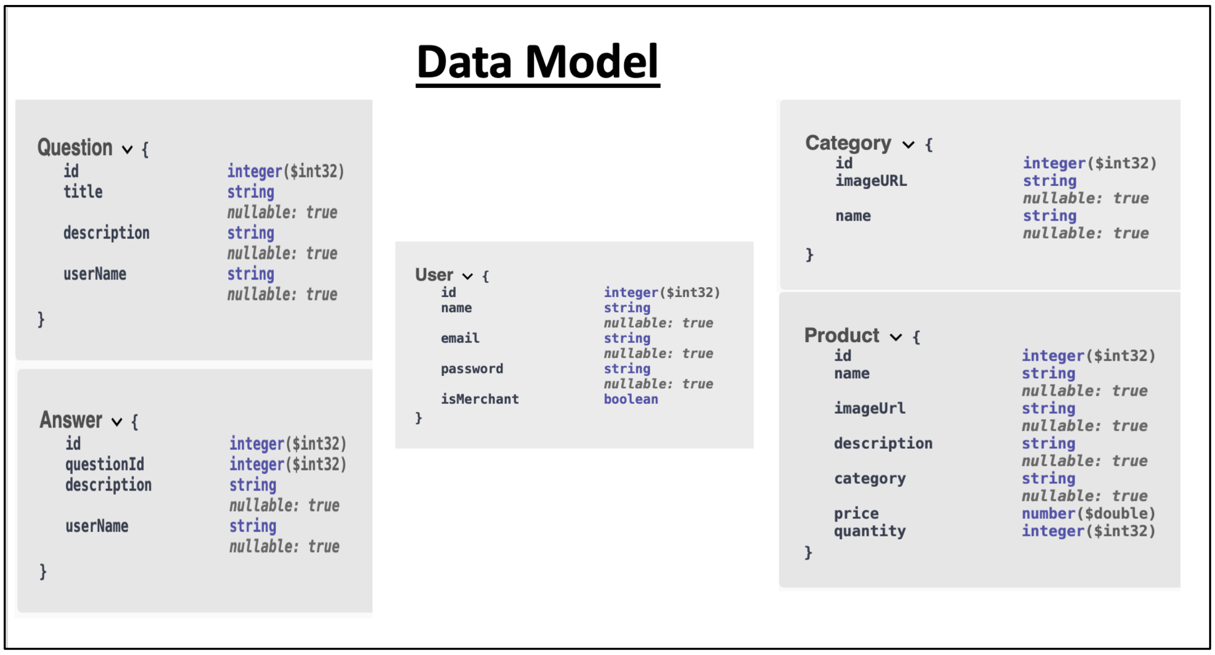
Marketplace Services – Data Model

Group 13



Data Model

```
Question ⌄ {
   id              integer($int32)
   title           string
                   nullable: true
   description     string
                   nullable: true
   userName        string
                   nullable: true
}
```

```
Answer ⌄ {
   id              integer($int32)
   questionId      integer($int32)
   description     string
                   nullable: true
   userName        string
                   nullable: true
}
```

```
User ⌄ {
   id              integer($int32)
   name            string
                   nullable: true
   email           string
                   nullable: true
   password        string
                   nullable: true
   isMerchant      boolean
}
```

```
Category ⌄ {
   id              integer($int32)
   imageURL        string
                   nullable: true
   name            string
                   nullable: true
}
```

```
Product ⌄ {
   id              integer($int32)
   name            string
                   nullable: true
   imageUrl        string
                   nullable: true
   description     string
                   nullable: true
   category        string
                   nullable: true
   price           number($double)
   quantity        integer($int32)
}
```

Figure 1 Data Model for Shoppers Marketplace

Relation between the Documents

1. User: We store basic user details including Name, email, password and a Boolean isMerchant flag. All users sign up as customers, and we have only one merchant with the credentials: merchant@marketplace.ca and 123456 as the password. We store the password in plain text since securing the application is not in the project's scope. If needed, we can store the password using SHA-256 hash and update our ValidateUser() API. Unique field is Id and Email.
2. Question: Each question contains two fields, question title and a brief question description along with the username of the user who posted the question. Unique field is Id.
3. Answer: Each answer has the description field that stores the responses of users posting replies to questions on the discussion forum. Each answer is linked with the question using the questionId field. Unique field is Id.
4. Product: Each product has a description, name, the associated category name, price and quantity. Unique field is Id and Name.
5. Category: Each category has its image URL and name. Updating and deleting a category also updates/deletes the associated products. Unique field is Id and Name.

Decision Rationale

Please note that since database is not integrated the string fields are nullable. We have added null checks in all API calls to ensure no null data exists. For ID fields, ideally we would have used UUID, since we have a limited number of users we are using only int32. If users increase, we can refactor it to String and use UUID generator to ensure uniqueness, we can also incorporate this change in our third deliverable.

For Category and Product, APIs are configured so that no two Products or Categories exist with same name i.e. Name is also unique along with Id field.

Using image URL as string instead of byte stream – considering the time and prioritizing developing other functionalities, we directly use the relative path of images uploaded in our assets folder. In the third deliverable, where we need to show all modules working properly, we plan to choose one option from the following:

1. Get web image URL as String (existing) from user and use Flutter's Image.Network() API (https://docs.flutter.dev/cookbook/images/network-image) to load the image
2. Store byte stream in database and encode/decode accordingly in UI