Kyle Lin 862314961 klin112@ucr.edu
Vansh Nagpal 862228203 vnagp002@ucr.edu

CS 166 Project Phase 1
ER Diagram & Requirement Analysis

In this document, we analyze the requirements of the Pizza Delivery Application and state the necessary assumptions in order to outline a tentative ER diagram illustrating our schema design. Additionally, we briefly discuss our individual contributions to the project in this phase.

1. Data Schema and ER Diagram

1.1 User (Entity)

A user has the following information:

- phoneNum (string, required)
    - We assume this is at most 16 digits.
- login (key, string, required)
    - This is the key, so each user must have a unique login name.
    - This can be at most 32 characters.
- password (string, required)
    - This can be at most 32 characters.
- role (string, required)
    - This can either be “Customer”, “Driver”, or “Manager”. No other values are allowed.
    - The default is “Customer”.
- address (string, required)
    - This can be at most 128 characters.
- favoriteItem (string, optional)

    - If this is not null, then this must equal the itemName of some item in the Items table.

We assume that all managers have access to all stores and managers. This is necessary to simplify design, since we would otherwise need a “Superuser” role to manage the managers. By letting all managers manage everything, there is no need for such a role.

1.2 Item (Entity)

An item has the following information:

- itemName (key, string, required)
    - This is at most 32 characters.
    - This is the key, so each item must have a unique itemName.
- type (string, required)
    - This is at most 32 characters.
- price (unsigned integer, required)
    - This is the price in U.S. cents. For example, if an item costs $1.23, then its price field should be 123.
    - We assume that an item will cost at most $1,000 (meaning price can be at most 100,000).
- ingredients (string, optional)
    - This is a comma-separated list of ingredients.
    - This can be at most 512 characters. We based this number on the ingredients list of Cheetos.
    - Were we pressed for space, we could have stored this as a relation instead of an attribute. However, most users don’t expect to perform fancy queries on items, so a comma-separated string should be sufficient. Also, we assume we are working with reasonably adequate amounts of disk storage, so the minor memory overhead of using a 512-character attribute (instead of a separate relation) is acceptable to us.

- description (string, optional)
    - This is one or more English sentences describing the food. The character limit is 256.
- imageUrl (string, optional)
    - This points to a URL of an image of the item. The character limit is 128.

We assume that all items are available at all stores. In other words, there are no items that are only available at specific stores. This may slightly reduce the flexibility of the chefs (since all chefs must now collaborate to produce a unified menu). However, it greatly simplifies business management, as well as the design of the application.

1.3 Order (Entity)

An order has the following information

- orderId (key, unsigned integer, required)
    - The maximum value is 2^{32} - 1. This allows us to handle over a billion orders, which should be more than enough for a small business.
    - This is the key, so each order must have a unique orderId.
- login (string, required)
    - This refers to the user that placed the order.
    - This must equal the login of a user in the User table.
- orderTimestamp (Date, required)
- totalPrice (unsigned integer, required)
    - We assume that the total price will be at most $1,000,000 (meaning totalPrice can be at most 100,000,000).
- orderStatus (string, required)
    - This can either be “Order Received”, “In Transit”, or “Delivered”.
- storeId (string, required)
    - This refers to the store that the order was placed at.
    - This must equal the storeId of a store in the Store table.

1.4 Store (Entity)

A store has the following information:

- storeId (key, unsigned integer, required)
    - The maximum value is 2^{32} - 1. This allows us to track over a billion stores, which should be more than enough for a small business.
    - This is the key, so each store must have a unique storeId.
- address (string, required)
    - This can be at most 128 characters.
- city (string, required)
    - This can be at most 32 characters.
- state (string, required)
    - This can be at most 32 characters.
- isOpen (boolean, required)
    - This describes whether the store is currently open to orders.
- reviewScore (float, optional)
    - This is an average based on user reviews.

1.5 reviews (Relationship)

A review is a one-to-one relationship between users and stores. Participation is optional for both entities.

