1)

Relational Model

The data in student and class project are simple and more static.

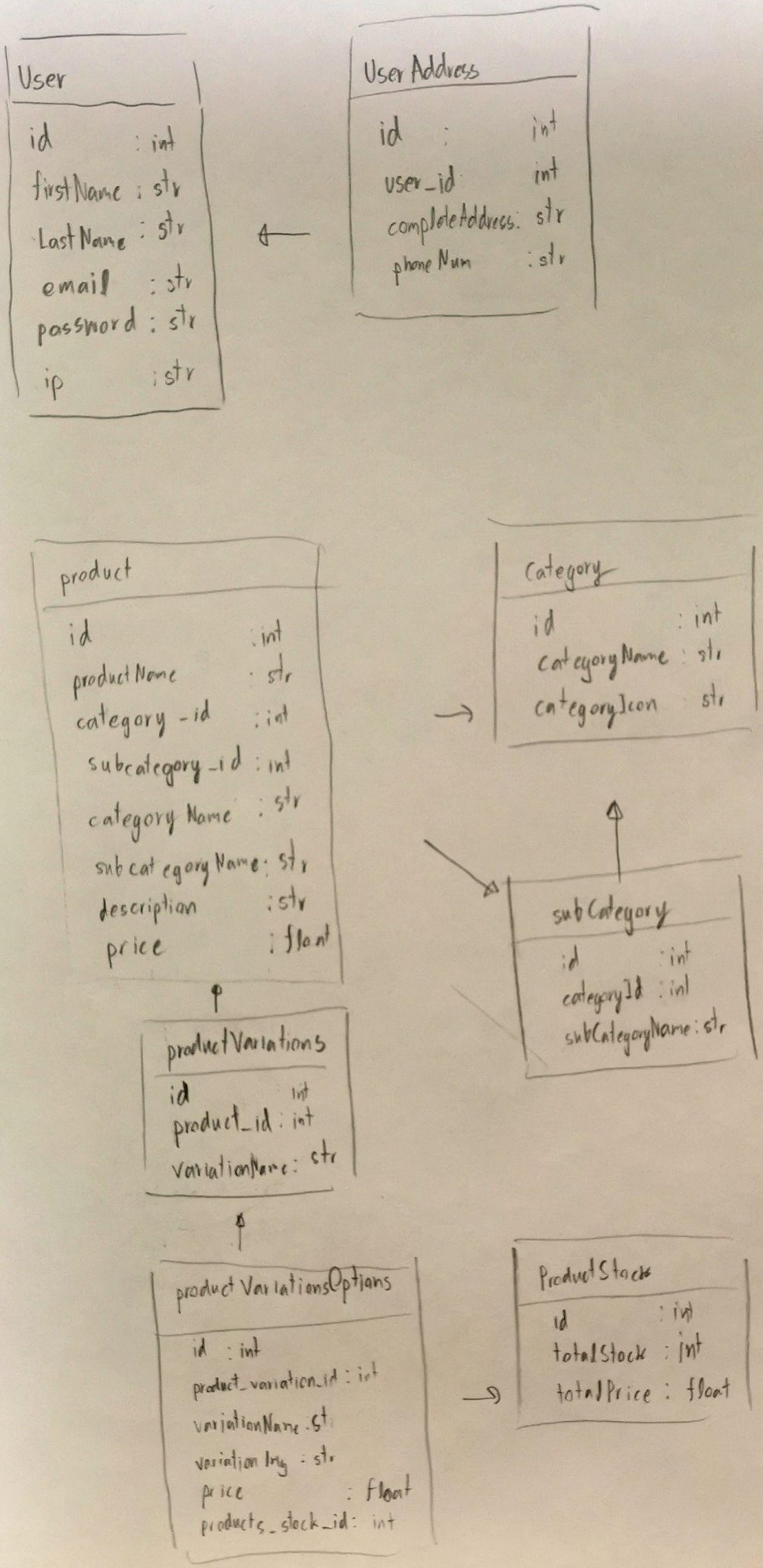
2) MongoDB

The data that require a lot of optional detail the mongodb is easier to collect the information since it's flexible can deal with the complex data.

3) MongoDB

Since it's use a sensor measurement and send data as time-sequence and reading that contains ten labeled values. That a big amount of data the MongoDB is much easier to update data and requirement of sensor change

4) E-commerce


User
id : int
firstName : str
LastName : str
email : str
password : str
ip : str
User Address
id : int
user_id int
completeAddress: str
phone Num : str
product
id : int
productName : str
category_id : int
subcategory_id : int
category Name : str
subcategoryName : str
description : str
price : float
Category
id : int
categoryName : str
categoryIcon : str
subCategory
id : int
categoryId : int
subCategoryName : str
productVariations
id int
product_id : int
variationName : str
productVariationsOptions
id : int
product_variation_id : int
variationName : str
variationImg : str
price : float
products_stock_id : int
ProductStock
id : int
totalStock : int
totalPrice : float

- Find the total marks for each student across all subjects.

```
db.students.aggregate([{ $group : { _id:"$name", sum_score: { $sum : "$marks" }}}]);
{ _id: 'Alison', sum_score: 252 }
{ _id: 'Rav', sum_score: 216 }
{ _id: 'Jan', sum_score: 0 }
{ _id: 'Ramesh', sum_score: 223 }
{ _id: 'Steve', sum_score: 247 }
```

- Find the maximum marks scored in each subject.

```
db.students.aggregate({ $group : { _id:"$subject", max_score: { $max : "$marks" }}});
{ _id: 'maths', max_score: 87 }
{ _id: 'english', max_score: 89 }
{ _id: 'science', max_score: 86 }
```

- Find the minimum marks scored by each student.

```
db.students.aggregate({ $group : { _id:"$subject", max_score: { $min : "$marks" }}});
{ _id: 'maths', max_score: 62 }
{ _id: 'english', max_score: 0 }
{ _id: 'science', max_score: 71 }
```

- Find the top two subjects based on average marks.

```
var limit = 2
db.students.aggregate([{ $group : { _id:"$subject", avg_score: { $avg : "$marks" }}}, {$sort:{avg_score:-1}} ,{$limit:limit}]);
{ _id: 'maths', avg_score: 78.5 }
{ _id: 'science', avg_score: 77.75 }
```