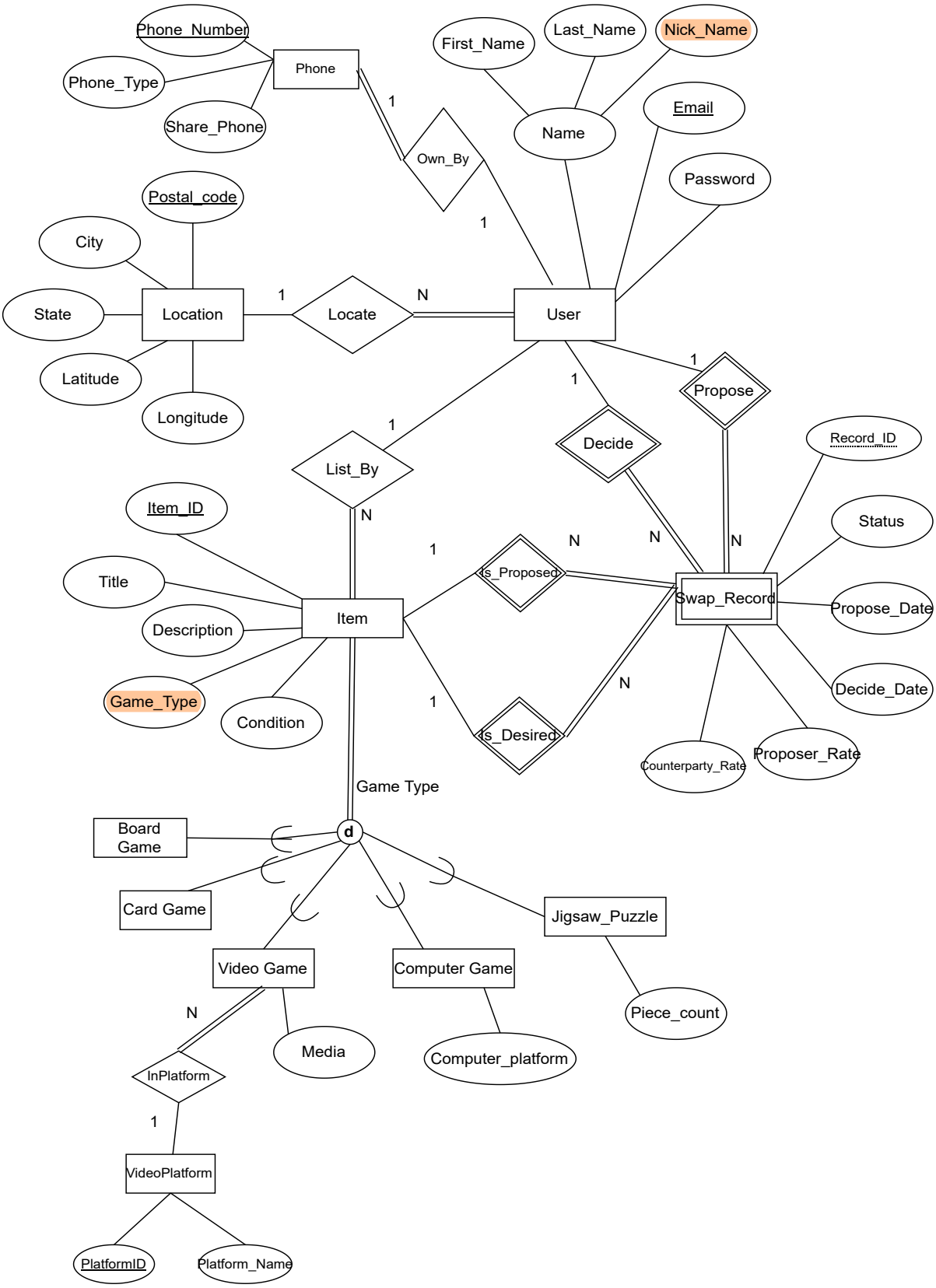
EER Diagram


Phone_Number
Phone_Type
Share_Phone
Phone
1
Own_By
First_Name
Last_Name
Nick_Name
Name
Email
Password
1
Postal_code
City
State
Location
1
Locate
N
User
Latitude
Longitude
1
List_By
1
Decide
1
Propose
Record_ID
Status
Item_ID
N
1
Is_Proposed
N
N
N
Swap_Record
Propose_Date
Title
Description
Item
Game_Type
Condition
1
Is_Desired
N
Decide_Date
Counterparty_Rate
Proposer_Rate
Game Type
d
Board Game
Card Game
Video Game
Computer Game
Jigsaw_Puzzle
Piece_count
N
Media
Computer_platform
InPlatform
1
VideoPlatform
PlatformID
Platform_Name