A review also has the following information:

- starCount (unsigned integer, required)
    - This may be any integer from 0 to 5.

We use the foreign multi-attribute key (login, storeId).

1.6 worksAt (Relationship)

The “worksAt” relationship is a many-to-many relationship between users and stores. In other words, we assume that a user may work at (e.g., drive, manage) multiple stores,

and that a store may have multiple workers. Participation is optional for both entities, because some users (e.g., customers) will not work at any store, and some stores may have no workers (e.g., closed stores).

We use the foreign multi-attribute key (login, storeId).

1.7 places (Relationship)

The “places” relationship is a one-to-many relationship between users and orders. Participation is mandatory for orders, while optional for users.

We use the multi-attribute key (login, storeId).

1.8 manages (Relationship)

The “manages” relationship is a one-to-many relationship between stores and orders. Participation is mandatory for orders, while optional for stores.

1.9 contains (Relationship)

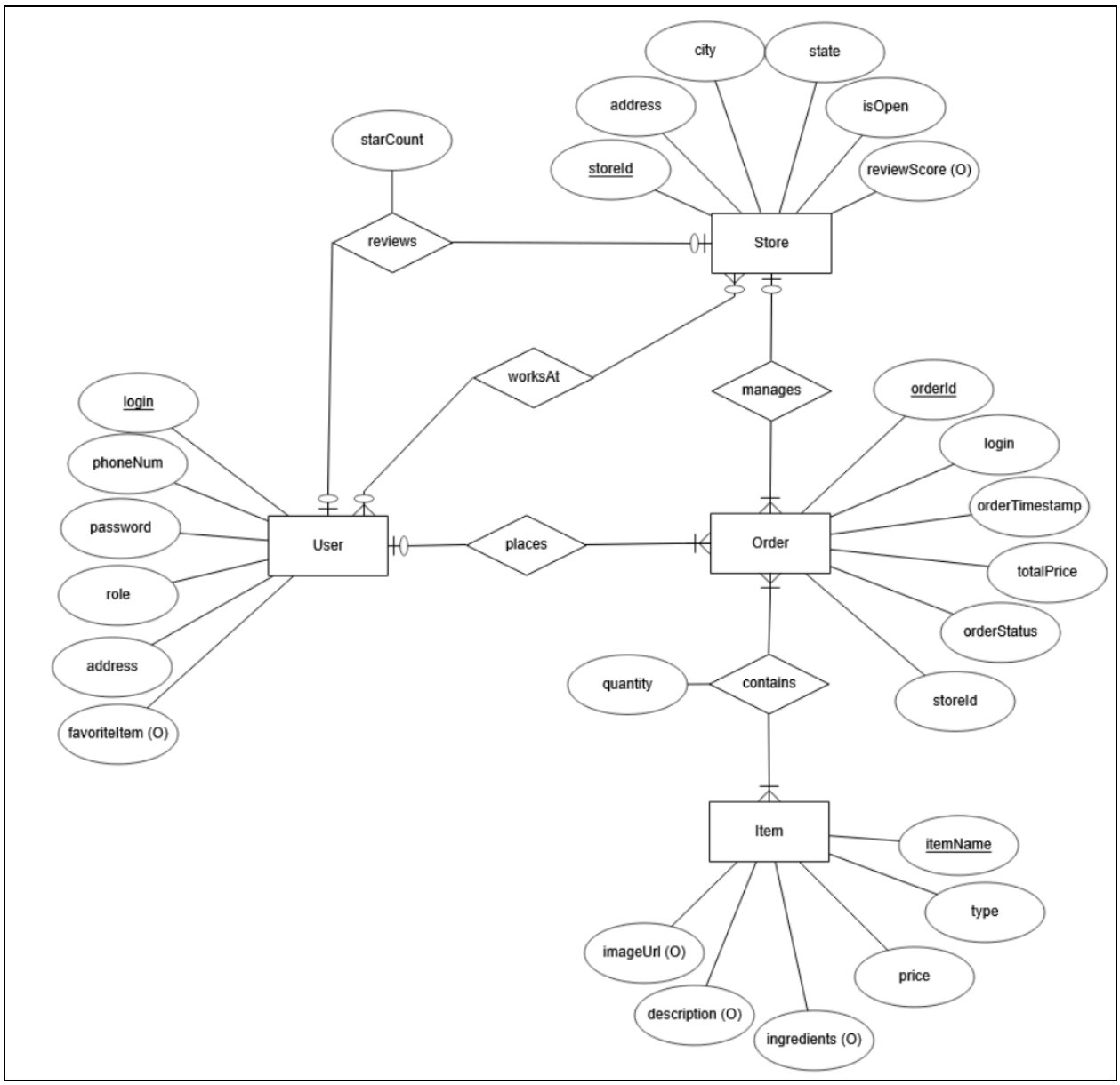
The “contains” relationship is a many-to-many relationship between orders and items. Participation is mandatory for orders but optional for items.

