Project: Lost and Found

- Our team is designing and developing *Lost and Found* system, which we anticipate to be added as an extension in BYU mobile application.

- Schema
    - Person(byuId, firstName, lastName, phoneNumber, email, isStudent)
        - Attributes
            - Primary Key: byuId(String)
            - firstName (String)
            - lastName (String)
            - phoneNumber (Number)
            - email(String)
            - isStudent (boolean)
        - Entity Description
            - This entity represents members of the service; members could be either students or staff members at BYU, basically any person at BYU. They could be either benefactors or beneficiaries of the service; for example, they could report lost items while they could also look for their lost stuff through the service.
        - Relationship with Other Relations
            - The primary key of this relation would be used as Foreign Key in other relations.
        - Normalization
            - The fact that all the attributes of this relation are dependent on byuId demonstrates the evidence of normalization.'

    - Item(itemId, type,color, picture, location, isRequested, name, description, date, founderId)
        - Attributes
            - Primary Key: itemId(String)
            - Foreign Key: founderId (byuId)
            - Name (String) //Name of the lost item; it would be used the title of a post
            - description(Siring)
            - date(Date)
            - type (String or Number) //This would be selected from pre-designated values
            - Color (String or Number) //This would be selected from pre-designated values
            - picture(Image)
            - location(Location or String)
            - isRequested(boolean) //When a user claims an object, this value would be true; otherwise, false.
        - Entity Description

- This entity represents lost Items. Data from this relation would be used to create posts on the service; for instance, when a user reports a lost item on the file, the post would be created automatically based on the data from `Item` relation.

- Relationship with Other Relations
  - The primary key, itemId, would be used as a foreign key in `Transaction` relation.
  - `founderId` is a primary key of `Person` relation, which represents a person who has reported/found a lost item.
- Normalization
  - There doesn't seem to be any transitive dependency; every single attribute seems to depend only on objectId, which is the primary key of the relation

- Transaction(transactionId, transactionDate, getterId, itemId)
  - Attributes
    - Primary Key: transactionId (String)
    - Foreign Key: getterId (byuId) //byuId of a person who claims/gets a lost item back
    - Foreign Key: itemId (byuId) //itemId of an item which has been lost
    - transactionDate (Date)
    - isReturned (boolean) //if a person with getterId actually receives an item back, this attribute would be `true`.
  - Empty Description
    - This entity represents transactions between those who have found/reported items and those who have lost/been finding items. If those falsely claim items receive items, they could be tracked down by byuId and would get penalties accordingly.
  - Relationship with Other Relations
    - This relation is using getterId(byuId), which represents a person who gets returned lost items
    - This relation is using itemId(itemId), which represents a lost item
  - Normalization
    - There doesn't seem to be any transitive dependency; every single attribute seems to depend only on transactionId, which is the primary key of the relation

Potential Ideas

- Create Request entity (one to many with Item)
- Instantiate Transaction whenever an owner receives an item