1. We will use schema as our application typically works with certain types of data (certain schema).
2. Let’s see how to think about modeling our schema.   
   I will give you some guidelines or hints that you can keep in mind when you think about how we should structure my data.
3. One important question is which data does my app need or generate.  
   **App**: Mobile App, Website, a smart device generating data like a tracker, a fitness tracker sending GPS coordinates.
4. You then also have to think about where do I need my data. So for example: If you’re building a website, do I need it on the welcome page or on the product list page, or on the order page.  
   Second, which kind of data do I need on all these pages. Because the idea with mongoDB is that you store your data in the format you need it in your application and if you need the same data in different formats, then you want to find a good structure that covers all these use cases or if absolutely necessary, you could even split it into multiple collections. So this defines your required collections. For example: You have an orders collection for the orders page, then you have a product collection for the product page and it also defines which fields you have in the documents of a collection and how you might grough these fields together.   
   So how you might relate to these fields, more about relations in a second.
5. You also should which kind of data or information do I want to display. So on these different pages, which kind of data do I want to show on there so define your queries accordingly. So do I display a list of products or a single product. Is it a find() or findOne() query? How should I configure this query? Am I looking for a product which a specific ID or Am I looking for all products and
6. You also need to ask youself how often do I fetch my data, on every page reload or every second or not that often because that also tells you if you should optimize for easy fetching because you’re fetching a lot but maybe you’re not writing data that much. So you might be fine with having some duplicate data if that speeds up the fetching process because you don’t have to join data manually or do you often write and change your data. If you have an application where your data gets changed a lot or parts of your data get changed a lot you want to ensure that these parts of the data that do get changed a lot. Let’s say we have a lot of orders but the product metadata doesn’t change that often you want to make sure that your orders are not unnecessarily duplicated across collections but that you have one collection with your orders where you can write new orders too, product metadata on the other hand might be distributing across different collections because well you don’t change it that often. So, if you change it, you have to touch multiple collections but that is no problem. The main focus is that you can fetch that data from the different parts of your application in the format you need it.
7. 