The relationship also has the following information:

- quantity (unsigned integer, required)
    - This allows a single order to contain multiple of the same item (e.g., three cheese pizzas).

1.10 ER Diagram

Here is the ER Diagram, generated with erdplus.com:



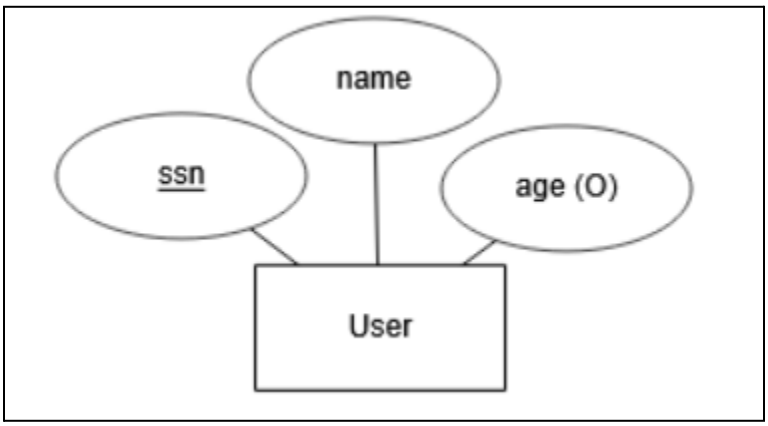
1.10 ER Diagram Notation

Primary Key:

A primary key is indicated by an underlined attribute belonging to the entity. Refer to the figure below.

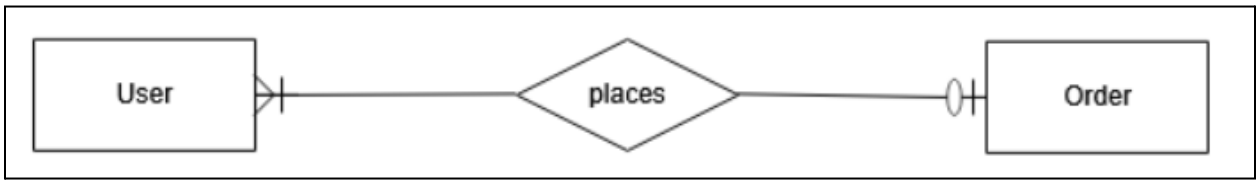
Optional Attribute:

An optional attribute is indicated by the presence of (O) in an attribute. Refer to the figure below.



Relationship notation (cardinality and participation constraints):

For the symbol nearest the entity box: a single line means “one”, and a crowsfoot means “many”. For the symbol farther from the entity box, a line means “mandatory” and a circle means “optional”.



2. Teamwork Distribution

Kyle and Vansh split the work in this phase roughly equally. Vansh worked more on the ER diagram, while Kyle worked more on the project report, but both partners contributed to both parts of the project.