Supplement notes on EER diagram

Supplement notes on EER diagram

In most cases, there are many equivalent ways to represent the same entity or relationship and this supplement note provides some additional descriptions/explanations on our EER diagram.
General Naming rules

- We follow the textbook recommendations, use singular nouns for entities.
- If given, we try to directly use the (entity/attribute/relationship) name listed in the instruction file (e.g.: *Password*). Otherwise, we create the name ourselves and the general principle is to make the names meaningful and foster easy understanding. E.g.: ***Is_Proposed*** relationship between item and ***Swap_Record*** suggests that this item is associated with the proposer (i.e. it is provided by proposer), while ***Is_Desired*** relationship between item and ***Swap_Record*** suggests that this item is what the proposer desired in the ***Swap_Record***.
- For (entity/attribute/relationship) names involving only one word, only the first character is capitalized; for names involving more than one word, words are connected with _ and first character for each word is capitalized.

We partition this note into three sections according to three major entities in the EER diagram: ***User***, ***Item***, ***Swap_Record***

User

(1) [Phone] ***Phone*** is NOT presented as an attribute of ***User***, but a separate entity that is connected to ***User*** via relationship ***Own_Phone***. We need this to guarantee that a phone can only be linked to a single user.
(2) [Location] ***Location*** (*Postal_Code*, *City*, *State*) is NOT presented as an attribute of User, but a separate entity that is connected to User via relationship Locate. This is because we want to restrict the valid postal code, city, state to be within the list provided.

Item

(1) [Item_ID] *Item_ID* is a surrogate key generated by the system. We include it in the EER graph as this attribute is mentioned in the instruction document but it can be removed in the current stage (as suggested by HW submission section on Canvas)
(2) [Game_Type] Item (totally) participates in the specialization to five subtypes: ***Board_Game***, ***Card_Game***, ***Video_Game***, ***Computer_Game***, ***Jigsaw_Puzzle***. In later stages, one separate table/schema will be created for each subtype and we won't create a table for the supertype. We may optionally add an attribute Game_Type to the supertype Item to make it clearer but it is not required. (Alternatively we can also create a table for Item including the common attributes and separate tables for those game types with additional attributes. This way we the Game_Type variable is necessary as both Board_Game and Card_Game do not have additional attributes and no tables are needed for them.)
(3) [Video_Platform] ***Video_Platform*** is NOT presented as an attribute of ***Video_Game*** but a separate entity. The main reason is that this way we can make the list of platforms easier to update, as required in the instruction file. In later stages, *Platform_ID* (this is a surrogate key, it is optional in this stage) will be inserted to ***Video_Game*** schema as a foreign key, so even if we change the name of Platform from "Xbox" to "Ybox" hypothetically, we only need to update

one row in ***Video_Platform*** table rather than directly update all rows involving "Xbox" in ***Video_Platform***. Similar practices can also be adopted for other attributes with pre-defined limited set of values such as *Computer_Platform*, *Condition*, *Media* but it is not required, since we can also just define the set of choices in a checkbox in front end.

Swap_Record

(1) [Swap_Record] In our preferred EER diagram, the ***Swap_Record*** entity will be created at the time when a user propose a swap and it covers all possible status (pending/accepted/rejected/completed). The advantage is that this entity is a one stop shop and the structure is simple and the draw back is that some attributes will be left as NULL (e.g.: *Decide_Date*, *Proposer_Rate* etc) and the swap record will be modified later as users proceed with the swap request. Alternatively, we can only choose to create an intermediate entity called Proposal, whose records are generated at the time of proposal. When counterparty accept/rejects a proposal, we generate a new record for ***Swap_Record*** (we can choose to delete the record in Proposal or not) containing attributes such as *Decide_Date*. The advantage of this alternative is that we won't have many missing values but the disadvantage is that the additional layer complicates the problem.

(2) [Owner] ***Swap_Record*** is a weak entity owned by ***User*** and ***Item*** via four relationship: ***Propose***, ***Decide***, ***Is_Proposed***, ***Is_Desired***. A concrete example: User A proposes swap record with counterparty B, who needs to Decide on it later. In this record, item C (currently owned by A) ***Is_Proposed*** by A and item D (currently owned by B) ***Is_Desired*** by A. Note that the relationship with item is NOT sufficient to identify ***Swap_Record*** because it is not enough to identify the proposer and counterparty since the owner if an item changes after a swap or so. On the other hand, relationship with user is NOT sufficient to identify ***Swap_Record*** because the same user may have multiple items listed.

(3) [Record_ID] This is a surrogate key (partial identifier), it is optional for the current EER and can be ignored